Name: Niemo, Christian Al C.	Date Performed: 9/29/23
Course/Section: CPE232 - CPE31S6	Date Submitted: 10/2/23
Instructor: Dr. Jonathan Taylar	Semester and SY: 1st sem - 2023-2024

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"
```

GNU nano 2.9.3   site.yml

```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"
```

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 2.9.3                                              inventory

[web_servers]
192.168.56.111
192.168.56.109

[db_servers]
192.168.56.112
192.168.56.109

[file_servers]
192.168.56.111
```

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.

```
workstation@workstation:~/hoa6$ sudo nano site.yml
workstation@workstation:~/hoa6$ ansible-playbook --ask-become-pass site.yml
BECOME password:

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

TASK [Gathering Facts] ********************************************************************
ok: [192.168.56.112]
ok: [192.168.56.111]
ok: [192.168.56.109]

TASK [install apache and php for Ubuntu servers] ****************************************
skipping: [192.168.56.109]
ok: [192.168.56.112]
ok: [192.168.56.111]

TASK [install apache and php for CentOS servers] ****************************************
skipping: [192.168.56.111]
skipping: [192.168.56.112]
ok: [192.168.56.109]

PLAY RECAP *********************************************************************************
192.168.56.109             : ok=2    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
192.168.56.111             : ok=2    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
192.168.56.112             : ok=2    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
```

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

```yaml
---

- 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.

```
  GNU nano 2.9.3                                              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.

```
workstation@workstation:~/hoa6$ ansible-playbook --ask-become-pass site.yml
BECOME password:

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

TASK [Gathering Facts] **********************************************************
ok: [192.168.56.112]
ok: [192.168.56.111]
ok: [192.168.56.109]

TASK [install updates (CentOS)] *************************************************
skipping: [192.168.56.111]
skipping: [192.168.56.112]
ok: [192.168.56.109]

TASK [install updates (Ubuntu)] *************************************************
skipping: [192.168.56.109]
ok: [192.168.56.112]
ok: [192.168.56.111]

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

TASK [Gathering Facts] **********************************************************
ok: [192.168.56.111]
ok: [192.168.56.109]

TASK [install apache and php for Ubuntu servers] *******************************
skipping: [192.168.56.109]
ok: [192.168.56.111]

TASK [install apache and php for CentOS servers] *******************************
skipping: [192.168.56.111]
ok: [192.168.56.109]

PLAY RECAP **********************************************************************
192.168.56.109             : ok=4    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
192.168.56.111             : ok=4    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
192.168.56.112             : ok=2    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
```

- It run successfully.
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.

```yaml
- 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.

```
GNU nano 2.9.3                                                    site.yml

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

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

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

```
File  Edit  View  Search  Terminal  Help
TASK [install updates (CentOS)] ********************************************************
skipping: [192.168.56.111]
skipping: [192.168.56.112]
ok: [192.168.56.109]

TASK [install updates (Ubuntu)] ********************************************************
skipping: [192.168.56.109]
ok: [192.168.56.112]
ok: [192.168.56.111]

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

TASK [Gathering Facts] *****************************************************************
ok: [192.168.56.111]
ok: [192.168.56.109]

TASK [install apache and php for Ubuntu servers] ***************************************
skipping: [192.168.56.109]
ok: [192.168.56.111]

TASK [install apache and php for CentOS servers] ***************************************
skipping: [192.168.56.111]
ok: [192.168.56.109]

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

TASK [Gathering Facts] *****************************************************************
ok: [192.168.56.112]
ok: [192.168.56.109]

TASK [install mariadb package (CentOS)] ************************************************
skipping: [192.168.56.112]
ok: [192.168.56.109]

TASK [install mariadb package (Ubuntu)] ************************************************
skipping: [192.168.56.109]
changed: [192.168.56.112]

TASK [Mariadb- Restarting/Enabling] ***************************************************
changed: [192.168.56.112]
changed: [192.168.56.109]

PLAY RECAP *****************************************************************************
192.168.56.109             : ok=7    changed=1    unreachable=0    failed=0    skipped=3    rescued=0    ignored=0
192.168.56.111             : ok=4    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
192.168.56.112             : ok=5    changed=2    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
```

- Just like the previous one, it smoothly run without error.

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.

```
● mariadb.service - MariaDB 10.1.48 database server
   Loaded: loaded (/lib/systemd/system/mariadb.service; enabled; vendor preset:
   Active: active (running) since Mon 2023-10-02 17:07:23 PST; 54s ago
     Docs: man:mysqld(8)
           https://mariadb.com/kb/en/library/systemd/
  Process: 12508 ExecStartPost=/bin/sh -c systemctl unset-environment _WSREP_STA
  Process: 12505 ExecStartPost=/etc/mysql/debian-start (code=exited, status=0/SU
  Process: 12404 ExecStartPre=/bin/sh -c [ ! -e /usr/bin/galera_recovery ] && VA
  Process: 12402 ExecStartPre=/bin/sh -c systemctl unset-environment _WSREP_STAR
  Process: 12401 ExecStartPre=/usr/bin/install -m 755 -o mysql -g root -d /var/r
 Main PID: 12478 (mysqld)
   Status: "Taking your SQL requests now..."
    Tasks: 27 (limit: 4915)
   CGroup: /system.slice/mariadb.service
           └─12478 /usr/sbin/mysqld

Oct 02 17:07:22 server2 systemd[1]: Starting MariaDB 10.1.48 database server...
Oct 02 17:07:23 server2 mysqld[12478]: 2023-10-02 17:07:23 139901552213120 [Note
Oct 02 17:07:23 server2 systemd[1]: Started MariaDB 10.1.48 database server.
Oct 02 17:07:23 server2 /etc/mysql/debian-start[12511]: /usr/bin/mysql_upgrade:
Oct 02 17:07:23 server2 /etc/mysql/debian-start[12511]: Looking for 'mysql' as:
Oct 02 17:07:23 server2 /etc/mysql/debian-start[12511]: Looking for 'mysqlcheck'
Oct 02 17:07:23 server2 /etc/mysql/debian-start[12511]: This installation of MyS
lines 1-23
```



```
                          workstation@localhost:~              _  □  ✕

File  Edit  View  Search  Terminal  Help
[workstation@localhost ~]$ systemctl status mariadb
● mariadb.service - MariaDB database server
   Loaded: loaded (/usr/lib/systemd/system/mariadb.service; enabled; vendor pres
et: disabled)
   Active: active (running) since Mon 2023-10-02 05:07:34 EDT; 1min 15s ago
  Process: 9927 ExecStartPost=/usr/libexec/mariadb-wait-ready $MAINPID (code=exi
ted, status=0/SUCCESS)
  Process: 9892 ExecStartPre=/usr/libexec/mariadb-prepare-db-dir %n (code=exited
, status=0/SUCCESS)
 Main PID: 9926 (mysqld_safe)
    Tasks: 20
   CGroup: /system.slice/mariadb.service
           ├─ 9926 /bin/sh /usr/bin/mysqld_safe --basedir=/usr
           └─10091 /usr/libexec/mysqld --basedir=/usr --datadir=/var/lib/mysq...

Oct 02 05:07:31 localhost.localdomain systemd[1]: Starting MariaDB database s...
Oct 02 05:07:31 localhost.localdomain mariadb-prepare-db-dir[9892]: Database ...
Oct 02 05:07:31 localhost.localdomain mariadb-prepare-db-dir[9892]: If this i...
Oct 02 05:07:32 localhost.localdomain mysqld_safe[9926]: 231002 05:07:32 mysq...
Oct 02 05:07:32 localhost.localdomain mysqld_safe[9926]: 231002 05:07:32 mysq...
Oct 02 05:07:34 localhost.localdomain systemd[1]: Started MariaDB database se...
Hint: Some lines were ellipsized, use -l to show in full.
[workstation@localhost ~]$
```

- The mariadb is successfully installed, so after that it runs automatically.

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: file_servers
  become: true
  tasks:

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

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

```
ok: [192.168.56.112]

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

TASK [Gathering Facts] *******************************************************************
ok: [192.168.56.111]
ok: [192.168.56.109]

TASK [install apache and php for Ubuntu servers] ****************************************
skipping: [192.168.56.109]
ok: [192.168.56.111]

TASK [install apache and php for CentOS servers] ****************************************
skipping: [192.168.56.111]
ok: [192.168.56.109]

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

TASK [Gathering Facts] *******************************************************************
ok: [192.168.56.112]
ok: [192.168.56.109]

TASK [install mariadb package (CentOS)] ************************************************
skipping: [192.168.56.112]
ok: [192.168.56.109]

TASK [install mariadb package (Ubuntu)] ************************************************
skipping: [192.168.56.109]
ok: [192.168.56.112]

TASK [Mariadb- Restarting/Enabling] ***************************************************
changed: [192.168.56.112]
changed: [192.168.56.109]

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

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

TASK [install samba package] **********************************************************
changed: [192.168.56.111]

PLAY RECAP *******************************************************************************
192.168.56.109             : ok=7    changed=1    unreachable=0    failed=0    skipped=3    rescued=0    ignored=0
192.168.56.111             : ok=6    changed=1    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
192.168.56.112             : ok=5    changed=1    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
```

- The samba package has been successfully installed.

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"


- 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"
```

```yaml
- 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
```

```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"
```

```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"
```

```
- 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: 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.

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

TASK [Gathering Facts] *************************************************************************
ok: [192.168.56.111]
ok: [192.168.56.109]

TASK [install apache and php for Ubuntu servers] **********************************************
skipping: [192.168.56.109]
ok: [192.168.56.111]

TASK [install apache and php for CentOS servers] **********************************************
skipping: [192.168.56.111]
ok: [192.168.56.109]

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

TASK [Gathering Facts] *************************************************************************
ok: [192.168.56.112]
ok: [192.168.56.109]

TASK [install mariadb package (CentOS)] *******************************************************
skipping: [192.168.56.112]
ok: [192.168.56.109]

TASK [install mariadb package (Ubuntu)] *******************************************************
skipping: [192.168.56.109]
ok: [192.168.56.112]

TASK [Mariadb- Restarting/Enabling] ***********************************************************
changed: [192.168.56.112]
changed: [192.168.56.109]

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

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

TASK [install samba package] ******************************************************************
ok: [192.168.56.111]

PLAY RECAP ************************************************************************************
192.168.56.109             : ok=7    changed=1    unreachable=0    failed=0    skipped=3    rescued=0    ignored=0
192.168.56.111             : ok=6    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
192.168.56.112             : ok=5    changed=1    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
```

- The output is the same just like before.

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

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

```
workstation@workstation:~/hoa6$ 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: [Ubuntu, apache, apache2, centos, httpd]

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

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

- *It shows the tags that we input after the name of the package*

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

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

TASK [Gathering Facts] *********************************************************
ok: [192.168.56.111]
ok: [192.168.56.112]
ok: [192.168.56.109]

TASK [install updates (CentOS)] ************************************************
skipping: [192.168.56.111]
skipping: [192.168.56.112]
ok: [192.168.56.109]

TASK [install updates (Ubuntu)] ************************************************
skipping: [192.168.56.109]
ok: [192.168.56.111]
ok: [192.168.56.112]

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

TASK [Gathering Facts] *********************************************************
ok: [192.168.56.111]
ok: [192.168.56.109]

TASK [install apache and php for CentOS servers] ******************************
skipping: [192.168.56.111]
ok: [192.168.56.109]

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

TASK [Gathering Facts] *********************************************************
ok: [192.168.56.112]
ok: [192.168.56.109]

TASK [install mariadb package (CentOS)] ***************************************
skipping: [192.168.56.112]
ok: [192.168.56.109]

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

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

PLAY RECAP ********************************************************************
192.168.56.109             : ok=6    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
192.168.56.111             : ok=4    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
192.168.56.112             : ok=3    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
```

- *The output is the same just like before*

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

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

TASK [Gathering Facts] *********************************************************
ok: [192.168.56.111]
ok: [192.168.56.112]
ok: [192.168.56.109]

TASK [install updates (CentOS)] ************************************************
skipping: [192.168.56.111]
skipping: [192.168.56.112]
ok: [192.168.56.109]

TASK [install updates (Ubuntu)] ************************************************
skipping: [192.168.56.109]
ok: [192.168.56.112]
ok: [192.168.56.111]

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

TASK [Gathering Facts] *********************************************************
ok: [192.168.56.111]
ok: [192.168.56.109]

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

TASK [Gathering Facts] *********************************************************
ok: [192.168.56.112]
ok: [192.168.56.109]

TASK [install mariadb package (CentOS)] ***************************************
skipping: [192.168.56.112]
ok: [192.168.56.109]

TASK [install mariadb package (Ubuntu)] ***************************************
skipping: [192.168.56.109]
ok: [192.168.56.112]

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

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

PLAY RECAP ********************************************************************
192.168.56.109             : ok=5    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
192.168.56.111             : ok=4    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
192.168.56.112             : ok=4    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
```

- *Nothing new on the output*

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

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

TASK [Gathering Facts] *********************************************************
ok: [192.168.56.111]
ok: [192.168.56.112]
ok: [192.168.56.109]

TASK [install updates (CentOS)] ***********************************************
skipping: [192.168.56.111]
skipping: [192.168.56.112]
ok: [192.168.56.109]

TASK [install updates (Ubuntu)] ***********************************************
skipping: [192.168.56.109]
ok: [192.168.56.112]
ok: [192.168.56.111]

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

TASK [Gathering Facts] *********************************************************
ok: [192.168.56.111]
ok: [192.168.56.109]

TASK [install apache and php for Ubuntu servers] ******************************
skipping: [192.168.56.109]
ok: [192.168.56.111]

TASK [install apache and php for CentOS servers] ******************************
skipping: [192.168.56.111]
ok: [192.168.56.109]

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

TASK [Gathering Facts] *********************************************************
ok: [192.168.56.112]
ok: [192.168.56.109]

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

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

PLAY RECAP ********************************************************************
192.168.56.109    : ok=5    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
192.168.56.111    : ok=5    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
192.168.56.112    : ok=3    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
```

- *Same just like before*

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

```
skipping: [192.168.56.111]
skipping: [192.168.56.112]
ok: [192.168.56.109]

TASK [install updates (Ubuntu)] ***********************************************
skipping: [192.168.56.109]
ok: [192.168.56.112]
ok: [192.168.56.111]

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

TASK [Gathering Facts] *********************************************************
ok: [192.168.56.111]
ok: [192.168.56.109]

TASK [install apache and php for Ubuntu servers] ******************************
skipping: [192.168.56.109]
ok: [192.168.56.111]

TASK [install apache and php for CentOS servers] ******************************
skipping: [192.168.56.111]
ok: [192.168.56.109]

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

TASK [Gathering Facts] *********************************************************
ok: [192.168.56.112]
ok: [192.168.56.109]

TASK [install mariadb package (CentOS)] ***************************************
skipping: [192.168.56.112]
ok: [192.168.56.109]

TASK [install mariadb package (Ubuntu)] ***************************************
skipping: [192.168.56.109]
ok: [192.168.56.112]

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

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

PLAY RECAP ********************************************************************
192.168.56.109    : ok=6    changed=0    unreachable=0    failed=0    skipped=3    rescued=0    ignored=0
192.168.56.111    : ok=5    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
192.168.56.112    : ok=4    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
```

- *Same from all outputs because of the –ask-become-pass command.*

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

```
- 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

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.

```
[workstation@localhost ~]$ sudo systemctl stop httpd
[sudo] password for workstation:
[workstation@localhost ~]$
```
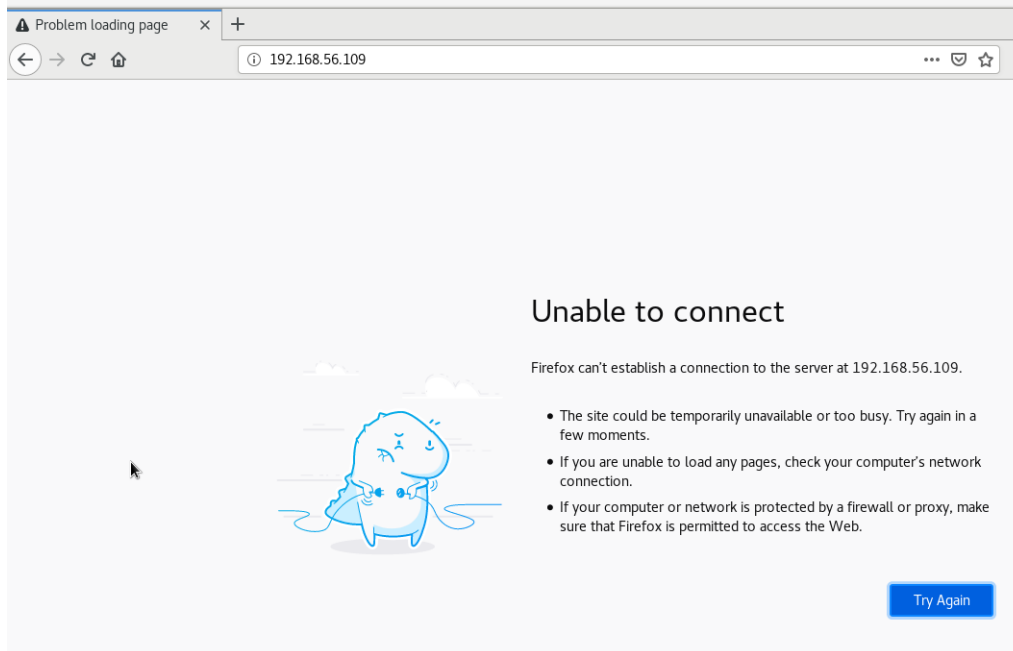
3. Go to the local machine and this time, run the *site.yml* file. Then after running the file, go again to the CentOS server and enter its IP address on the browser. Describe the result.


```
TASK [Gathering Facts] ***********************************************************
ok: [192.168.56.111]
ok: [192.168.56.109]

TASK [install apache and php for Ubuntu servers] *********************************
skipping: [192.168.56.109]
ok: [192.168.56.111]

TASK [install apache and php for CentOS servers] *********************************
skipping: [192.168.56.111]
ok: [192.168.56.109]

TASK [start httpd (CentOS)] ******************************************************
skipping: [192.168.56.111]
changed: [192.168.56.109]

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

TASK [Gathering Facts] ***********************************************************
ok: [192.168.56.112]
ok: [192.168.56.109]

TASK [install mariadb package (CentOS)] ******************************************
skipping: [192.168.56.112]
ok: [192.168.56.109]

TASK [install mariadb package (Ubuntu)] ******************************************
skipping: [192.168.56.109]
ok: [192.168.56.112]

TASK [Mariadb- Restarting/Enabling] **********************************************
changed: [192.168.56.109]
changed: [192.168.56.112]

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

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

TASK [install samba package] *****************************************************
ok: [192.168.56.111]

PLAY RECAP ***********************************************************************
192.168.56.109             : ok=8    changed=2    unreachable=0    failed=0    skipped=3    rescued=0    ignored=0
192.168.56.111             : ok=6    changed=0    unreachable=0    failed=0    skipped=3    rescued=0    ignored=0
192.168.56.112             : ok=5    changed=1    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
```
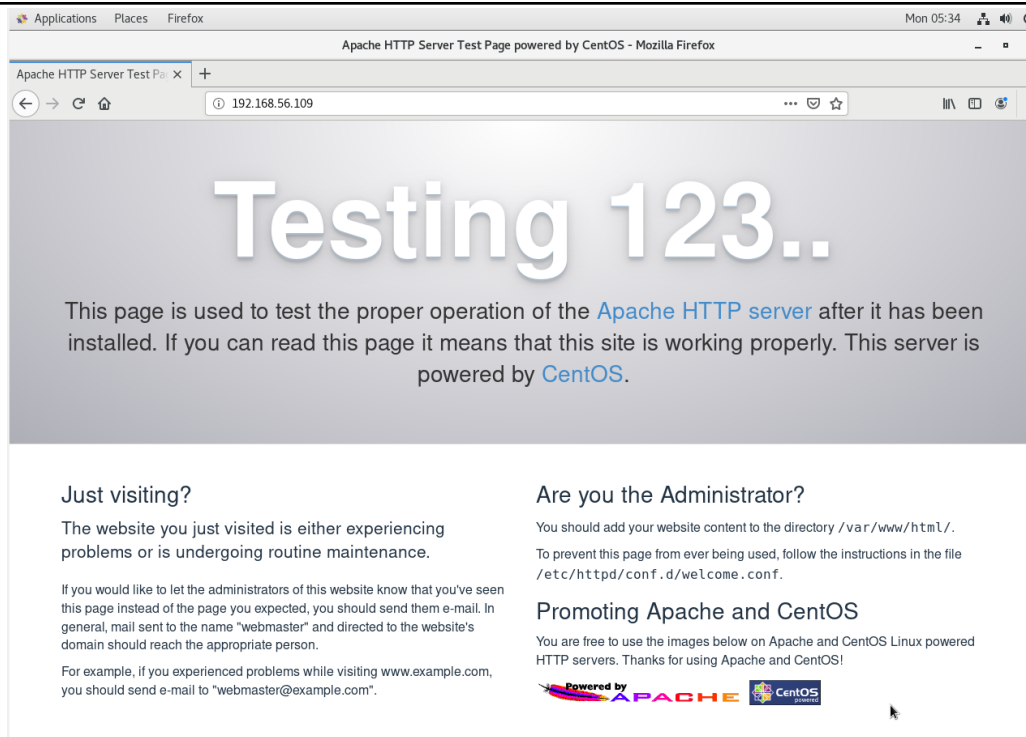
- The httpd successfully runs, just like before

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"
```

Reflections:

Answer the following:

1. What is the importance of putting our remote servers into groups?
   - Grouping remote servers together is important for several reasons. Firstly, it allows system administrators to manage and monitor the corporate network remotely, making it easier to perform tasks on multiple servers at once. Secondly, it enables faster troubleshooting, as technicians can more quickly identify and troubleshoot problems that may occur. Thirdly, grouping servers together can improve security by applying security policies to groups of

servers, ensuring that all servers in the group have the same level of security. Lastly, remote server access can increase productivity by allowing system administrators to respond to issues more quickly, resulting in faster and more responsive services for customers.

2. What is the importance of tags in playbooks?
   - Tags are metadata that can be attached to tasks in an Ansible playbook, allowing for selective targeting of certain tasks at runtime. Adding tags to a play or statically imported tasks and roles adds those tags to all of the contained tasks, referred to as tag inheritance. Tags can be added to a single task or include, or to multiple tasks by defining them at the level of a block, play, role, or import. Tags can be used to select or skip tasks when running a playbook, making it possible to run specific tasks within a playbook on demand. Understanding how tags are passed within the hierarchy of imports is important because tags become tricky when using includes or imports. Tags can also be used to filter roles in a playbook, allowing for the selective running of one or two roles out of many. Overall, tags are important in playbooks because they allow for more efficient and selective execution of tasks.

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

   - Managing services automatically in playbooks can provide several benefits, including increased efficiency, faster response times, improved customer satisfaction, operational excellence, and scalability, as automated playbooks can detect new threats faster, reduce the need for manual intervention, and provide self-service options for users.

Conclusion:

In conclusion, individualizing hosts, applying tags in selecting plays to run, and managing services from remote servers using playbooks are important objectives that can be achieved through the use of automation, which can provide benefits such as increased efficiency, faster response times, improved customer satisfaction, operational excellence, and scalability