Name: Jefferson Langbid	Date Performed:
Course/Section: CPE232	Date Submitted:
Instructor: Dr. Taylar	Semester and SY:

Activity 6: Targeting Specific Nodes and Managing Services

1. **Objectives:**

1.1 Individualize hosts
1.2 Apply tags in selecting plays to run
1.3 Managing Services from remote servers using playbooks

2. **Discussion**:

In this activity, we try to individualize hosts. For example, we don't want apache on all our servers, or maybe only one of our servers is a web server, or maybe we have different servers like database or file servers running different things on different categories of servers and that is what we are going to take a look at in this activity.

We also try to manage services that do not automatically run using the automations in playbook. For example, when we install web servers or httpd for CentOS, we notice that the service did not start automatically.

Requirement:
In this activity, you will need to create another Ubuntu VM and name it Server 3. Likewise, you need to activate the second adapter to a host-only adapter after the installations. Take note of the IP address of the Server 3. Make sure to use the command *ssh-copy-id* to copy the public key to Server 3. Verify if you can successfully SSH to Server 3.

Task 1: Targeting Specific Nodes

1. Create a new playbook and named it site.yml. Follow the commands as shown in the image below. Make sure to save the file and exit.

```
---

- hosts: all
  become: true
  tasks:

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

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

jefferson@LocalMachine: ~/Langbid_PrelimExam

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
[remote_servers]
192.168.56.105
192.168.56.106

[web_server]
192.168.56.105
192.168.56.106

[db_server]
192.168.56.105

[file_server]
192.168.56.106
```

```
jefferson@LocalMachine:~/Langbid_PrelimExam$ ansible -m ping all
192.168.56.105 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
192.168.56.106 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": false,
    "ping": "pong"
}
```

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.

```
jefferson@LocalMachine: ~/Langbid_PrelimExam
GNU nano 6.2                    site.yml
---

- hosts: all
  become: true
  pre_tasks:

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

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

- hosts: web_servers
  become: true
  tasks:

  - name: install apache and php for Ubuntu servers
    apt:
      name:
                                [ Read 39 lines ]
```

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.

```
jefferson@LocalMachine: ~/Langbid_PrelimExam
skipping: [192.168.56.106]
ok: [192.168.56.105]

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

TASK [Gathering Facts] ********************************************************
*
ok: [192.168.56.106]
ok: [192.168.56.105]

TASK [install apache and php for Ubuntu servers] ******************************
*
skipping: [192.168.56.106]
ok: [192.168.56.105]

TASK [install apache and php for CentOS servers] ******************************
*
skipping: [192.168.56.105]
ok: [192.168.56.106]

PLAY RECAP ********************************************************************
*
192.168.56.105             : ok=4    changed=0    unreachable=0    failed=0
skipped=2    rescued=0    ignored=0
192.168.56.106             : ok=4    changed=0    unreachable=0    failed=0
skipped=2    rescued=0    ignored=0

jefferson@LocalMachine:~/Langbid_PrelimExam$
```

The site.yml run successfully and it updated the devices that are connected to ansible.

4. Let's try to edit again the *site.yml* file. This time, we are going to add plays targeting the other servers. This time we target the *db_servers* by adding it on the current *site.yml*. Below is an example: (Note add this at the end of the playbooks from task 1.3.

```
- hosts: db_servers
  become: true
  tasks:

  - name: install mariadb package (CentOS)
    yum:
      name: mariadb-server
      state: latest
    when: ansible_distribution == "CentOS"

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

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

Make sure to save the file and exit.

```
jefferson@LocalMachine: ~/Langbid_PrelimExam
GNU nano 6.2                    site.yml *
        - php
      state: latest
    when: ansible_distribution == "CentOS"

- hosts: db_servers
  become: true
  tasks:

  - name: install mariadb package (CentOS)
    yum:
      name: mariadb-server
      state: latest
    when: ansible_distribution == "CentOS"

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

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

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

```
jefferson@LocalMachine: ~/Langbid_PrelimExam

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

TASK [Gathering Facts] *********************************************************
ok: [192.168.56.105]
ok: [192.168.56.106]

TASK [install apache and php for Ubuntu servers] *******************************
skipping: [192.168.56.106]
ok: [192.168.56.105]

TASK [install apache and php for CentOS servers] *******************************
skipping: [192.168.56.105]
ok: [192.168.56.106]

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

TASK [Gathering Facts] *********************************************************
ok: [192.168.56.105]

TASK [install mariadb package (CentOS)] ****************************************
skipping: [192.168.56.105]

TASK [Mariadb- Restarting/Enabling] ********************************************
changed: [192.168.56.105]

TASK [install mariadb package (Ubuntu)] ****************************************
ok: [192.168.56.105]

PLAY RECAP *********************************************************************
192.168.56.105             : ok=7    changed=1    unreachable=0    failed=0    skipped=3    rescued=0
    ignored=0
192.168.56.106             : ok=4    changed=0    unreachable=0    failed=0    skipped=2    rescued=0
    ignored=0
```

The command is successfully running and it restarting/enabling is changed that proves that is successful and it skipped in the centos because I used the ubuntu

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.

```
jefferson@Server1: ~

Setting up mariadb-server (1:10.6.7-2ubuntu1.1) ...
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for libc-bin (2.35-0ubuntu3.1) ...
jefferson@Server1:~$ systemctl status mariadb
● mariadb.service - MariaDB 10.6.7 database server
     Loaded: loaded (/lib/systemd/system/mariadb.service; enabled; vendor pres>
     Active: active (running) since Sat 2022-10-08 16:37:42 PST; 3min 44s ago
       Docs: man:mariadbd(8)
             https://mariadb.com/kb/en/library/systemd/
    Process: 36808 ExecStartPre=/usr/bin/install -m 755 -o mysql -g root -d /v>
    Process: 36809 ExecStartPre=/bin/sh -c systemctl unset-environment _WSREP_>
    Process: 36811 ExecStartPre=/bin/sh -c [ ! -e /usr/bin/galera_recovery ] &>
    Process: 36850 ExecStartPost=/bin/sh -c systemctl unset-environment _WSREP>
    Process: 36852 ExecStartPost=/etc/mysql/debian-start (code=exited, status=>
   Main PID: 36840 (mariadbd)
     Status: "Taking your SQL requests now..."
      Tasks: 8 (limit: 1080)
     Memory: 57.5M
        CPU: 448ms
     CGroup: /system.slice/mariadb.service
             └─36840 /usr/sbin/mariadbd

Oct 08 16:37:42 Server1 mariadbd[36840]: Version: '10.6.7-MariaDB-2ubuntu1.1' >
Oct 08 16:37:42 Server1 systemd[1]: Started MariaDB 10.6.7 database server.
Oct 08 16:37:43 Server1 /etc/mysql/debian-start[36854]: Upgrading MySQL tables>
Oct 08 16:37:43 Server1 /etc/mysql/debian-start[36857]: Looking for 'mysql' as>
Oct 08 16:37:43 Server1 /etc/mysql/debian-start[36857]: Looking for 'mysqlchec>
Oct 08 16:37:43 Server1 /etc/mysql/debian-start[36857]: This installation of M>
Oct 08 16:37:43 Server1 /etc/mysql/debian-start[36857]: There is no need to ru>
Oct 08 16:37:43 Server1 /etc/mysql/debian-start[36857]: You can use --force if>
Oct 08 16:37:43 Server1 /etc/mysql/debian-start[36869]: Checking for insecure >
Oct 08 16:37:43 Server1 /etc/mysql/debian-start[36873]: Triggering myisam-reco>

jefferson@Server1:~$
```

The mariadb-server is now active because it is just installed

```
[jefferson@localhost ~]$ systemctl status mariadb
● mariadb.service - MariaDB database server
   Loaded: loaded (/usr/lib/systemd/system/mariadb.service; disabled; vendor preset: di
sabled)
   Active: inactive (dead)
```

The mariadb-server is inactive because it is not installed in the centos because of the ip address i used.

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.

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

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

```
jefferson@LocalMachine: ~/Langbid_PrelimExam

TASK [install apache and php for CentOS servers] ***************************************************
skipping: [192.168.56.105]
ok: [192.168.56.106]

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

TASK [Gathering Facts] *****************************************************************************
ok: [192.168.56.105]

TASK [install mariadb package (CentOS)] ************************************************************
skipping: [192.168.56.105]

TASK [Mariadb- Restarting/Enabling] ****************************************************************
changed: [192.168.56.105]

TASK [install mariadb package (Ubuntu)] ************************************************************
ok: [192.168.56.105]

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

TASK [Gathering Facts] *****************************************************************************
ok: [192.168.56.106]

TASK [install samba package] ***********************************************************************
changed: [192.168.56.106]

PLAY RECAP *****************************************************************************************
192.168.56.105             : ok=7    changed=1    unreachable=0    failed=0    skipped=3    rescued=0
    ignored=0
192.168.56.106             : ok=6    changed=1    unreachable=0    failed=0    skipped=2    rescued=0
    ignored=0

jefferson@LocalMachine:~/Langbid_PrelimExam$
```

The command is successful to download samba packages in both os.

The testing of the *file_servers* is beyond the scope of this activity, and as well as our topics and objectives. However, in this activity we were able to show that we can target hosts or servers using grouping in ansible playbooks.

Task 2: Using Tags in running playbooks

In this task, our goal is to add metadata to our plays so that we can only run the plays that we want to run, and not all the plays in our playbook.

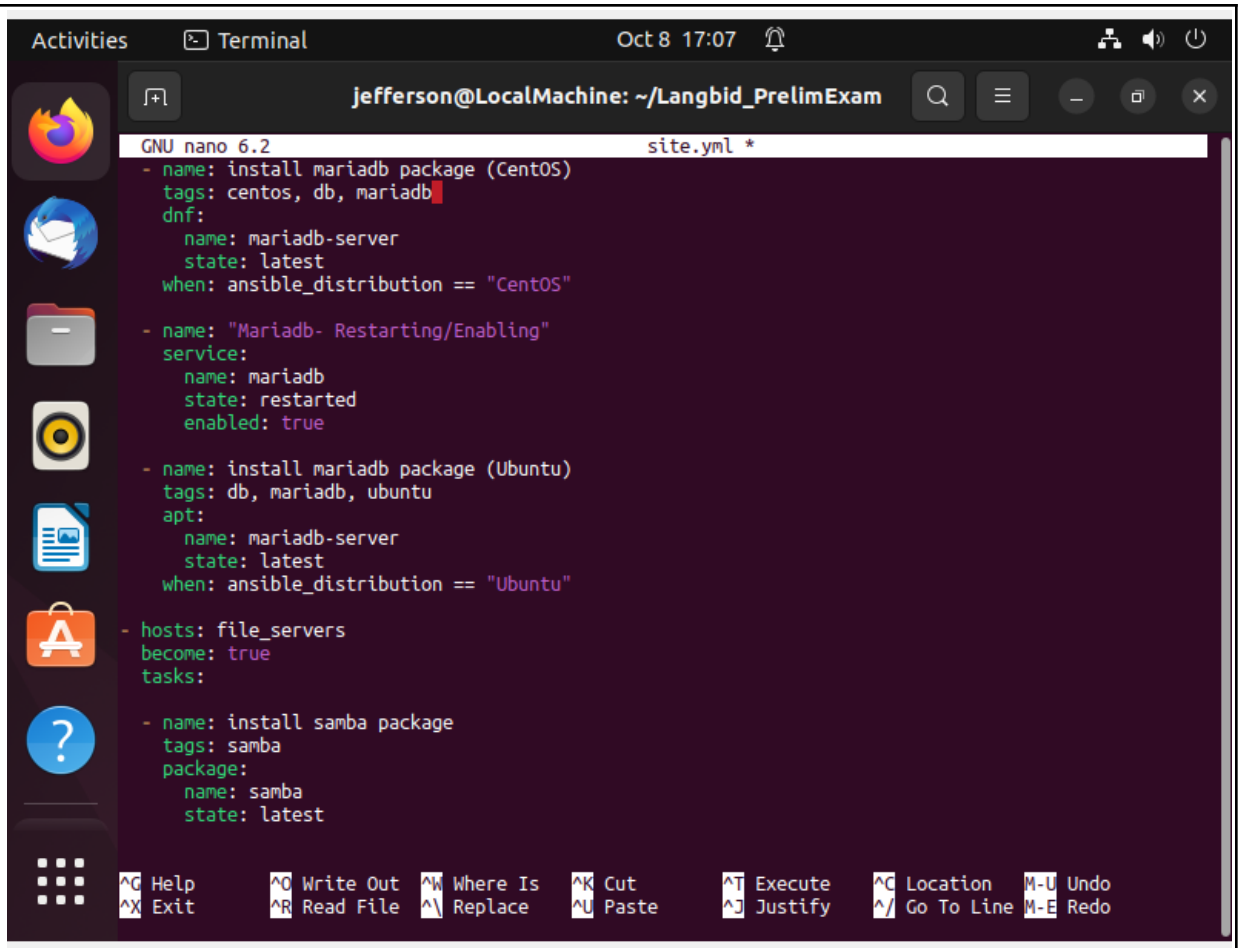
1. Edit the *site.yml* file. Add tags to the playbook. After the name, we can place the tags: *name_of_tag*. This is an arbitrary command, which means you can use any name for a tag.

```
---

- hosts: all
  become: true
  pre_tasks:

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

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

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

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

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

```
- hosts: db_servers
  become: true
  tasks:

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

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

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

- hosts: file_servers
  become: true
  tasks:

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

Make sure to save the file and exit.



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

```
jefferson@LocalMachine: ~/Langbid_PrelimExam

TASK [Gathering Facts] ***********************************************************************
ok: [192.168.56.105]
ok: [192.168.56.106]

TASK [install apache and php for Ubuntu servers] *********************************************
skipping: [192.168.56.106]
ok: [192.168.56.105]

TASK [install apache and php for CentOS servers] *********************************************
skipping: [192.168.56.105]
ok: [192.168.56.106]

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

TASK [Gathering Facts] ***********************************************************************
ok: [192.168.56.105]

TASK [install mariadb package (CentOS)] ******************************************************
skipping: [192.168.56.105]

TASK [Mariadb- Restarting/Enabling] ******************************************************
changed: [192.168.56.105]

TASK [install mariadb package (Ubuntu)] ******************************************************
ok: [192.168.56.105]

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

TASK [Gathering Facts] ***********************************************************************
ok: [192.168.56.106]

TASK [install samba package] *****************************************************************
ok: [192.168.56.106]

PLAY RECAP ***********************************************************************************
192.168.56.105             : ok=7    changed=1    unreachable=0    failed=0    skipped=3    rescued=0    ignored=0
192.168.56.106             : ok=6    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0

jefferson@LocalMachine:~/Langbid_PrelimExam$
```

The command is successful and it puts tags.

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

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

```
jefferson@LocalMachine:~/Langbid_PrelimExam$ 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]
```

it display the list of tags

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

```
jefferson@LocalMachine: ~/Langbid_PrelimExam

TASK [install updates (Ubuntu)] ***********************************************************
skipping: [192.168.56.106]
ok: [192.168.56.105]

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

TASK [Gathering Facts] ********************************************************************
ok: [192.168.56.106]
ok: [192.168.56.105]

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

TASK [Gathering Facts] ********************************************************************
ok: [192.168.56.105]

TASK [install mariadb package (CentOS)] ***************************************************
skipping: [192.168.56.105]

TASK [install mariadb package (Ubuntu)] ***************************************************
ok: [192.168.56.105]

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

TASK [Gathering Facts] ********************************************************************
ok: [192.168.56.106]

PLAY RECAP ********************************************************************************
192.168.56.105             : ok=5    changed=0    unreachable=0    failed=0    skipped=2    rescued=0
    ignored=0
192.168.56.106             : ok=4    changed=0    unreachable=0    failed=0    skipped=1    rescued=0
    ignored=0
```

it runs the tag with centos

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

```
jefferson@LocalMachine: ~/Langbid_PrelimExam
ok: [192.168.56.105]
ok: [192.168.56.106]

TASK [install updates (CentOS)] ***********************************************************************
skipping: [192.168.56.105]
ok: [192.168.56.106]

TASK [install updates (Ubuntu)] ***********************************************************************
skipping: [192.168.56.106]
ok: [192.168.56.105]

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

TASK [Gathering Facts] ********************************************************************************
ok: [192.168.56.105]
ok: [192.168.56.106]

TASK [install apache and php for Ubuntu servers] ******************************************************
skipping: [192.168.56.106]
ok: [192.168.56.105]

TASK [install apache and php for CentOS servers] ******************************************************
skipping: [192.168.56.105]
ok: [192.168.56.106]

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

TASK [Gathering Facts] ********************************************************************************
ok: [192.168.56.105]

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

TASK [Gathering Facts] ********************************************************************************
ok: [192.168.56.106]

PLAY RECAP ********************************************************************************************
192.168.56.105             : ok=5    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
192.168.56.106             : ok=5    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
```

it runs the command with db tag on it.

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

```
jefferson@LocalMachine: ~/Langbid_PrelimExam
skipping: [192.168.56.106]
ok: [192.168.56.105]

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

TASK [Gathering Facts] *********************************************************************
ok: [192.168.56.106]
ok: [192.168.56.105]

TASK [install apache and php for Ubuntu servers] *******************************************
skipping: [192.168.56.106]
ok: [192.168.56.105]

TASK [install apache and php for CentOS servers] *******************************************
skipping: [192.168.56.105]
ok: [192.168.56.106]

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

TASK [Gathering Facts] *********************************************************************
ok: [192.168.56.105]

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

TASK [Gathering Facts] *********************************************************************
ok: [192.168.56.106]

PLAY RECAP *********************************************************************************
192.168.56.105             : ok=5    changed=0    unreachable=0    failed=0    skipped=2    rescued=0
    ignored=0
192.168.56.106             : ok=5    changed=0    unreachable=0    failed=0    skipped=2    rescued=0
    ignored=0
```

It runs the command that has apache tag

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

```
jefferson@LocalMachine: ~/Langbid_PrelimExam

TASK [install updates (Ubuntu)] ************************************************************
skipping: [192.168.56.106]
ok: [192.168.56.105]

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

TASK [Gathering Facts] *********************************************************************
ok: [192.168.56.105]
ok: [192.168.56.106]

TASK [install apache and php for Ubuntu servers] *******************************************
skipping: [192.168.56.106]
ok: [192.168.56.105]

TASK [install apache and php for CentOS servers] *******************************************
skipping: [192.168.56.105]
ok: [192.168.56.106]

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

TASK [Gathering Facts] *********************************************************************
ok: [192.168.56.105]

TASK [install mariadb package (CentOS)] ****************************************************
skipping: [192.168.56.105]

TASK [install mariadb package (Ubuntu)] ****************************************************
ok: [192.168.56.105]

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

TASK [Gathering Facts] *********************************************************************
ok: [192.168.56.106]

PLAY RECAP *********************************************************************************
192.168.56.105             : ok=6    changed=0    unreachable=0    failed=0    skipped=3    rescued=0    ignored=0
192.168.56.106             : ok=5    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
```

It runs the command that has apache and db tags.

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.

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

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

```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
```

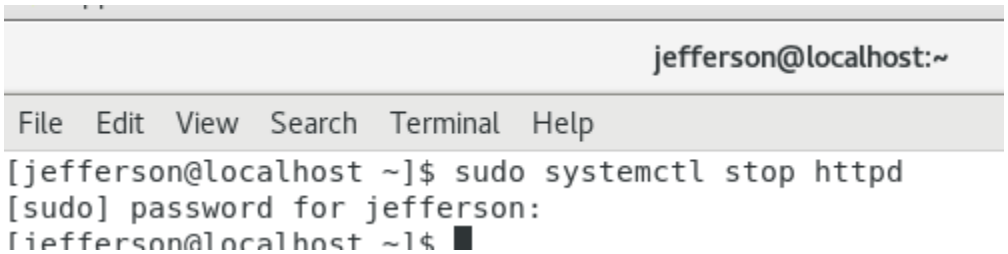
Figure 3.1.2

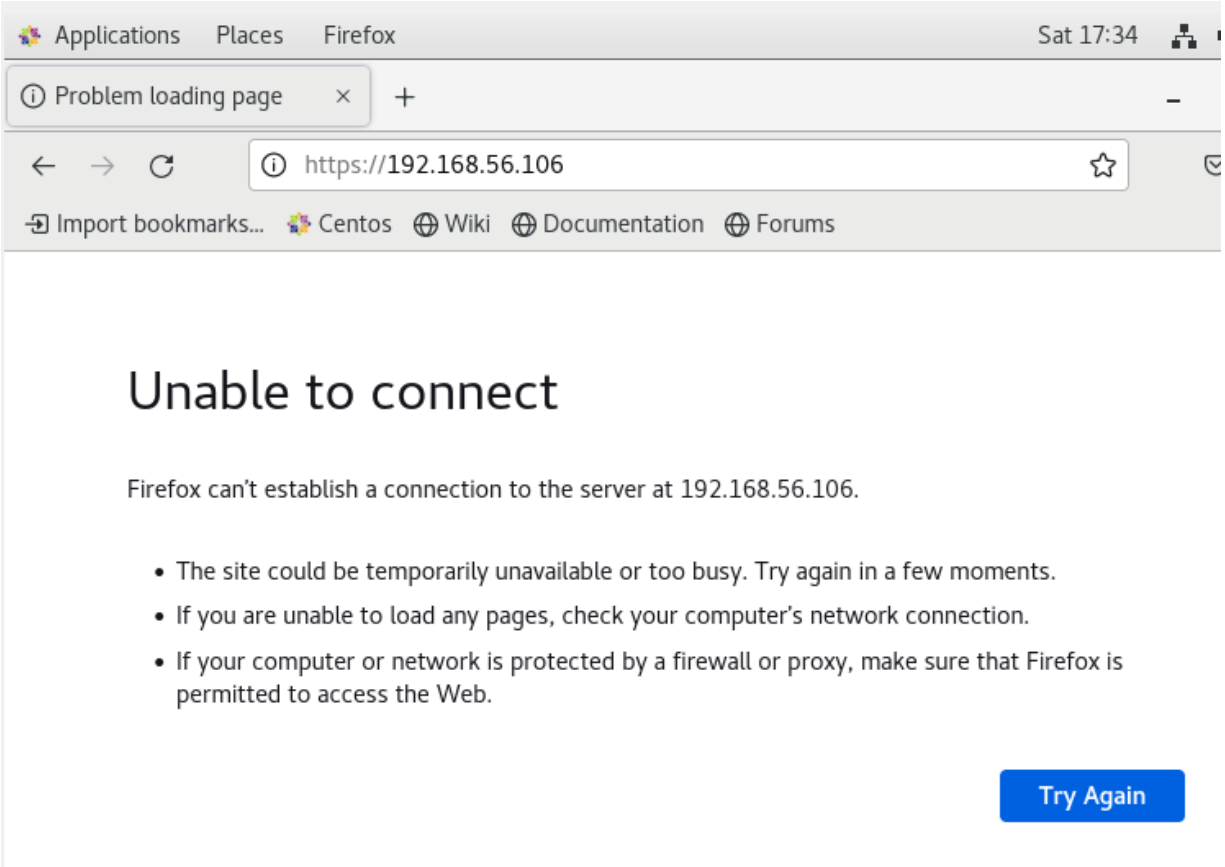
```
jefferson@LocalMachine: ~/Langbid_PrelimExam

ok: [192.168.56.105]

TASK [install apache and php for CentOS servers] ******************************************************
skipping: [192.168.56.105]
ok: [192.168.56.106]

TASK [start httpd (CentOS)] ***************************************************************************
skipping: [192.168.56.105]
changed: [192.168.56.106]

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

TASK [Gathering Facts] ********************************************************************************
ok: [192.168.56.105]

TASK [install mariadb package (CentOS)] ***************************************************************
skipping: [192.168.56.105]

TASK [Mariadb- Restarting/Enabling] *******************************************************************
changed: [192.168.56.105]

TASK [Mariadb- Restarting/Enabling] *******************************************************************
changed: [192.168.56.105]

TASK [install mariadb package (Ubuntu)] ***************************************************************
ok: [192.168.56.105]

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

TASK [Gathering Facts] ********************************************************************************
ok: [192.168.56.106]

TASK [install samba package] **************************************************************************
ok: [192.168.56.106]

PLAY RECAP ********************************************************************************************
192.168.56.105             : ok=8    changed=2    unreachable=0    failed=0    skipped=4    rescued=0    ignored=0
192.168.56.106             : ok=7    changed=1    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
```

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.

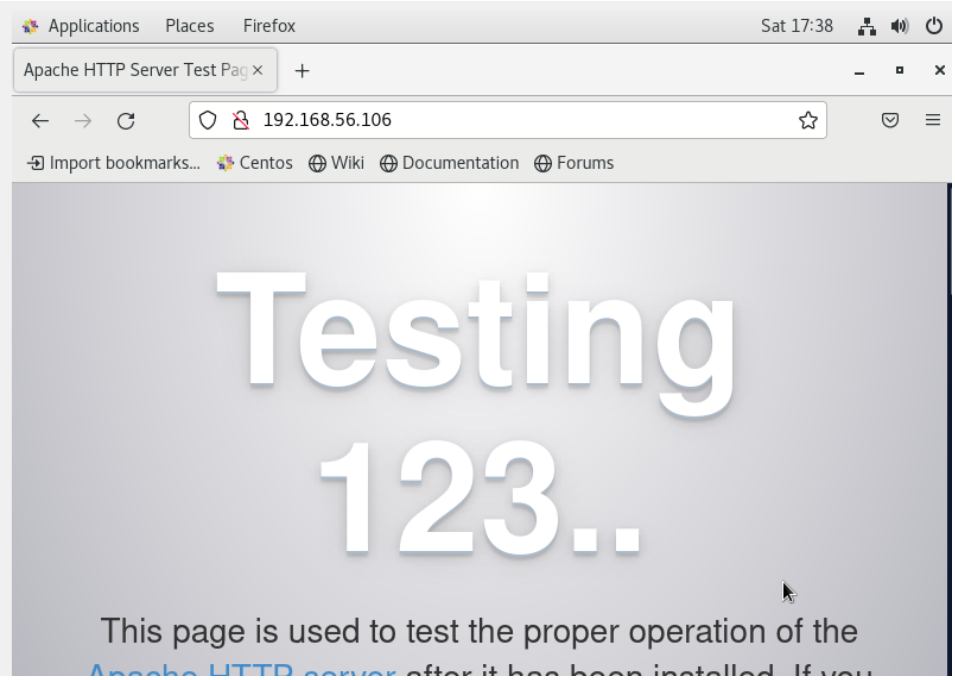




3. Go to the local machine and this time, run the *site.yml* file. Then after running the file, go again to the CentOS server and enter its IP address on the browser. Describe the result.
   To automatically enable the service every time we run the playbook, use the command *enabled: true* similar to Figure 7.1.2 and save the playbook.

```
jefferson@LocalMachine: ~/Langbid_PrelimExam

ok: [192.168.56.105]

TASK [install apache and php for CentOS servers] ******************************************************
skipping: [192.168.56.105]
ok: [192.168.56.106]

TASK [start httpd (CentOS)] ***************************************************************************
skipping: [192.168.56.105]
changed: [192.168.56.106]

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

TASK [Gathering Facts] ********************************************************************************
ok: [192.168.56.105]

TASK [install mariadb package (CentOS)] ***************************************************************
skipping: [192.168.56.105]

TASK [Mariadb- Restarting/Enabling] *******************************************************************
changed: [192.168.56.105]

TASK [Mariadb- Restarting/Enabling] *******************************************************************
changed: [192.168.56.105]

TASK [install mariadb package (Ubuntu)] ***************************************************************
ok: [192.168.56.105]

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

TASK [Gathering Facts] ********************************************************************************
ok: [192.168.56.106]

TASK [install samba package] **************************************************************************
ok: [192.168.56.106]

PLAY RECAP ********************************************************************************************
192.168.56.105             : ok=8    changed=2    unreachable=0    failed=0    skipped=4    rescued=0    ignored=0
192.168.56.106             : ok=7    changed=1    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
```

The command runs successfully and it can also connect to apache.



Reflections:

Answer the following:

1. What is the importance of putting our remote servers into groups?
   Putting our remote servers into groups is important because they access a computer or remote network. Remote users can access the devices using ssh and do not need physical connection.

2. What is the importance of tags in playbooks?
   Tags in playbooks are important because it is attached to the data that runs the command. It also makes using the playbook easier because it can run the command that has its specific tags.

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

   Some services are needed to be managed automatically in playbooks so that it can run automatically and helps to finish the job faster because it runs commands in the local machine to the specific ansible server.

Conclusion:

In conclusion, after conducting this activity I learned how to individualize hosts by putting the specific ip address with its role in the inventory file such as remote users, db servers, and file servers. Applying tags in selecting plays to run, putting tags in the commands between the name and the dnf or apt command in the site.yml and it runs specific tags. Also, managing services from remote servers by running a playbook command to install specific services.