

Name: Villasenor, Hans Rainier A.	Date Performed: 10/3/25
Course/Section: CPE 212 CPE31S2	Date Submitted:10/3/25
Instructor: Engr. Robin Valenzuela	Semester and SY: 1st sem 2025-26
Activity 6: Targeting Specific Nodes and Managing Services	
<p>1. Objectives:</p> <ul style="list-style-type: none"> 1.1 Individualize hosts 1.2 Apply tags in selecting plays to run 1.3 Managing Services from remote servers using playbooks 	
<p>2. Discussion:</p> <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>Requirement:</p> <p>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>	
Task 1: Targeting Specific Nodes	
<ul style="list-style-type: none"> 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"

```

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

```

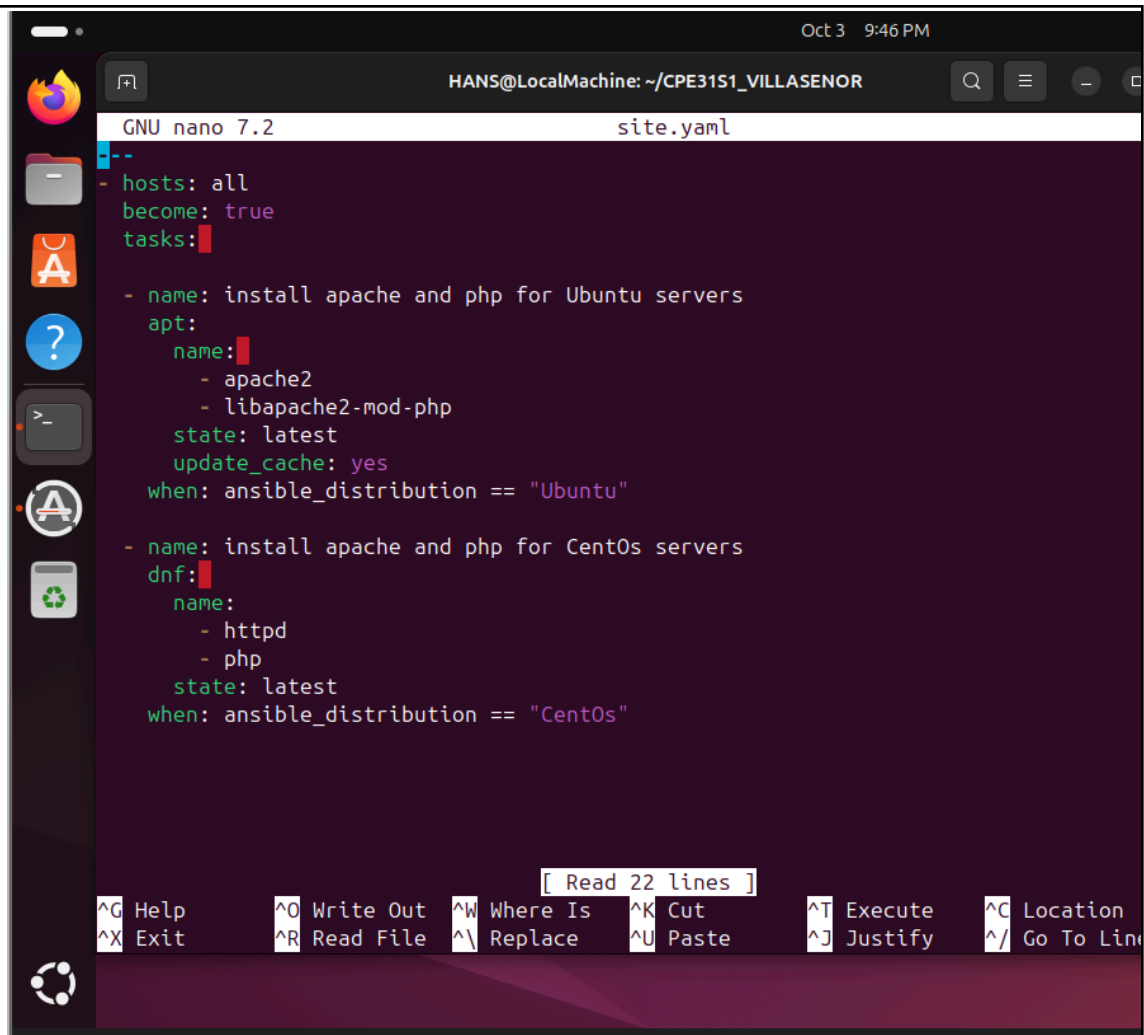
Make sure to save the file and exit.

```

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

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

```



```
Oct 3 9:46 PM
HANS@LocalMachine: ~/CPE3151_VILLASENOR
GNU nano 7.2 site.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"
```

[Read 22 lines]

^G Help	^O Write Out	^W Where Is	^K Cut	^T Execute	^C Location
^X Exit	^R Read File	^\ Replace	^U Paste	^J Justify	^_ Go To Line

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.yaml* 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.

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.

```

PLAY [web_servers] *****

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

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

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

PLAY RECAP *****
192.168.56.102      : ok=4    changed=0    unreachable=0    failed=0    skip
2   rescued=0    ignored=0
192.168.56.103      : ok=0    changed=0    unreachable=1    failed=0    skip
0   rescued=0    ignored=0
192.168.56.104      : ok=4    changed=0    unreachable=0    failed=0    skip
2   rescued=0    ignored=0
192.168.56.105      : ok=0    changed=0    unreachable=1    failed=0    skip
0   rescued=0    ignored=0

```

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 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: [192.168.56.102]
ok: [192.168.56.104]

TASK [install mariadb package(CentOs)] *****
skipping: [192.168.56.102]
ok: [192.168.56.104]

TASK [Mariadb- Restarting/Enabling] *****
changed: [192.168.56.104]
changed: [192.168.56.102]

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

```

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.

```
TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

[hans@localhost ~]$ systemctl status mariadb
● mariadb.service - MariaDB 10.5 database server
   Loaded: loaded (/usr/lib/systemd/system/mariadb.service; enabled; preset: enabled)
   Active: active (running) since Fri 2025-10-03 22:04:41 PST; 36s ago
     Docs: man:mariadb(8)
           https://mariadb.com/kb/en/library/systemd/
  Process: 39528 ExecStartPre=/usr/libexec/mariadb-check-socket (code=exited, status=0/SUCCESS)
  Process: 39550 ExecStartPre=/usr/libexec/mariadb-prepare-db-dir mariadb.service (code=exited, status=0/SUCCESS)
  Process: 39598 ExecStartPost=/usr/libexec/mariadb-check-upgrade (code=exited, status=0/SUCCESS)
 Main PID: 39585 (mariabdd)
    Status: "Taking your SQL requests now..."
     Tasks: 12 (limit: 10518)
  Memory: 65.2M (peak: 66.6M)
     CPU: 322ms
    CGroup: /system.slice/mariadb.service
            └─39585 /usr/libexec/mariabdd --basedir=/usr

Oct 03 22:04:40 localhost.localdomain systemd[1]: Starting MariaDB 10.5 database server: mariadbd.
Oct 03 22:04:41 localhost.localdomain mariadb-prepare-db-dir[39550]: Database system is not initialized. If this is the first time that MariaDB is started, run: mysql_install_db or mariadb_install_db to initialize it.
Oct 03 22:04:41 localhost.localdomain mariadb-prepare-db-dir[39550]: If this error persists, then please read the documentation for your version of MariaDB.
Oct 03 22:04:41 localhost.localdomain systemd[1]: Started MariaDB 10.5 database server: mariadbd.

lines 1-20/20 (END)
```

CentOs

```
HANS@Server1:~$ systemctl status mariadb
● mariadb.service - MariaDB 10.11.13 database server
   Loaded: loaded (/usr/lib/systemd/system/mariadb.service; enabled; preset: enabled)
   Active: active (running) since Fri 2025-10-03 14:04:42 UTC; 1min 52s ago
     Docs: man:mariadb(8)
           https://mariadb.com/kb/en/library/systemd/
  Process: 10832 ExecStartPre=/usr/bin/install -m 755 -o mysql -g root -d /usr/share/mysql (code=exited, status=0/SUCCESS)
  Process: 10834 ExecStartPre=/bin/sh -c systemctl unset-environment _WSR (code=exited, status=0/SUCCESS)
  Process: 10836 ExecStartPre=/bin/sh -c [ ! -e /usr/bin/galera_recovery ] && mv /usr/bin/galera_recovery /usr/share/galera_recovery || true; systemctl start mariadb (code=exited, status=0/SUCCESS)
  Process: 10907 ExecStartPost=/bin/sh -c systemctl unset-environment _WSR (code=exited, status=0/SUCCESS)
  Process: 10909 ExecStartPost=/etc/mysql/debian-start (code=exited, status=0/SUCCESS)
 Main PID: 10896 (mariadbd)
    Status: "Taking your SQL requests now..."
     Tasks: 11 (limit: 16656)
  Memory: 103.0M (peak: 106.3M)
     CPU: 3.258s
    CGroup: /system.slice/mariadb.service
            └─10896 /usr/sbin/mariadbd

lines 1-17/17 (END)
```

Ubuntu

Describe the output.

Both ubuntu and CentOs all downloaded an mariadb

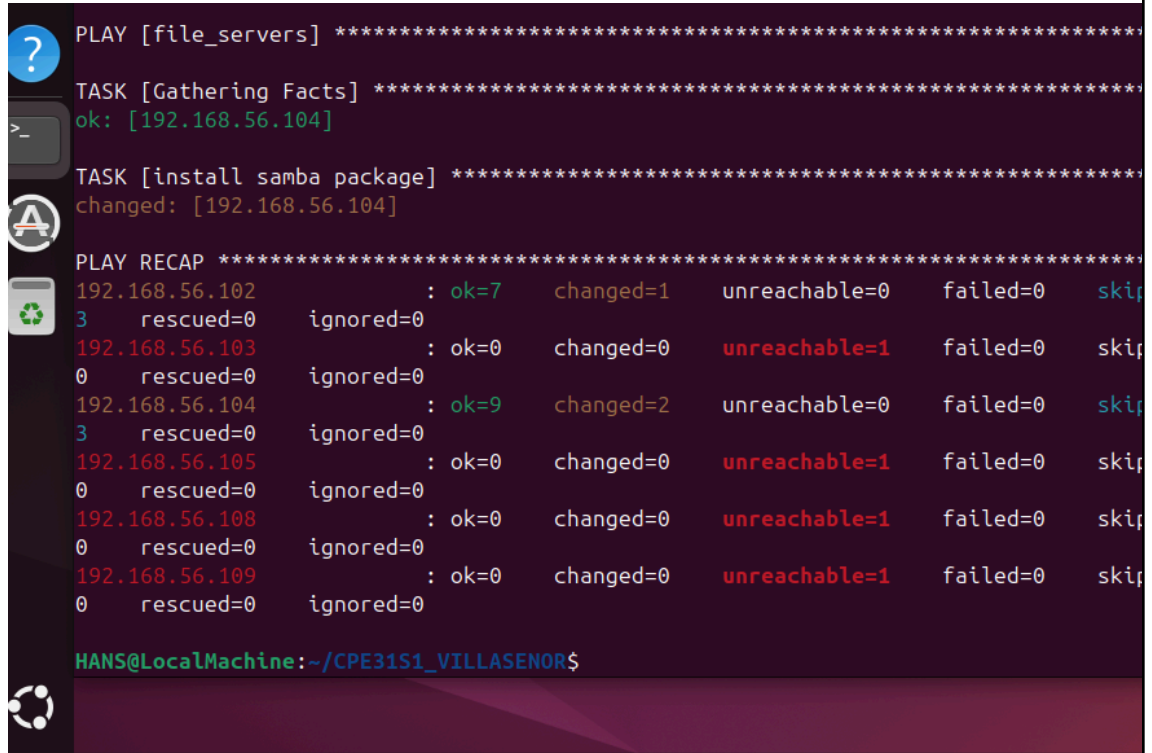
6. Edit the *site.yml* again. This time we will append the code to configure installation on the *file_servers* group. We can add the following on our file.

```
- 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: [192.168.56.104]

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

PLAY RECAP *****
192.168.56.102      : ok=7    changed=1    unreachable=0    failed=0    skip
3   rescued=0    ignored=0
192.168.56.103      : ok=0    changed=0    unreachable=1    failed=0    skip
0   rescued=0    ignored=0
192.168.56.104      : ok=9    changed=2    unreachable=0    failed=0    skip
3   rescued=0    ignored=0
192.168.56.105      : ok=0    changed=0    unreachable=1    failed=0    skip
0   rescued=0    ignored=0
192.168.56.108      : ok=0    changed=0    unreachable=1    failed=0    skip
0   rescued=0    ignored=0
192.168.56.109      : ok=0    changed=0    unreachable=1    failed=0    skip
0   rescued=0    ignored=0

HANS@LocalMachine:~/CPE31S1_VILLASENOR$
```

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

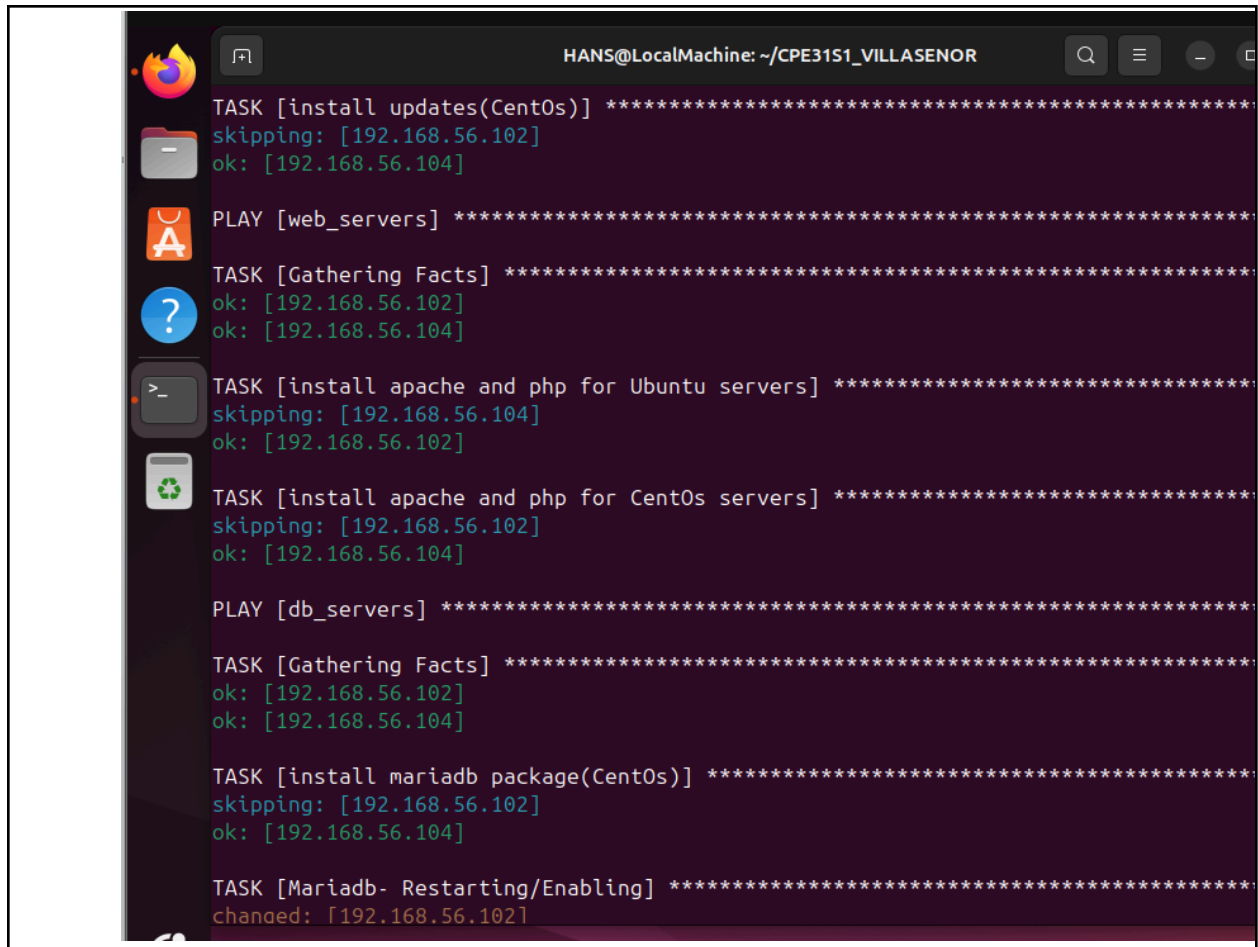
    - name: install mariadb package (Ubuntu)
      tags: db, mariadb, ubuntu
      apt:
        name: mariadb-server
        state: latest
        when: ansible_distribution == "Ubuntu"

- 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.



```
HANS@LocalMachine: ~/CPE3151_VILLASENOR

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

PLAY [web_servers] *****

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

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

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

PLAY [db_servers] *****

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

TASK [install mariadb package(CentOs)] *****
skipping: [192.168.56.102]
ok: [192.168.56.104]

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

```
HANS@LocalMachine: ~/CPE3151_VILLASENOR
changed: [192.168.56.102]
changed: [192.168.56.104]

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

PLAY [file_servers] *****

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

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

PLAY RECAP *****
192.168.56.102      : ok=6    changed=1    unreachable=0    failed=0    skip
3   rescued=0    ignored=0
192.168.56.103      : ok=0    changed=0    unreachable=1    failed=0    skip
0   rescued=0    ignored=0
192.168.56.104      : ok=9    changed=1    unreachable=0    failed=0    skip
2   rescued=0    ignored=0
192.168.56.105      : ok=0    changed=0    unreachable=1    failed=0    skip
0   rescued=0    ignored=0
192.168.56.108      : ok=0    changed=0    unreachable=1    failed=0    skip
0   rescued=0    ignored=0
192.168.56.109      : ok=0    changed=0    unreachable=1    failed=0    skip
0   rescued=0    ignored=0

HANS@LocalMachine:~/CPE3151_VILLASENOR$
```

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

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

```
HANS@LocalMachine:~/CPE3151_VILLASENOR$ ansible-playbook --list-tags site1.yaml

playbook: site1.yaml

  play #1 (all): all    TAGS: []
    TASK TAGS: [always]

  play #2 (web_servers): web_servers    TAGS: []
    TASK TAGS: [apache, apache2, centos, httpd, ubuntu]

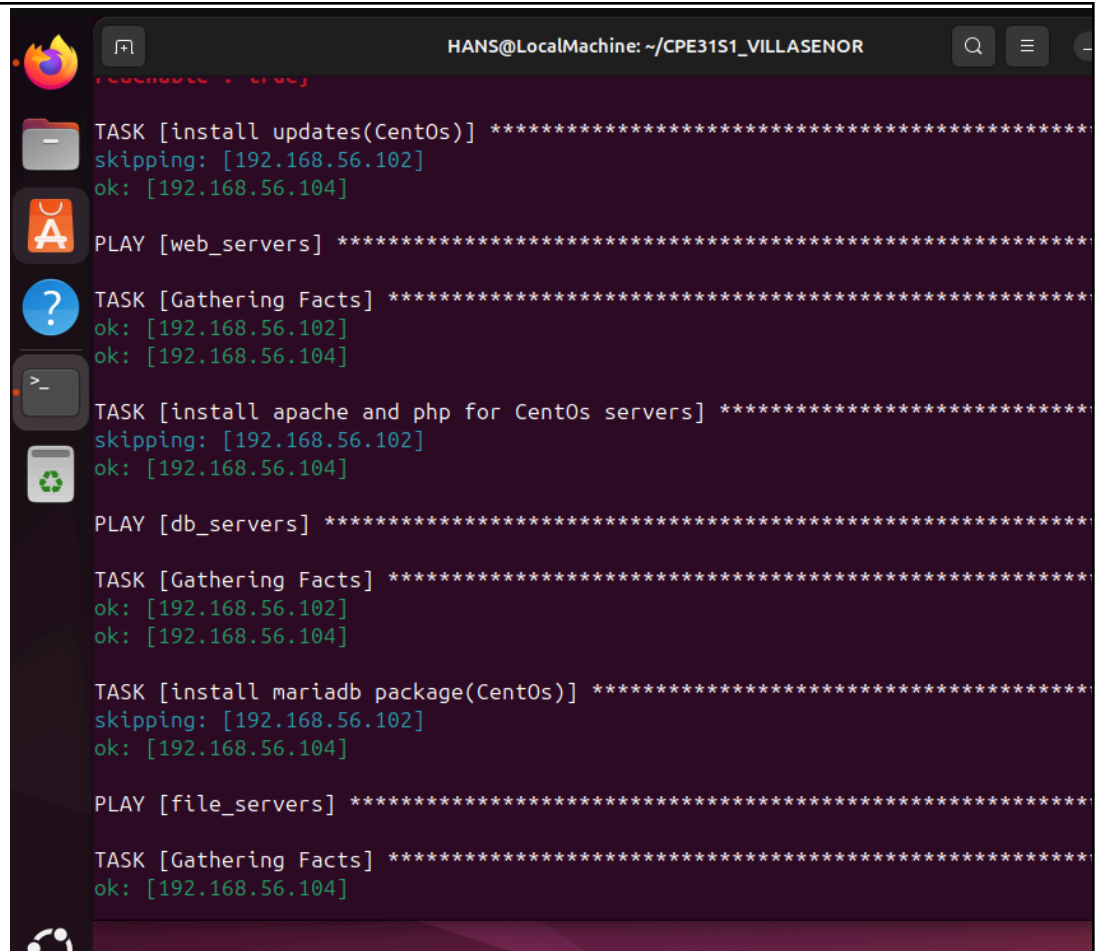
  play #3 (db_servers): db_servers    TAGS: []
    TASK TAGS: [centos, db, mariadb, ubuntu]

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

HANS@LocalMachine:~/CPE3151_VILLASENOR$
```

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

This tags just run or play all have centOs

A terminal window titled 'HANS@LocalMachine: ~/CPE31S1_VILLASENOR' displays the output of an Ansible playbook run with the 'db' tag. The output shows several tasks and plays. The first task, 'install updates(CentOs)', is skipped for 192.168.56.102 and successful for 192.168.56.104. This is followed by a 'PLAY [web_servers]' block. Then, a 'TASK [Gathering Facts]' is shown, successful for both hosts. Next, a task to 'install apache and php for CentOs servers' is skipped for 192.168.56.102 and successful for 192.168.56.104. This is followed by a 'PLAY [db_servers]' block. Then, another 'TASK [Gathering Facts]' is shown, successful for both hosts. Next, a task to 'install mariadb package(CentOs)' is skipped for 192.168.56.102 and successful for 192.168.56.104. This is followed by a 'PLAY [file_servers]' block. Finally, a last 'TASK [Gathering Facts]' is shown, successful for 192.168.56.104. The terminal has a dark purple background with a sidebar on the left containing icons for a file manager, application store, help, terminal, and a trash can.

```
HANS@LocalMachine: ~/CPE31S1_VILLASENOR
TASK [install updates(CentOs)] *****
skipping: [192.168.56.102]
ok: [192.168.56.104]

PLAY [web_servers] *****

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

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

PLAY [db_servers] *****

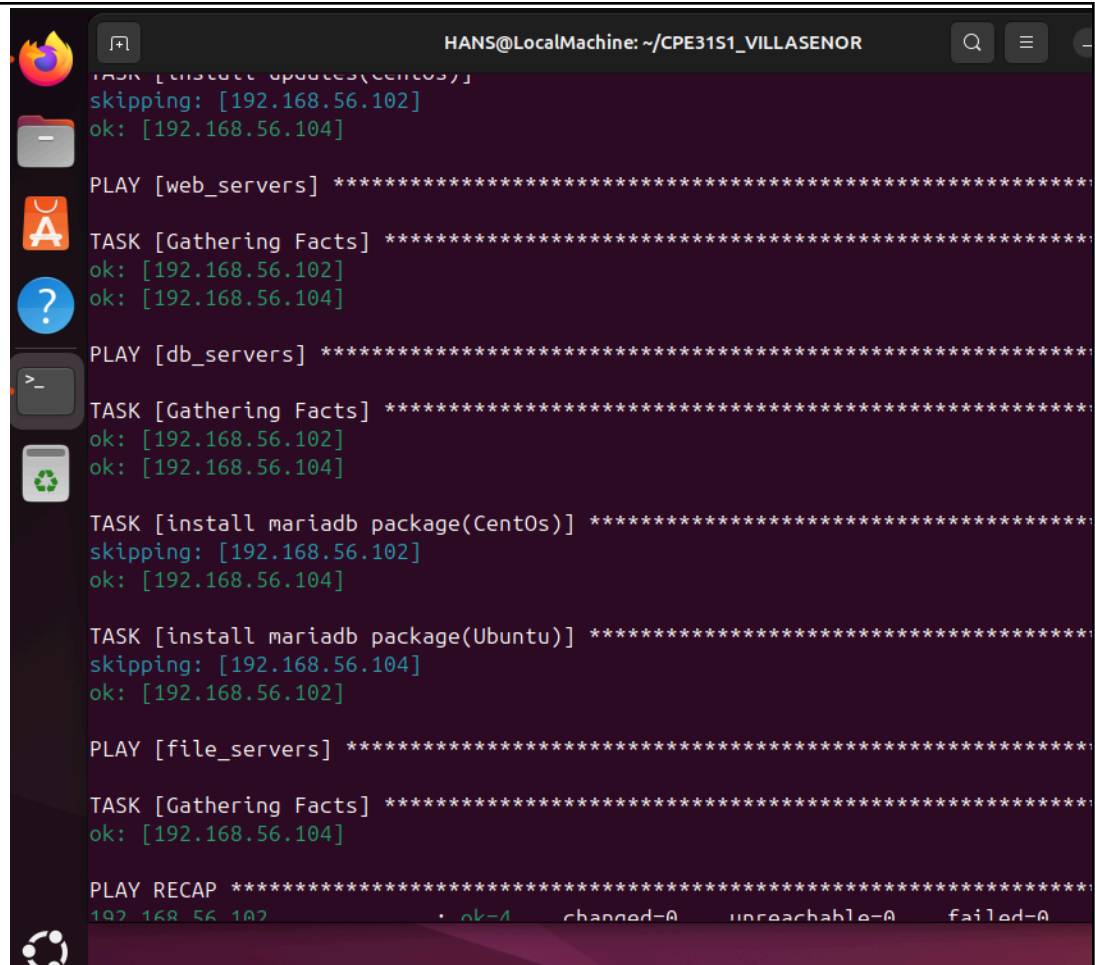
TASK [Gathering Facts] *****
ok: [192.168.56.102]
ok: [192.168.56.104]

TASK [install mariadb package(CentOs)] *****
skipping: [192.168.56.102]
ok: [192.168.56.104]

PLAY [file_servers] *****

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

2.3 *ansible-playbook --tags db --ask-become-pass site.yml*
it just run or play does have db in their tags

A terminal window titled 'HANS@LocalMachine: ~/CPE3151_VILLASENOR' displays the output of an Ansible playbook. The output shows several tasks being executed on two hosts: 192.168.56.102 and 192.168.56.104. The tasks include installing updates on CentOS, gathering facts, installing the mariadb package on CentOS and Ubuntu, and setting up web and database servers. The output is color-coded: green for 'ok', blue for 'skipping', and red for 'failed'. The terminal window has a dark background and a sidebar with various application icons on the left.

```
HANS@LocalMachine: ~/CPE3151_VILLASENOR
TASK [install updates(CentOS)] *****
skipping: [192.168.56.102]
ok: [192.168.56.104]

PLAY [web_servers] *****

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

PLAY [db_servers] *****

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

TASK [install mariadb package(CentOs)] *****
skipping: [192.168.56.102]
ok: [192.168.56.104]

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

PLAY [file_servers] *****

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

PLAY RECAP *****
192.168.56.102 : ok=4 changed=0 unreachable=0 failed=0
```

2.4 *ansible-playbook --tags apache --ask-become-pass site.yml*
It just run or play does have apache in their tags

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

PLAY [web_servers] *****

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

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

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

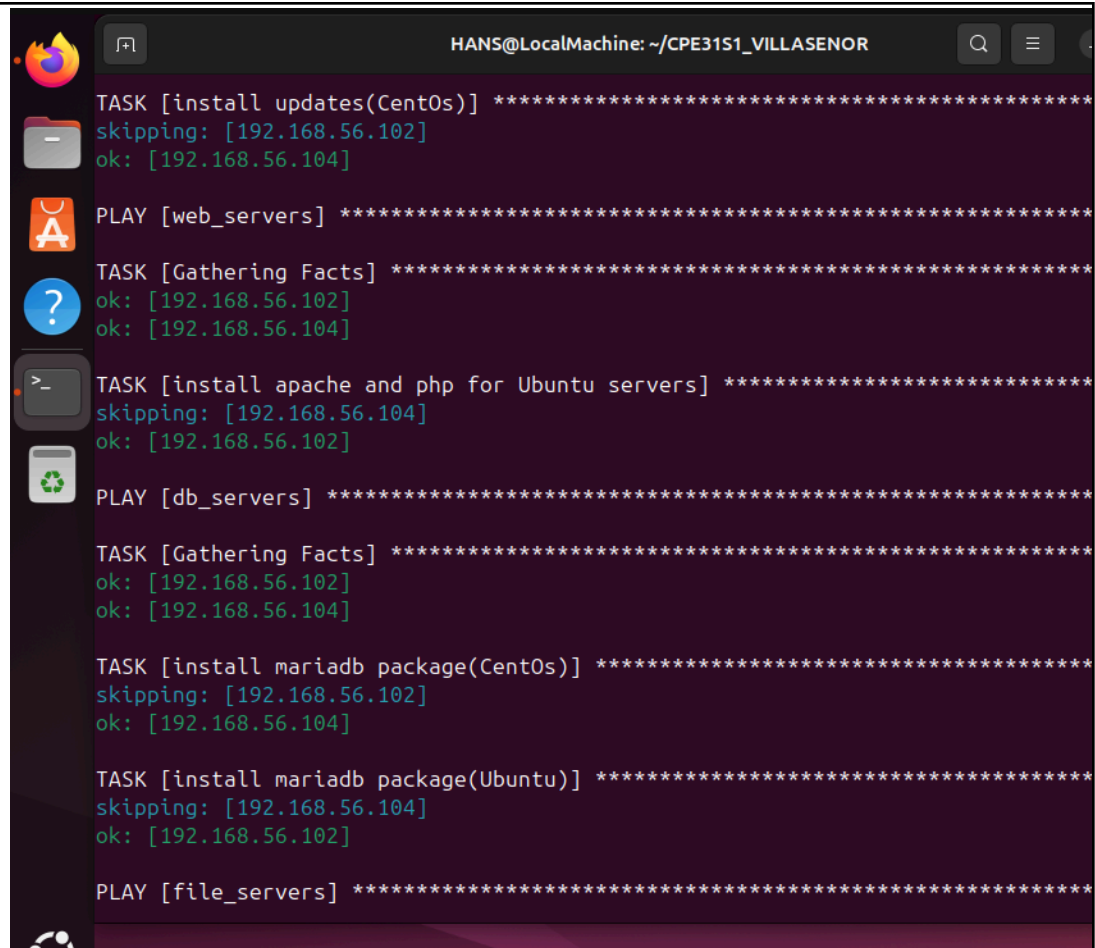
PLAY [db_servers] *****

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

PLAY [file_servers] *****

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

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

A terminal window titled 'HANS@LocalMachine: ~/CPE31S1_VILLASENOR' displays the output of an Ansible playbook. The output is color-coded: green for success, red for failure, and yellow for warnings. The tasks are grouped by playbooks: 'install updates(CentOs)', 'web_servers', 'Gathering Facts', 'install apache and php for Ubuntu servers', 'db_servers', and 'file_servers'. The 'install updates(CentOs)' task shows 'skipping: [192.168.56.102]' and 'ok: [192.168.56.104]'. The 'web_servers' play shows 'Gathering Facts' with 'ok: [192.168.56.102]' and 'ok: [192.168.56.104]'. The 'install apache and php for Ubuntu servers' task shows 'skipping: [192.168.56.104]' and 'ok: [192.168.56.102]'. The 'db_servers' play shows 'Gathering Facts' with 'ok: [192.168.56.102]' and 'ok: [192.168.56.104]'. The 'install mariadb package(CentOs)' task shows 'skipping: [192.168.56.102]' and 'ok: [192.168.56.104]'. The 'install mariadb package(Ubuntu)' task shows 'skipping: [192.168.56.104]' and 'ok: [192.168.56.102]'. The 'file_servers' play is partially visible at the bottom.

```
HANS@LocalMachine: ~/CPE31S1_VILLASENOR

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

PLAY [web_servers] *****

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

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

PLAY [db_servers] *****

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

TASK [install mariadb package(CentOs)] *****
skipping: [192.168.56.102]
ok: [192.168.56.104]

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

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

Task 3: Managing Services

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

```

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

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

```

Figure 3.1.1

Make sure to save the file and exit.

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

```

- hosts: db_servers
  become: true
  tasks:

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

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

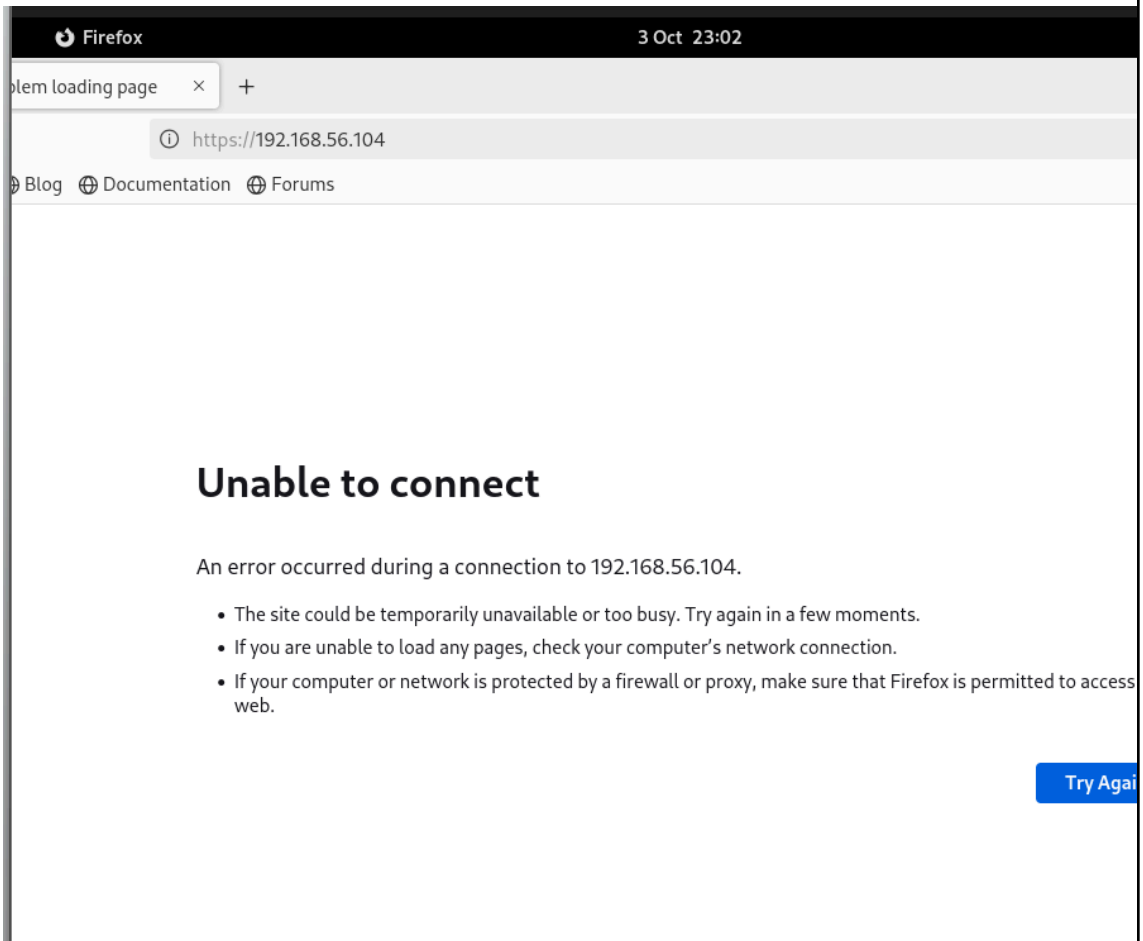
```

Figure 3.1.2

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

2. To test it, before you run the saved playbook, go to the CentOS server and stop the currently running httpd using the command *sudo systemctl stop httpd.*

When prompted, enter the sudo password. After that, open the browser and enter the CentOS server's IP address. You should not be getting a display because we stopped the httpd service already.



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.

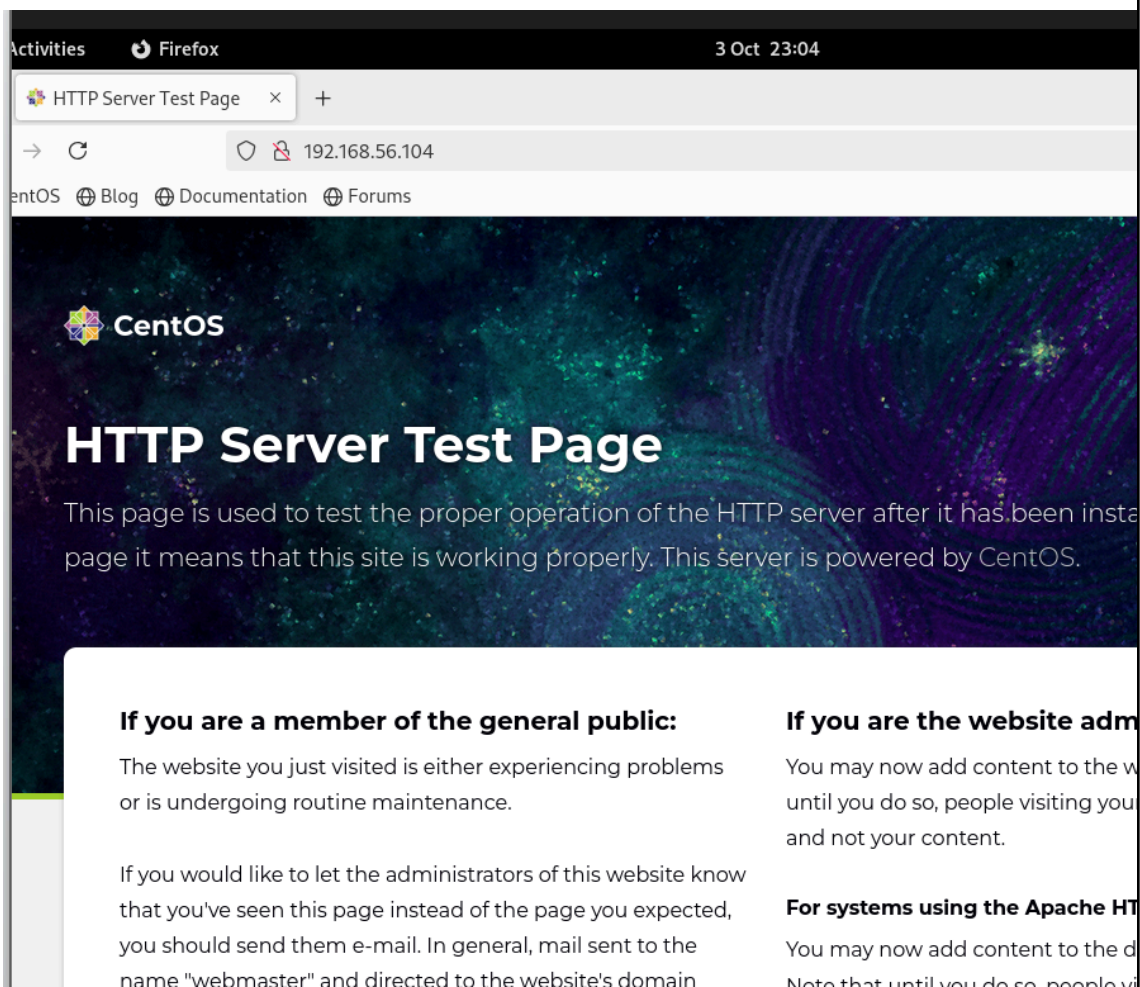
```
OCT 3 11:04 PM
HANS@LocalMachine: ~/CPE31S1_VILLASENOR

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

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

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

TASK [start httpd(CentOs)] *****
skipping: [192.168.56.102]
changed: [192.168.56.104]
```



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?
It become more easy to manage while using remote servers as a groups
2. What is the importance of tags in playbooks?
It will allow you to run an specific part in the playbook
3. Why do think some services need to be managed automatically in playbooks?
Automating their management makes sure they are started, restarted, or updated without human error.