

Name: Peñas, Issa Victoria H.	Date Performed: 10/04/2022
Course/Section: CPE232 - CPE31S1	Date Submitted: 10/04/2022
Instructor: Dr. Jonathan V. Taylor	Semester and SY: 1st Semester (SY: 2022- 2023)

Activity 6: Targeting Specific Nodes and Managing Services

1. Objectives:

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

2. Discussion:

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

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

Requirement:

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

Task 1: Targeting Specific Nodes

Filter	IP Address
[web_servers]	192.168.56.109 (Server 1)
	192.168.56.107 (CentOS)
[db_servers]	192.168.56.102 (Server 2)
[file_servers]	192.168.56.113 (Server 3)

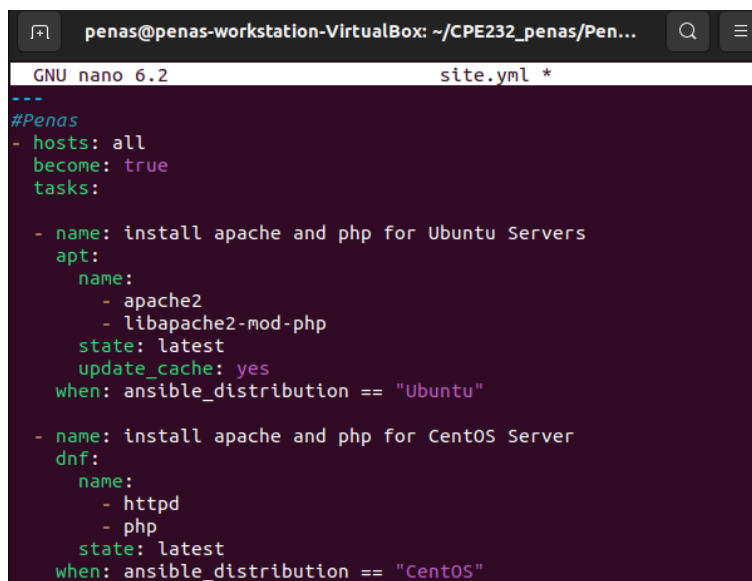
Table 1.1. Shows the IP Addressing table of the Servers

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"
```



```
penas@penas-workstation-VirtualBox: ~/CPE232_penas/Pen...
GNU nano 6.2 site.yml *
---
#Penas
- 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 Server
    dnf:
      name:
        - httpd
        - php
      state: latest
      when: ansible_distribution == "CentOS"
```

Figure 1.1. Creating a new .yml text file name *site.yml* with the given playbook commands

```

penas@penas-workstation-VirtualBox:~$ ssh-copy-id penas@192.168.56.113
The authenticity of host '192.168.56.113 (192.168.56.113)' can't be established.
ED25519 key fingerprint is SHA256:/bKErT9Pi5svawFtqoAXRx51InkcRUIefrgRA1AM64c.
This host key is known by the following other names/addresses:
  ~/.ssh/known_hosts:4: [hashed name]
  ~/.ssh/known_hosts:10: [hashed name]
  ~/.ssh/known_hosts:12: [hashed name]
  ~/.ssh/known_hosts:13: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? y
Please type 'yes', 'no' or the fingerprint: yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter
out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompt
ed now it is to install the new keys
penas@192.168.56.113's password:
Permission denied, please try again.
penas@192.168.56.113's password:

Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'penas@192.168.56.113'"
and check to make sure that only the key(s) you wanted were added.

penas@penas-workstation-VirtualBox:~$ ssh penas@Server_3
The authenticity of host 'server_3 (192.168.56.113)' can't be established.
ED25519 key fingerprint is SHA256:/bKErT9Pi5svawFtqoAXRx51InkcRUIefrgRA1AM64c.
This host key is known by the following other names/addresses:
  ~/.ssh/known_hosts:4: [hashed name]
  ~/.ssh/known_hosts:10: [hashed name]
  ~/.ssh/known_hosts:12: [hashed name]
  ~/.ssh/known_hosts:13: [hashed name]
  ~/.ssh/known_hosts:16: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'server_3' (ED25519) to the list of known hosts.
Welcome to Ubuntu 22.04.1 LTS (GNU/Linux 5.15.0-47-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

141 updates can be applied immediately.
42 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Last login: Tue Sep 13 12:28:00 2022 from 192.168.56.103

```

Figure 1.2. Adding the Public Key of the Server 3 to the Local Host with the IP Address of 192.168.56.113

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.

```

[web_servers]
#Server - 1
192.168.56.109
#CentOS
192.168.56.107

[db_servers]
#Server - 2
192.168.56.102

[file_servers]
#Server - 3
192.168.56.113

```

Figure 1.3. Shows the updated Inventory Text File containing the Filtered Out IP Addresses

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"

```

```

---
#Penas

- 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 Server
      dnf:
        name:
          - httpd
          - php
        state: latest
        when: ansible_distribution == "CentOS"

```

Figure 1.4. Updating the Playbook

Make sure to save the file and exit.

The *pre-tasks* command tells the ansible to run it before any other thing. In the *pre-tasks*, CentOS will install updates while Ubuntu will upgrade its distribution package. This will run before running the second play, which is targeted at *web_servers*. In the second play, apache and php will be installed on both Ubuntu servers and CentOS servers.

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

```

penas@penas-workstation-VirtualBox:~/CPE232_penas/penas/ansible$ ansible-playbook --ask-become-pass site.yml
BECOME password:

PLAY [all] *****

TASK [Gathering Facts] *****
ok: [192.168.56.107]
ok: [192.168.56.109]
fatal: [192.168.56.102]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to the host via ssh: ssh: connect to host 192.168.56.102 port 22: No route to host", "unreachable": true}
fatal: [192.168.56.113]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to the host via ssh: ssh: connect to host 192.168.56.113 port 22: No route to host", "unreachable": true}

TASK [Install Updates (CentOS)] *****
skipping: [192.168.56.109]
ok: [192.168.56.107]

TASK [Install Updates (Ubuntu)] *****
skipping: [192.168.56.107]
changed: [192.168.56.109]

PLAY [web_servers] *****

TASK [Gathering Facts] *****
ok: [192.168.56.107]
ok: [192.168.56.109]

TASK [install apache and php for Ubuntu Servers] *****
skipping: [192.168.56.107]
ok: [192.168.56.109]

TASK [install apache and php for CentOS Server] *****
skipping: [192.168.56.109]
ok: [192.168.56.107]

PLAY RECAP *****
192.168.56.102 : ok=0    changed=0    unreachable=1    failed=0    skipped=0
               rescued=0    ignored=0
192.168.56.107 : ok=4    changed=0    unreachable=0    failed=0    skipped=2
               rescued=0    ignored=0
192.168.56.109 : ok=4    changed=1    unreachable=0    failed=0    skipped=2
               rescued=0    ignored=0
192.168.56.113 : ok=0    changed=0    unreachable=1    failed=0    skipped=0

```

Figure 1.5.1. Shows the Output of the Playbook where Server 1 (192.168.56.109) and CentOS (192.168.56.107) was present

```

PLAY [web_servers] *****

PLAY RECAP *****
192.168.56.102 : ok=1  changed=0  unreachable=0  failed=1  skipped=1
               rescued=0  ignored=0
192.168.56.107 : ok=0  changed=0  unreachable=1  failed=0  skipped=0
               rescued=0  ignored=0
192.168.56.109 : ok=0  changed=0  unreachable=1  failed=0  skipped=0
               rescued=0  ignored=0
192.168.56.113 : ok=2  changed=0  unreachable=0  failed=0  skipped=1
               rescued=0  ignored=0

penas@penas-workstation-VirtualBox:~/CPE232_penas/Penas/ansible$ ansible-playbook --ask-b
ecome-pass site.yml
BECOME password:

PLAY [all] *****

TASK [Gathering Facts] *****
ok: [192.168.56.113]
fatal: [192.168.56.109]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to
the host via ssh: ssh: connect to host 192.168.56.109 port 22: No route to host", "unreach
able": true}
fatal: [192.168.56.107]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to
the host via ssh: ssh: connect to host 192.168.56.107 port 22: No route to host", "unreach
able": true}
ok: [192.168.56.102]

TASK [Install Updates (CentOS)] *****
skipping: [192.168.56.102]
skipping: [192.168.56.113]

TASK [Install Updates (Ubuntu)] *****
ok: [192.168.56.113]
ok: [192.168.56.102]

PLAY [web_servers] *****

PLAY RECAP *****
192.168.56.102 : ok=2  changed=0  unreachable=0  failed=0  skipped=1
               rescued=0  ignored=0
192.168.56.107 : ok=0  changed=0  unreachable=1  failed=0  skipped=0
               rescued=0  ignored=0
192.168.56.109 : ok=0  changed=0  unreachable=1  failed=0  skipped=0
               rescued=0  ignored=0
192.168.56.113 : ok=2  changed=0  unreachable=0  failed=0  skipped=1

```

Figure 1.5.2. Shows the Output of the Playbook where Server 2 (192.168.56.102) and Server 3 (192.168.56.113) was present

- As you can see on the output displayed on both Figure 1.5.1 and 1.5.2 that all commands were executed on the recorded Servers whereas its playbook commands can be accessed by all host, meaning all IP Address in the inventory text file will be affected by the changes due to the executed commands from the playbook
- 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"
```

```
- 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"
```

Figure 1.6. Updating the Playbook

Make sure to save the file and exit.

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


```

PLAY [all] *****

TASK [Gathering Facts] *****
ok: [192.168.56.113]
ok: [192.168.56.102]
fatal: [192.168.56.107]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to the host via ssh: ssh: connect to host 192.168.56.107 port 22: No route to host", "unreachable": true}
fatal: [192.168.56.109]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to the host via ssh: ssh: connect to host 192.168.56.109 port 22: No route to host", "unreachable": true}

TASK [Install Updates (CentOS)] *****
skipping: [192.168.56.102]
skipping: [192.168.56.113]

TASK [Install Updates (Ubuntu)] *****
ok: [192.168.56.113]
ok: [192.168.56.102]

PLAY [web_servers] *****

PLAY [db_servers] *****

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

TASK [Install mariadb package (CentOS)] *****
skipping: [192.168.56.102]

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

TASK [Install mariadb package (Ubuntu)] *****
ok: [192.168.56.102]

PLAY RECAP *****
192.168.56.102 : ok=5 changed=1 unreachable=0 failed=0 skipped=2
               rescued=0 ignored=0
192.168.56.107 : ok=0 changed=0 unreachable=1 failed=0 skipped=0
               rescued=0 ignored=0
192.168.56.109 : ok=0 changed=0 unreachable=1 failed=0 skipped=0
               rescued=0 ignored=0
192.168.56.113 : ok=2 changed=0 unreachable=0 failed=0 skipped=1
               rescued=0 ignored=0

```

Figure 1.7. Shows the Output of the Playbook where Server 2 (192.168.56.102) installed mariadb-server as its tags was [db_server]

- Appending a new set of commands, where recorded IP Addresses in **[db_servers]** will only be affected whenever running the playbook. In the output shown in Figure 1.7. the IP Address of 192.168.56.102 or Server 2 according to the IP Addressing Table is only affected on the changes after executing the ansible as the appended command was exclusively for **[db_servers]**.
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.

```
penas@penas-Server2-VirtualBox: ~  
penas@penas-Server2-VirtualBox:~$ systemctl status mariadb  
● mariadb.service - MariaDB 10.6.7 database server  
   Loaded: loaded (/lib/systemd/system/mariadb.service; enabled; vendor prese  
   Active: active (running) since Tue 2022-10-04 12:06:28 PST; 1min 34s ago  
     Docs: man:mariadb(8)  
           https://mariadb.com/kb/en/library/systemd/  
   Process: 47645 ExecStartPre=/usr/bin/install -m 755 -o mysql -g root -d /va  
   Process: 47646 ExecStartPre=/bin/sh -c systemctl unset-environment _WSREP_S  
   Process: 47648 ExecStartPre=/bin/sh -c [ ! -e /usr/bin/galera_recovery ] &&  
   Process: 47690 ExecStartPost=/bin/sh -c systemctl unset-environment _WSREP  
   Process: 47692 ExecStartPost=/etc/mysql/debian-start (code=exited, status=0  
 Main PID: 47677 (mariadb)  
   Status: "Taking your SQL requests now..."  
    Tasks: 9 (limit: 1402)  
  Memory: 61.1M  
     CPU: 498ms  
   CGroup: /system.slice/mariadb.service  
           └─47677 /usr/sbin/mariadb
```

Figure 1.8. Shows that the mariadb-server was Active(running) under the [db_server] which is Server 2 (192.168.56.102)

Describe the output.

- Executing the latest Ansible Playbook after appending the installation of mariadb whereas only [db_servers] were applied to these changes which is the Server 2. Inputting the given command that allows the user to view the status of mariadb in Server 2 proved that the package was successfully installed as its output was Active.
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
```

Figure 1.9. Appending the installation command of samba package
Run the *site.yml* file and describe the result.

```

TASK [Install Updates (CentOS)] *****
skipping: [192.168.56.102]
skipping: [192.168.56.113]

TASK [Install Updates (Ubuntu)] *****
ok: [192.168.56.113]
ok: [192.168.56.102]

PLAY [web_servers] *****

PLAY [db_servers] *****

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

TASK [Install mariadb package (CentOS)] *****
skipping: [192.168.56.102]

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

TASK [Install mariadb package (Ubuntu)] *****
ok: [192.168.56.102]

PLAY [file_servers] *****

TASK [Gathering Facts] *****
ok: [192.168.56.113]

TASK [install samba package] *****
changed: [192.168.56.113]

PLAY RECAP *****
192.168.56.102      : ok=5    changed=1    unreachable=0    failed=0    skipped=2
rescued=0    ignored=0
192.168.56.107      : ok=0    changed=0    unreachable=1    failed=0    skipped=0
rescued=0    ignored=0
192.168.56.109      : ok=0    changed=0    unreachable=1    failed=0    skipped=0
rescued=0    ignored=0
192.168.56.113      : ok=4    changed=1    unreachable=0    failed=0    skipped=1
rescued=0    ignored=0

```

Figure 1.10. the samba package was fully installed under [file_servers] which is the Server 3 (192.168.56.113) assigned

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

```

```

- 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
      update_cache: yes
    when: ansible_distribution == "Ubuntu"

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

```

```

- name: Install mariadb package (CentOS)
  tags: centos, db, mariadb
  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)
  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

```

Figure 2.1. Inputting tags on the respective Playbook Commands

Make sure to save the file and exit.

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

```
the host via ssh: ssh: connect to host 192.168.56.109 port 22: No route to host", "unreachable": true}
ok: [192.168.56.113]

TASK [Install Updates (CentOS)] *****
skipping: [192.168.56.102]
skipping: [192.168.56.113]

TASK [Install Updates (Ubuntu)] *****
ok: [192.168.56.102]
ok: [192.168.56.113]

PLAY [web_servers] *****

PLAY [db_servers] *****

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

TASK [Install mariadb package (CentOS)] *****
skipping: [192.168.56.102]

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

TASK [Install mariadb package (Ubuntu)] *****
ok: [192.168.56.102]

PLAY [file_servers] *****

TASK [Gathering Facts] *****
ok: [192.168.56.113]

TASK [install samba package] *****
ok: [192.168.56.113]

PLAY RECAP *****
192.168.56.102      : ok=5    changed=1    unreachable=0    failed=0    skipped=2
   rescued=0    ignored=0
192.168.56.107      : ok=0    changed=0    unreachable=1    failed=0    skipped=0
   rescued=0    ignored=0
192.168.56.109      : ok=0    changed=0    unreachable=1    failed=0    skipped=0
   rescued=0    ignored=0
192.168.56.113      : ok=4    changed=0    unreachable=0    failed=0    skipped=1
   rescued=0    ignored=0
```

Figure 2.2.1. Shows the Output of the Playbook where Server 2 (192.168.56.102) and Server 3 (192.168.56.113) was present

```

TASK [Gathering Facts] *****
ok: [192.168.56.107]
Fatal: [192.168.56.102]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to
the host via ssh: ssh: connect to host 192.168.56.102 port 22: Connection timed out", "un
reachable": true}
Fatal: [192.168.56.113]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to
the host via ssh: ssh: connect to host 192.168.56.113 port 22: Connection timed out", "un
reachable": true}
ok: [192.168.56.109]

TASK [Install Updates (CentOS)] *****
skipping: [192.168.56.109]
ok: [192.168.56.107]

TASK [Install Updates (Ubuntu)] *****
skipping: [192.168.56.107]
ok: [192.168.56.109]

PLAY [web_servers] *****

TASK [Gathering Facts] *****
ok: [192.168.56.107]
ok: [192.168.56.109]

TASK [install apache and php for Ubuntu Servers] *****
skipping: [192.168.56.107]
ok: [192.168.56.109]

TASK [install apache and php for CentOS Server] *****
skipping: [192.168.56.109]
ok: [192.168.56.107]

PLAY [db_servers] *****

PLAY [file_servers] *****

PLAY RECAP *****
192.168.56.102 : ok=0    changed=0    unreachable=1    failed=0    skipped=0
               rescued=0    ignored=0
192.168.56.107 : ok=4    changed=0    unreachable=0    failed=0    skipped=2
               rescued=0    ignored=0
192.168.56.109 : ok=4    changed=0    unreachable=0    failed=0    skipped=2
               rescued=0    ignored=0
192.168.56.113 : ok=0    changed=0    unreachable=1    failed=0    skipped=0
               rescued=0    ignored=0

```

Figure 2.2.2. Shows the Output of the Playbook where Server 1 (192.168.56.109) and CentOS (192.168.56.107) was present

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

NOTE: THE OUTPUTS WERE SHOWN SPECIFICALLY ON SPECIFIED SERVERS RELATED OT THE COMMANDS AS THE COMPUTER WAS SLOW WHERE IT CRASHES THE SERVER WHEN SIMULTANEOUSLY OPENED

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

```

penas@penas-workstation-VirtualBox:~/CPE232_penas/Penas/ansible$ 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: []

```

Figure 2.3. Shows the Summary of the playbook where only the tags of each set was outputted

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

```

penas@penas-workstation-VirtualBox:~/CPE232_penas/Penas/ansible$ ansible-playbook --tags
centos --ask-become-pass site.yml
BECOME password:

PLAY [all] *****

TASK [Gathering Facts] *****
fatal: [192.168.56.102]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to
the host via ssh: ssh: connect to host 192.168.56.102 port 22: No route to host", "unrea
chable": true}
fatal: [192.168.56.113]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to
the host via ssh: ssh: connect to host 192.168.56.113 port 22: No route to host", "unrea
chable": true}
ok: [192.168.56.107]
ok: [192.168.56.109]

TASK [Install Updates (CentOS)] *****
skipping: [192.168.56.109]
ok: [192.168.56.107]

TASK [Install Updates (Ubuntu)] *****
skipping: [192.168.56.107]
ok: [192.168.56.109]

PLAY [web_servers] *****

TASK [Gathering Facts] *****
ok: [192.168.56.107]
ok: [192.168.56.109]

PLAY [db_servers] *****

PLAY [file_servers] *****

PLAY RECAP *****
192.168.56.102      : ok=0    changed=0    unreachable=1    failed=0    skipped=0
  rescued=0    ignored=0
192.168.56.107      : ok=3    changed=0    unreachable=0    failed=0    skipped=1
  rescued=0    ignored=0
192.168.56.109      : ok=3    changed=0    unreachable=0    failed=0    skipped=1
  rescued=0    ignored=0
192.168.56.113      : ok=0    changed=0    unreachable=1    failed=0    skipped=0
  rescued=0    ignored=0

```

Figure 2.4. Runs the Playbook specifically on command sets that has **centos** tag where IP Address of 192.168.56.109 and 192.168.56.107 were only affected as they have the said tag

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

```

TASK [Gathering Facts] *****
fatal: [192.168.56.109]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to
the host via ssh: ssh: connect to host 192.168.56.109 port 22: No route to host", "unrea
chable": true}
fatal: [192.168.56.107]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to
the host via ssh: ssh: connect to host 192.168.56.107 port 22: No route to host", "unrea
chable": true}
ok: [192.168.56.113]
ok: [192.168.56.102]

TASK [Install Updates (CentOS)] *****
skipping: [192.168.56.102]
skipping: [192.168.56.113]

TASK [Install Updates (Ubuntu)] *****
ok: [192.168.56.113]
ok: [192.168.56.102]

PLAY [web_servers] *****

PLAY [db_servers] *****

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

TASK [Install mariadb package (CentOS)] *****
skipping: [192.168.56.102]

TASK [Install mariadb package (Ubuntu)] *****
ok: [192.168.56.102]

PLAY [file_servers] *****

TASK [Gathering Facts] *****
ok: [192.168.56.113]

PLAY RECAP *****
192.168.56.102      : ok=4    changed=0    unreachable=0    failed=0    skipped=2
  rescued=0    ignored=0
192.168.56.107      : ok=0    changed=0    unreachable=1    failed=0    skipped=0
  rescued=0    ignored=0
192.168.56.109      : ok=0    changed=0    unreachable=1    failed=0    skipped=0
  rescued=0    ignored=0
192.168.56.113      : ok=3    changed=0    unreachable=0    failed=0    skipped=1
  rescued=0    ignored=0

```

Figure 2.5. Runs the Playbook specifically on command sets that has **db** tag where IP Address of 192.168.56.113 and 192.168.56.102 were only affected as they have the said tag

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

```

penas@penas-workstation-VirtualBox:~/CPE232_penas/Penas/ansible$ ansible-playbook --tags
apache --ask-become-pass site.yml
BECOME password:

PLAY [all] *****

TASK [Gathering Facts] *****
ok: [192.168.56.113]
ok: [192.168.56.102]
fatal: [192.168.56.109]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to
the host via ssh: ssh: connect to host 192.168.56.109 port 22: No route to host", "unreach
able": true}
fatal: [192.168.56.107]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to
the host via ssh: ssh: connect to host 192.168.56.107 port 22: No route to host", "unreach
able": true}

TASK [Install Updates (CentOS)] *****
skipping: [192.168.56.102]
skipping: [192.168.56.113]

TASK [Install Updates (Ubuntu)] *****
ok: [192.168.56.113]
ok: [192.168.56.102]

PLAY [web_servers] *****

PLAY [db_servers] *****

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

PLAY [file_servers] *****

TASK [Gathering Facts] *****
ok: [192.168.56.113]

PLAY RECAP *****
192.168.56.102      : ok=3    changed=0    unreachable=0    failed=0    skipped=1
rescued=0    ignored=0
192.168.56.107      : ok=0    changed=0    unreachable=1    failed=0    skipped=0
rescued=0    ignored=0
192.168.56.109      : ok=0    changed=0    unreachable=1    failed=0    skipped=0
rescued=0    ignored=0
192.168.56.113      : ok=3    changed=0    unreachable=0    failed=0    skipped=1
rescued=0    ignored=0

```

Figure 2.6. Runs the Playbook specifically on command sets that has **apache** tag where IP Address of 192.168.56.113 and 192.168.56.102 were only affected as they have the said tag

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

```

TASK [Gathering Facts] *****
ok: [192.168.56.102]
ok: [192.168.56.113]
fatal: [192.168.56.109]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to
the host via ssh: ssh: connect to host 192.168.56.109 port 22: No route to host", "unreach
able": true}
fatal: [192.168.56.107]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to
the host via ssh: ssh: connect to host 192.168.56.107 port 22: No route to host", "unreach
able": true}

TASK [Install Updates (CentOS)] *****
skipping: [192.168.56.102]
skipping: [192.168.56.113]

TASK [Install Updates (Ubuntu)] *****
ok: [192.168.56.113]
ok: [192.168.56.102]

PLAY [web_servers] *****

PLAY [db_servers] *****

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

TASK [Install mariadb package (CentOS)] *****
skipping: [192.168.56.102]

TASK [Install mariadb package (Ubuntu)] *****
ok: [192.168.56.102]

PLAY [file_servers] *****

TASK [Gathering Facts] *****
ok: [192.168.56.113]

PLAY RECAP *****
192.168.56.102      : ok=4    changed=0    unreachable=0    failed=0    skipped=2
rescued=0    ignored=0
192.168.56.107      : ok=0    changed=0    unreachable=1    failed=0    skipped=0
rescued=0    ignored=0
192.168.56.109      : ok=0    changed=0    unreachable=1    failed=0    skipped=0
rescued=0    ignored=0
192.168.56.113      : ok=3    changed=0    unreachable=0    failed=0    skipped=1
rescued=0    ignored=0

```

Figure 2.7. Runs the Playbook specifically on command sets that has **apache** and **db** tag where IP Address of 192.168.56.102 and 192.168.56.113 were only affected as they have the said tag

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"
```

Figure 3.1. Appending the new set of commands under *'install apache and php for CentOS server'*

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:

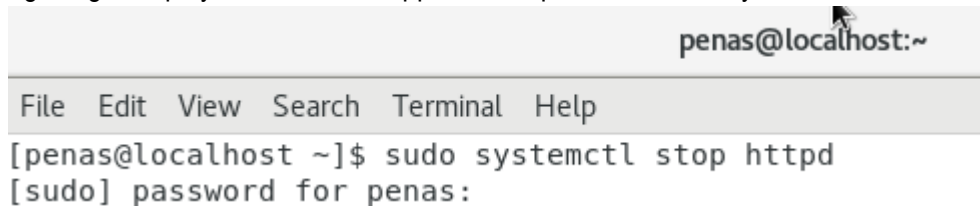
    - name: install mariadb package (CentOS)
      tags: centos, db,mariadb
      dnf:
        name: mariadb-server
        state: latest
        when: ansible_distribution == "CentOS"

    - 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.



```
penas@localhost:~
File Edit View Search Terminal Help
[penas@localhost ~]$ sudo systemctl stop httpd
[sudo] password for penas:
```

Figure 3.2. Halting the operations of httpd from CentOS

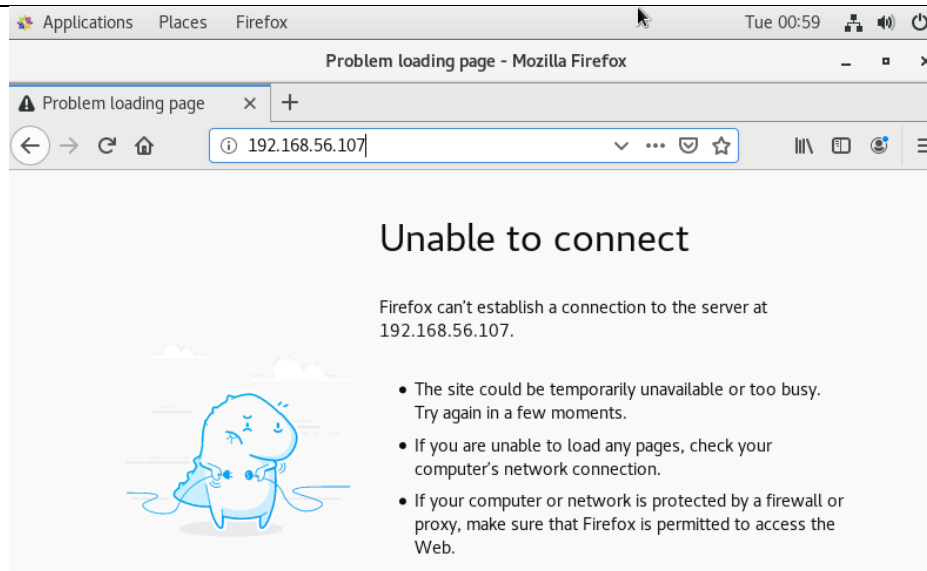


Figure 3.3. Verifying the httpd by entering the IP Address of the CentOS Server resulted on an error

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.

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

Figure 3.4. Inputting the command *enabled:true* from the new appended set of commands

```
penaz@penaz-workstation-VirtualBox: ~/CPE232_penaz/Penaz/ansible
become: pass
PLAY [all] *****
TASK [Gathering Facts] *****
ok: [192.168.56.113]
ok: [192.168.56.102]
ok: [192.168.56.109]
ok: [192.168.56.107]

TASK [Install updates (CentOS)] *****
skipping: [192.168.56.113]
skipping: [192.168.56.102]
ok: [192.168.56.109]
ok: [192.168.56.107]

TASK [Install updates (Ubuntu)] *****
ok: [192.168.56.113]
ok: [192.168.56.102]

PLAY [web_servers] *****

PLAY [db_servers] *****

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

TASK [Install mariadb package (CentOS)] *****
skipping: [192.168.56.102]

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

TASK [Install mariadb package (Ubuntu)] *****
ok: [192.168.56.113]
ok: [192.168.56.102]

PLAY [web_servers] *****

PLAY [db_servers] *****

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

TASK [Install mariadb package (CentOS)] *****
skipping: [192.168.56.102]

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

TASK [Install mariadb package (Ubuntu)] *****
ok: [192.168.56.113]
ok: [192.168.56.102]

PLAY [file_servers] *****

TASK [Gathering Facts] *****
ok: [192.168.56.113]

TASK [Install samba package] *****
ok: [192.168.56.113]

PLAY RECAP *****
192.168.56.102 : ok=5 changed=1 unreachable=0 failed=0 skipped=2 rescued=0 ignored=0
192.168.56.109 : ok=5 changed=0 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0
192.168.56.107 : ok=5 changed=0 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0
192.168.56.113 : ok=5 changed=0 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0
```

Reflections:

Answer the following:

1. What is the importance of putting our remote servers into groups?
 - [1] Its **easy to differentiate the specific needs on every servers**, it'll be highly unnecessary for a PC that only uses basic applications to install database packages resulting in the Administrator wasting valuable time and the PC wearing down it speed which comes to, [2] it **saves the companies budget**, Computers are always required to get an update to avoid security threats and to ease the employee's heavy labor. If specific Computers or Devices get their appropriate applications not only the company will make a profit but also saves its budget on fixing, updating, and buying applications as not all Computer requires to have the same copy of all storage. Lastly, [3] It is easier for the System Administrator to debug and determine a specific error from the group and saves valuable time.
2. What is the importance of tags in playbooks?

- Tags are somewhat similar to groups yet, Tags represent the details of the command by this [1] if an administrator wishes to only **update a specific command** that was related to the database the admin can only call the ansible to only execute commands that have tags on the database. Tags were only used to call specific details or categories on one and multiple commands from the playbook in only one file or one place. Practicing Tags will help the System Administrator to be concise and straightforward in executing the code, otherwise, it'll be redundant and confusing upon reading Playbook giving the admin or other staff a misconception.

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

- It'll be impossible for the System Administrator to install, update, repair, and debug issues within the server and computers/devices if done manually as it's tedious enough to update and a waste of time for the Admin to wait it one-by-one to avoid incoming issues while running resulting on the never-ending and heavy workload of the Administrator. Applying playbooks to services in a company allows the System Administrator to work on multiple tasks on different servers and computers/devices simultaneously, making it time efficient. If an error occurred the admin will be able to fix it as playbooks can able to pinpoint failed executed commands on a specific server or a computer by this information the Admin will be able to update the playbook according on the needs of the device or server.