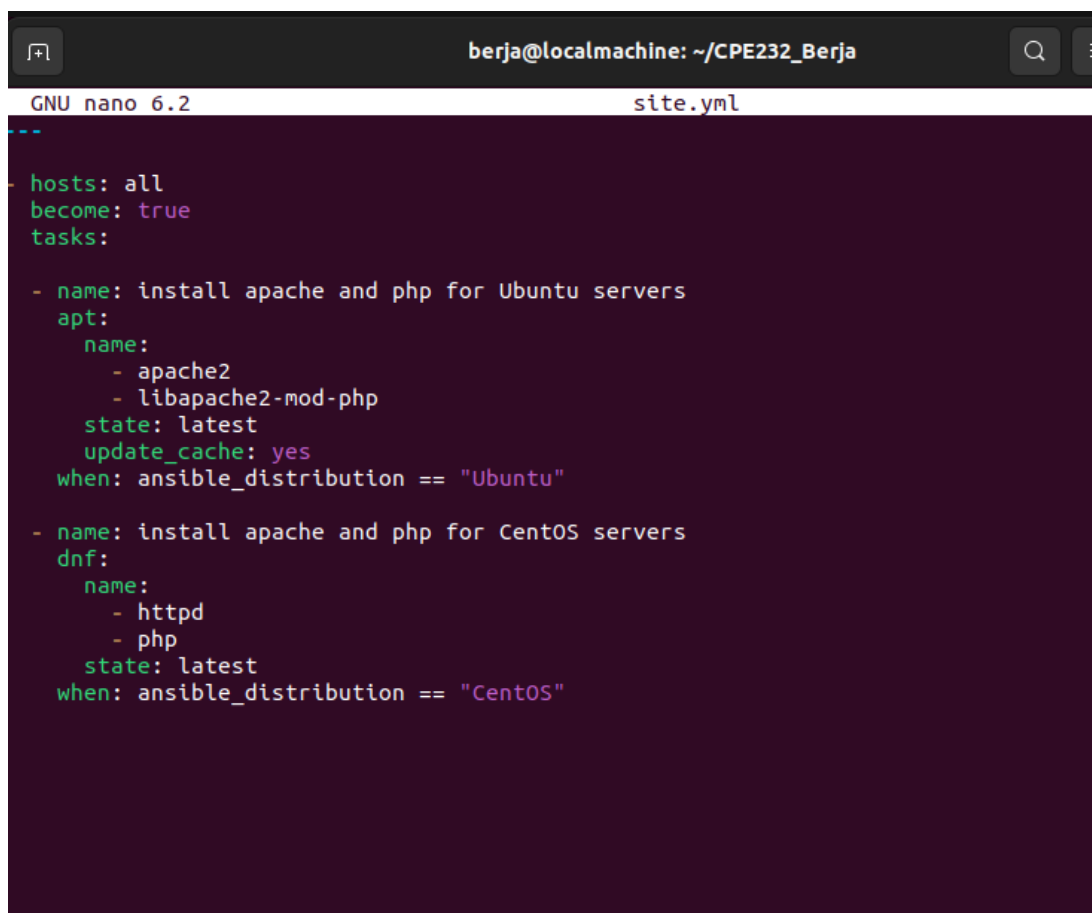


|   |   |
|---|---|
| <b>Name: Christian Dave L. Berja</b>  | <b>Date Performed: 10/10/2023</b>         |
| <b>Course/Section: CPE31S5</b>  | <b>Date Submitted: 15/10/2023</b>         |
| <b>Instructor: Engr. Roman Richard</b>  | <b>Semester and SY: 1st semester 2023</b> |
| <b>Activity 6: Targeting Specific Nodes and Managing Services</b>   |   |
| <p><b>1. Objectives:</b></p> <ul style="list-style-type: none"> <li>1.1 Individualize hosts</li> <li>1.2 Apply tags in selecting plays to run</li> <li>1.3 Managing Services from remote servers using playbooks</li> </ul>   |   |
| <p><b>2. Discussion:</b></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><b>Requirement:</b></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> |   |
| <b>Task 1: Targeting Specific Nodes</b>   |   |
| <ul style="list-style-type: none"> <li>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.</li> </ul>   |   |

```
---
- 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"
```



```
GNU nano 6.2 site.yml
---
- hosts: all
  become: true
  tasks:

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

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

2. Edit the inventory file. Remove the variables we put in our last activity and group according to the image shown below:

```
[web_servers]
192.168.56.120
192.168.56.121

[db_servers]
192.168.56.122

[file_servers]
192.168.56.123
```

Make sure to save the file and exit.

```
GNU nano 6.2 inventory
[web_servers]
192.168.56.109
192.168.56.111

[db_servers]
192.168.110
192.168.56.111

[file_servers]
192.168.56.110|

[ Wrote 10 lines ]
^G Help      ^O Write Out ^W Where Is  ^K Cut      ^T Execute  ^C Locati
^X Exit      ^R Read File ^\ Replace  ^U Paste    ^J Justify  ^/ Go To
```

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.

```
berja@localmachine: ~/CPE232_Berja
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
        upgrade_cache: yes
      when: ansible_distribution == "Ubuntu"

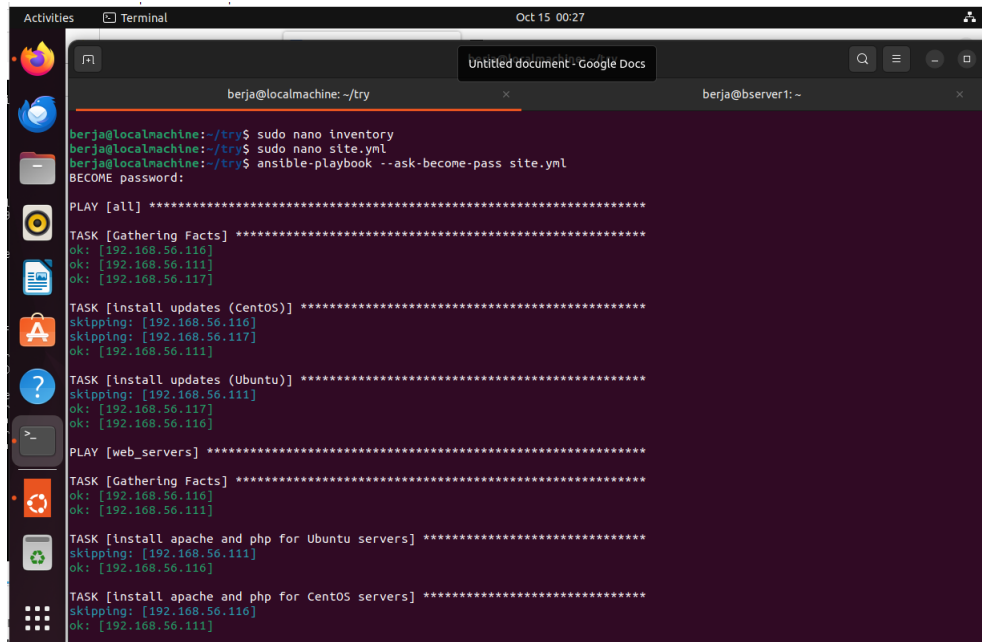
- hosts: web_servers
  become: true
  tasks:

    - 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
```

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.



```
berja@localmachine: ~/try
berja@localmachine:~/try$ sudo nano inventory
berja@localmachine:~/try$ sudo nano site.yml
berja@localmachine:~/try$ ansible-playbook --ask-become-pass site.yml
BECOME password:

PLAY [all] *****
TASK [Gathering Facts] *****
ok: [192.168.56.116]
ok: [192.168.56.111]
ok: [192.168.56.117]
TASK [Install updates (CentOS)] *****
skipping: [192.168.56.116]
skipping: [192.168.56.117]
ok: [192.168.56.111]
TASK [Install updates (Ubuntu)] *****
skipping: [192.168.56.111]
ok: [192.168.56.117]
ok: [192.168.56.116]
PLAY [web_servers] *****
TASK [Gathering Facts] *****
ok: [192.168.56.116]
ok: [192.168.56.111]
TASK [Install apache and php for Ubuntu servers] *****
skipping: [192.168.56.111]
ok: [192.168.56.116]
TASK [Install apache and php for CentOS servers] *****
skipping: [192.168.56.116]
ok: [192.168.56.111]
```

-In this part it showed the `site.yml` executed properly it updated the CentOS and Ubuntu. After that this install the apache and php for ubuntu then is install apache and php CentOS.

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.

```

berja@localmachine: ~/try
berja@bserver1: ~

ok: [192.168.56.117]
ok: [192.168.56.116]

PLAY [web_servers] *****
TASK [Gathering Facts] *****
ok: [192.168.56.116]
ok: [192.168.56.111]

TASK [install apache and php for Ubuntu servers] *****
skipping: [192.168.56.111]
ok: [192.168.56.116]

TASK [install apache and php for CentOS servers] *****
skipping: [192.168.56.116]
ok: [192.168.56.111]

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

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

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

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

PLAY RECAP *****
192.168.56.111      : ok=4    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
192.168.56.116      : ok=7    changed=1    unreachable=0    failed=0    skipped=3    rescued=0    ignored=0
192.168.56.117      : ok=2    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0

```

- In this part in the db\_servers it installs the mariadb package. After the installation this changed to restart and enable.

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.

```
berja@localmachine:~/tr$ systemctl status mariadb
● mariadb.service - MariaDB 10.6.12 database server
   Loaded: loaded (/lib/systemd/system/mariadb.service; enabled; vendor preset: enabled)
   Active: active (running) since Sun 2023-10-15 00:25:18 PST; 3min 30s ago
     Docs: man:mariadb(8)
           https://mariadb.com/kb/en/library/systemd/
   Process: 19436 ExecStartPre=/usr/bin/install -m 755 -o mysql -g root -d /var/run/mysql (code=exited, status=0/SUCCESS)
   Process: 19437 ExecStartPre=/bin/sh -c systemctl unset-environment _WSREP_START_POSITION (code=exited, status=0/SUCCESS)
   Process: 19439 ExecStartPre=/bin/sh -c [ ! -e /usr/bin/galera_recovery ] && VAR= || VAR=cd /usr/bin/..; /usr/bin/galera_re
   Process: 19479 ExecStartPost=/bin/sh -c systemctl unset-environment _WSREP_START_POSITION (code=exited, status=0/SUCCESS)
   Process: 19481 ExecStartPost=/etc/mysql/debian-start (code=exited, status=0/SUCCESS)
  Main PID: 19468 (mariabdd)
    Status: "Taking your SQL requests now..."
     Tasks: 10 (limit: 2262)
    Memory: 61.7M
       CPU: 325ms
    CGroup: /system.slice/mariadb.service
            └─19468 /usr/sbin/mariabdd

Oct 15 00:25:17 localmachine mariabdd[19468]: Version: '10.6.12-MariaDB-0ubuntu0.22.04.1' socket: '/run/mysql/mysql.sock' port:
Oct 15 00:25:18 localmachine systemd[1]: Started MariaDB 10.6.12 database server.
Oct 15 00:25:18 localmachine /etc/mysql/debian-start[19483]: Upgrading MySQL tables if necessary.
Oct 15 00:25:18 localmachine /etc/mysql/debian-start[19486]: Looking for 'mariadb' as: /usr/bin/mariadb
Oct 15 00:25:18 localmachine /etc/mysql/debian-start[19486]: Looking for 'mariadb-check' as: /usr/bin/mariadb-check
Oct 15 00:25:18 localmachine /etc/mysql/debian-start[19486]: This installation of MariaDB is already upgraded to 10.6.12-MariaDB.
Oct 15 00:25:18 localmachine /etc/mysql/debian-start[19486]: There is no need to run mysql_upgrade again for 10.6.12-MariaDB.
Oct 15 00:25:18 localmachine /etc/mysql/debian-start[19486]: You can use --force if you still want to run mysql_upgrade
Oct 15 00:25:18 localmachine /etc/mysql/debian-start[19499]: Checking for insecure root accounts.
Oct 15 00:25:18 localmachine /etc/mysql/debian-start[19504]: Triggering mysan-recover for all MyISAM tables and aria-recover for
```

```
[berja@CentOS ~]$ systemctl status mariadb
● mariadb.service - MariaDB database server
   Loaded: loaded (/usr/lib/systemd/system/mariadb.service; enabled; vendor preset: disabled)
   Active: active (running) since Sat 2023-10-14 13:55:57 EDT; 43s ago
     Process: 1248 ExecStartPost=/usr/libexec/mariadb-wait-ready $MAINPID (code=exited, status=0/SUCCESS)
     Process: 1155 ExecStartPre=/usr/libexec/mariadb-prepare-db-dir %n (code=exited, status=0/SUCCESS)
  Main PID: 1247 (mysqld_safe)
     Tasks: 20
    CGroup: /system.slice/mariadb.service
            └─1247 /bin/sh /usr/bin/mysqld_safe --basedir=/usr
              1616 /usr/libexec/mysqld --basedir=/usr --datadir=/var/lib/mysql --plug...

Oct 14 13:55:55 CentOS systemd[1]: Starting MariaDB database server...
Oct 14 13:55:55 CentOS mariadb-prepare-db-dir[1155]: Database MariaDB is probably i...
Oct 14 13:55:55 CentOS mariadb-prepare-db-dir[1155]: If this is not the case, make ....
Oct 14 13:55:55 CentOS mysqld_safe[1247]: 231014 13:55:55 mysqld_safe Logging to '...'.
Oct 14 13:55:55 CentOS mysqld_safe[1247]: 231014 13:55:55 mysqld_safe Starting mys...ql
Oct 14 13:55:57 CentOS systemd[1]: Started MariaDB database server.
Hint: Some lines were ellipsized, use -l to show in full.
[berja@CentOS ~]$
```

- The output code shows that the installation shows it is successful running for both nodes.

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.

```

berja@localmachine: ~/try
ok: [192.168.56.116]

TASK [Install apache and php for Ubuntu servers] *****
skipping: [192.168.56.111]
ok: [192.168.56.116]

TASK [Install apache and php for CentOS servers] *****
skipping: [192.168.56.116]
ok: [192.168.56.111]

PLAY [db_servers] *****

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

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

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

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

PLAY [file_servers] *****

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

TASK [Install samba package] *****
changed: [192.168.56.116]

PLAY RECAP *****
192.168.56.111      : ok=4    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
192.168.56.116      : ok=9    changed=2    unreachable=0    failed=0    skipped=3    rescued=0    ignored=0

```

- The samba package installation successfully changed in the server. 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.

```

berja@localmachine:~/try$ ansible-playbook --ask-become-pass site.yml
BECOME password:

PLAY [all] *****

TASK [Gathering Facts] *****
ok: [192.168.56.116]
ok: [192.168.56.111]

TASK [install updates (CentOS)] *****
skipping: [192.168.56.116]
ok: [192.168.56.111]

TASK [install updates (Ubuntu)] *****
skipping: [192.168.56.111]
ok: [192.168.56.116]

PLAY [web_servers] *****

TASK [Gathering Facts] *****
ok: [192.168.56.116]
ok: [192.168.56.111]

TASK [install apache and php for Ubuntu servers] *****
skipping: [192.168.56.111]
ok: [192.168.56.116]

TASK [install apache and php for CentOS servers] *****
skipping: [192.168.56.116]
ok: [192.168.56.111]

PLAY [db_servers] *****

TASK [Gathering Facts] *****
ok: [192.168.56.111]

TASK [install apache and php for Ubuntu servers] *****
skipping: [192.168.56.111]
ok: [192.168.56.116]

TASK [install apache and php for CentOS servers] *****
skipping: [192.168.56.116]
ok: [192.168.56.111]

PLAY [db_servers] *****

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

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

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

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

PLAY [file_servers] *****

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

TASK [install samba package] *****
ok: [192.168.56.116]

PLAY RECAP *****
192.168.56.111      : ok=4    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
192.168.56.116      : ok=9    changed=1    unreachable=0    failed=0    skipped=3    rescued=0    ignored=0

berja@localmachine:~/try$

```

- After the running code the output shows all of the package are a success no error shows
2. On the local machine, try to issue the following commands and describe each result:
    - 2.1 *ansible-playbook --list-tags site.yml*

```
berja@localmachine:~/try$ 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]
berja@localmachine:~/try$
```

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

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

PLAY [all] *****

TASK [Gathering Facts] *****
ok: [192.168.56.116]
ok: [192.168.56.111]

TASK [install updates (CentOS)] *****
skipping: [192.168.56.116]
ok: [192.168.56.111]

TASK [install updates (Ubuntu)] *****
skipping: [192.168.56.111]
ok: [192.168.56.116]

PLAY [web_servers] *****

TASK [Gathering Facts] *****
ok: [192.168.56.116]
ok: [192.168.56.111]

TASK [install apache and php for CentOS servers] *****
skipping: [192.168.56.116]
ok: [192.168.56.111]

PLAY [db_servers] *****

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

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

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

```

TASK [install updates (CentOS)] *****
skipping: [192.168.56.116]
ok: [192.168.56.111]

TASK [install updates (Ubuntu)] *****
skipping: [192.168.56.111]
ok: [192.168.56.116]

PLAY [web_servers] *****

TASK [Gathering Facts] *****
ok: [192.168.56.116]
ok: [192.168.56.111]

TASK [install apache and php for CentOS servers] *****
skipping: [192.168.56.116]
ok: [192.168.56.111]

PLAY [db_servers] *****

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

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

PLAY [file_servers] *****

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

PLAY RECAP *****
192.168.56.111      : ok=4   changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
192.168.56.116      : ok=5   changed=0    unreachable=0    failed=0    skipped=3    rescued=0    ignored=0
berja@localmachine: ~/try$

```

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



```

TASK [install updates (CentOS)] *****
skipping: [192.168.56.116]
ok: [192.168.56.111]

TASK [install updates (Ubuntu)] *****
skipping: [192.168.56.111]
ok: [192.168.56.116]

PLAY [web_servers] *****

TASK [Gathering Facts] *****
ok: [192.168.56.116]
ok: [192.168.56.111]

PLAY [db_servers] *****

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

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

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

PLAY [file_servers] *****

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

PLAY RECAP *****
192.168.56.111      : ok=3   changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
192.168.56.116      : ok=6   changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
berja@localmachine: ~/try$

```

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

```
Activities Terminal Oct 15 01:06
berja@localmachine: ~/try
berja@localmachine: ~/try
berja@bserver1: ~
TASK [install updates (CentOS)] *****
skipping: [192.168.56.116]
ok: [192.168.56.111]
TASK [install updates (Ubuntu)] *****
skipping: [192.168.56.111]
ok: [192.168.56.116]
PLAY [web_servers] *****
TASK [Gathering Facts] *****
ok: [192.168.56.111]
ok: [192.168.56.116]
TASK [install apache and php for Ubuntu servers] *****
skipping: [192.168.56.111]
ok: [192.168.56.116]
TASK [install apache and php for CentOS servers] *****
skipping: [192.168.56.116]
ok: [192.168.56.111]
PLAY [db_servers] *****
TASK [Gathering Facts] *****
ok: [192.168.56.116]
PLAY [file_servers] *****
TASK [Gathering Facts] *****
ok: [192.168.56.116]
PLAY RECAP *****
192.168.56.111 : ok=4 changed=0 unreachable=0 failed=0 skipped=2 rescued=0 ignored=0
192.168.56.116 : ok=6 changed=0 unreachable=0 failed=0 skipped=2 rescued=0 ignored=0
berja@localmachine: ~/try$
```

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

```
Activities Terminal Oct 15 01:08
berja@localmachine: ~/try
berja@localmachine: ~/try
berja@bserver1: ~
ok: [192.168.56.116]
PLAY [web_servers] *****
TASK [Gathering Facts] *****
ok: [192.168.56.116]
ok: [192.168.56.111]
TASK [install apache and php for Ubuntu servers] *****
skipping: [192.168.56.111]
ok: [192.168.56.116]
TASK [install apache and php for CentOS servers] *****
skipping: [192.168.56.116]
ok: [192.168.56.111]
PLAY [db_servers] *****
TASK [Gathering Facts] *****
ok: [192.168.56.116]
TASK [install mariadb package (CentOS)] *****
skipping: [192.168.56.116]
TASK [install mariadb package (Ubuntu)] *****
ok: [192.168.56.116]
PLAY [file_servers] *****
TASK [Gathering Facts] *****
ok: [192.168.56.116]
PLAY RECAP *****
192.168.56.111 : ok=4 changed=0 unreachable=0 failed=0 skipped=2 rescued=0 ignored=0
192.168.56.116 : ok=7 changed=0 unreachable=0 failed=0 skipped=3 rescued=0 ignored=0
berja@localmachine: ~/try$
```

## 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.

```

Oct 15 01:13
berja@localmachine: ~
berja@localmachine: ~/try
ok: [192.168.56.116]
TASK [install apache and php for CentOS servers] *****
skipping: [192.168.56.116]
ok: [192.168.56.111]

TASK [start httpd (CentOS)] *****
skipping: [192.168.56.116]
changed: [192.168.56.111]

PLAY [db_servers] *****

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

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

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

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

PLAY [file_servers] *****

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

TASK [install samba package] *****
ok: [192.168.56.116]

PLAY RECAP *****
192.168.56.111      : ok=5    changed=1    unreachable=0    failed=0    skipped=2    rescued=0
192.168.56.116    : ok=9    changed=1    unreachable=0    failed=0    skipped=4    rescued=0
berja@localmachine:~/try$

```

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"

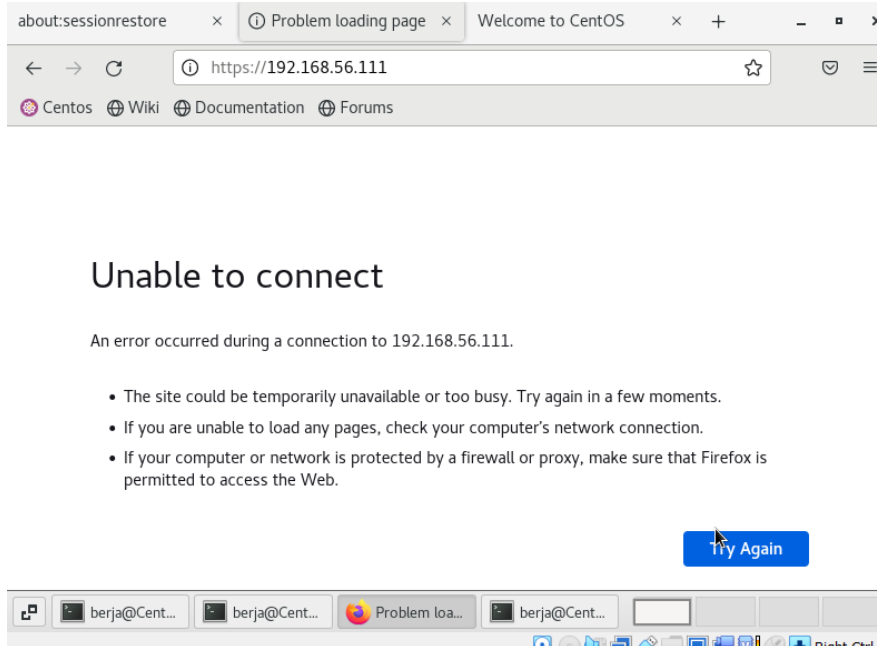
  - name: "Mariadb- Restarting/Enabling"
    service:
      name: mariadb
      state: restarted
      enabled: true
```

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.

```
[berja@CentOS ~]$ sudo systemctl stop httpd
[sudo] password for berjа:
[berja@CentOS ~]$ sudo systemctl stop httpd
[berja@CentOS ~]$
```



- 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.

```

berja@localmachine: ~/try
ok: [192.168.56.116]

TASK [install apache and php for CentOS servers] *****
skipping: [192.168.56.116]
ok: [192.168.56.111]

TASK [start httpd (CentOS)] *****
skipping: [192.168.56.116]
changed: [192.168.56.111]

PLAY [db_servers] *****

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

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

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

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

PLAY [file_servers] *****

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

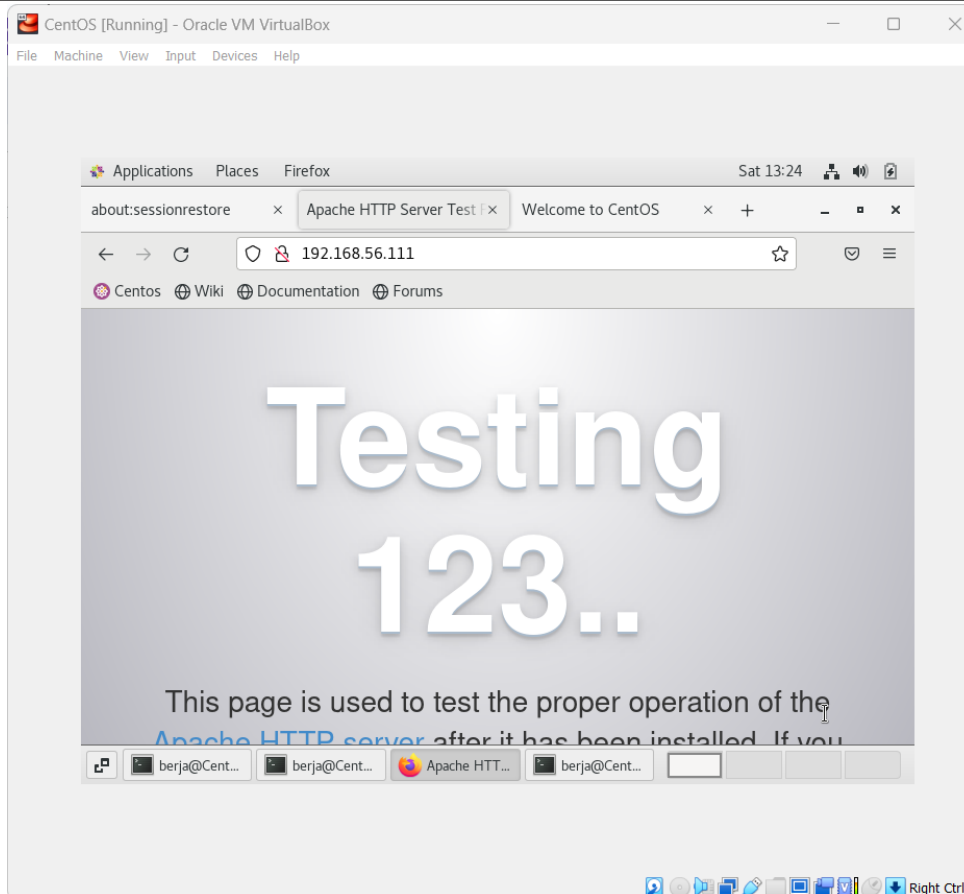
TASK [install samba package] *****
ok: [192.168.56.116]

PLAY RECAP *****
192.168.56.111      : ok=5    changed=1    unreachable=0    failed=0    skipped=2    rescued=0
192.168.56.116    : ok=9    changed=1    unreachable=0    failed=0    skipped=4    rescued=0

berja@localmachine:~/try$

```

To automatically enable the service every time we run the playbook, use the command *enabled: true* similar to Figure 7.1.2 and save the playbook.



## Reflections:

Answer the following:

1. What is the importance of putting our remote servers into groups?
  - Grouping remote servers will help more on security and be much more efficient on working this into groups, this simplifies control. This group helps on checking the servers and maintenance, as group servers would have more efficient documentation enabling my performance as a User to enhance more efficient and control
2. What is the importance of tags in playbooks?
  - The tags in playbooks are commonly known for Ansible. First, this help the user to run specific tasks and roles inside the playbook. Tags are very useful in a way that when the user can run many tasks. Second, the tags are beneficial when the user has multiple roles in the playbook. It can use tags to execute only particular tasks. The playbooks tags provide more control and clarity in managing automation tasks.

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

- For this reason is the consistency. Automation helps us know that the services are managed consistently across to other servers that is provided. Error reduction are one of the reasons why due to the fact that it can include error-checking and reduce the risk of misconfig.