Name: Kazuki A. Ogata	**Date Performed:** October 10, 2023
Course/Section: CPE 232 - CPE31S5	**Date Submitted:** October 14, 2023
Instructor: Engr. Roman Richard	**Semester and SY:** 1st semester S.Y 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 databases 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 the 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 installation. Take note of the IP address of 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.

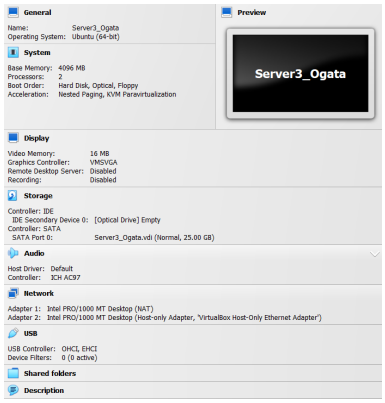


Figure 2.1 - Creating Ubuntu VM (Server 3)

I created another Ubuntu VM and activated the second adapter to a "host-only adapter".

```
kazuki@server3:~$ hostname -I
10.0.2.15 192.168.56.125
```

Figure 2.2 - Server 3 IP Address

I used the command "hostname -I" to display a list of all IP Address present with the current host.

```
kazuki@workstation:~$ ssh-copy-id -i ~/.ssh/id_rsa kazuki@192.168.56.125
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/kazuki/.ssh/id_rsa.pub"
The authenticity of host '192.168.56.125 (192.168.56.125)' can't be established.
ED25519 key fingerprint is SHA256:1BBFzgFmMVmbci+dpGifrfgBcYEWlQ5aJqzdr+voloI.
This host key is known by the following other names/addresses:
    ~/.ssh/known_hosts:4: [hashed name]
    ~/.ssh/known_hosts:7: [hashed name]
    ~/.ssh/known_hosts:14: [hashed name]
    ~/.ssh/known_hosts:15: [hashed name]
Are you sure you want to continue connecting (yes/no/[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 prompted now it is to install the new keys
kazuki@192.168.56.125's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'kazuki@192.168.56.125'"
and check to make sure that only the key(s) you wanted were added.
```

Figure 2.3 - Copying Public Key to Server 3

I used the command “ssh-copy-id” to copy the SSH public key to server 3 securely.

```
  GNU nano 6.2                                    /etc/hosts *
127.0.0.1       localhost
127.0.0.1       workstation

192.168.56.126  server1
192.168.56.123  server2
192.168.56.125  server3
192.168.56.124  centos
```

Figure 2.4 - Adding server 3 in hosts

I used the command “sudo nano /etc/hosts” and added there the IP address of server 3 and its hostname so it will be easy to use for connection.

```
kazuki@workstation:~$ ssh kazuki@server3
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 6.2.0-33-generic x86_64)

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

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Last login: Sat Oct 14 14:42:40 2023 from 192.168.56.121
kazuki@server3:~$
```

Figure 2.5 - Verifying Connection from Workstation to Server 3

After adding the IP address and hostname of server 3 in hosts in the workstation, I successfully created a connection between my workstation and server 3.

Task 1: Targeting Specific Nodes

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

```
  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]
192.168.56.126
192.168.56.123

[db_servers]
192.168.56.125

[file_servers]
192.168.56.124
```

Make sure to save the file and exit.

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:

```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.

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.

```
kazuki@workstation:~/CPE232_HOA6.1$ ansible-playbook --ask-become-pass site.yml
BECOME password:

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

TASK [Gathering Facts] *********************************************************************
ok: [192.168.56.123]
ok: [192.168.56.125]
ok: [192.168.56.127]
ok: [192.168.56.126]

TASK [install updates (CentOS)] ************************************************************
skipping: [192.168.56.126]
skipping: [192.168.56.123]
skipping: [192.168.56.125]
changed: [192.168.56.127]

TASK [install updates (Ubuntu)] ************************************************************
skipping: [192.168.56.127]
changed: [192.168.56.125]
changed: [192.168.56.126]
changed: [192.168.56.123]

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

TASK [Gathering Facts] *********************************************************************
ok: [192.168.56.123]
ok: [192.168.56.126]

TASK [install apache and php for Ubuntu servers] *******************************************
changed: [192.168.56.123]
changed: [192.168.56.126]

TASK [install apache and php for CentOS servers] *******************************************
skipping: [192.168.56.126]
skipping: [192.168.56.123]

PLAY RECAP *********************************************************************************
192.168.56.123             : ok=4    changed=2    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
192.168.56.125             : ok=2    changed=1    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
192.168.56.126             : ok=4    changed=2    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
192.168.56.127             : ok=2    changed=1    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
```

Figure 3.1.3 - Output of site.yml

The output shows a successful execution of all tasks/plays inside my playbook site.yml. The pre tasks command does other tasks and the targeted hosts are also separated. We can notice that in Ubuntu tasks it skips the IP Address of CentOS and vice versa wind CentOS tasks.

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

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

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

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

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

Make sure to save the file and exit.

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

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

TASK [Gathering Facts] **********************************************************************
ok: [192.168.56.125]

TASK [install mariadb package (CentOS)] *****************************************************
skipping: [192.168.56.125]

TASK [Mariadb- Restarting/Enabling] *********************************************************
changed: [192.168.56.125]

TASK [install mariadb package (Ubuntu)] *****************************************************
ok: [192.168.56.125]

PLAY RECAP ******************************************************************************************
192.168.56.123             : ok=4    changed=2    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
192.168.56.125             : ok=5    changed=2    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
192.168.56.126             : ok=4    changed=2    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
192.168.56.127             : ok=2    changed=1    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
```

Figure 3.1.4 - Output of db_server play

It shows that only the IP address/hosts included in "db_server" is the only one being executed.

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.

```
kazuki@server3:~$ hostname -I
10.0.2.15 192.168.56.125
kazuki@server3:~$ systemctl status mariadb
● mariadb.service - MariaDB 10.6.12 database server
     Loaded: loaded (/lib/systemd/system/mariadb.service; enabled; vendor prese
     Active: active (running) since Sat 2023-10-14 21:45:06 +08; 1min 56s ago
       Docs: man:mariadbd(8)
             https://mariadb.com/kb/en/library/systemd/
    Process: 17363 ExecStartPre=/usr/bin/install -m 755 -o mysql -g root -d /va
    Process: 17364 ExecStartPre=/bin/sh -c systemctl unset-environment _WSREP_S
    Process: 17366 ExecStartPre=/bin/sh -c [ ! -e /usr/bin/galera_recovery ] &&
    Process: 17406 ExecStartPost=/bin/sh -c systemctl unset-environment _WSREP_
    Process: 17408 ExecStartPost=/etc/mysql/debian-start (code=exited, status=0
   Main PID: 17395 (mariadbd)
     Status: "Taking your SQL requests now..."
      Tasks: 10 (limit: 4599)
     Memory: 61.0M
        CPU: 1.417s
     CGroup: /system.slice/mariadb.service
             └─17395 /usr/sbin/mariadbd
lines 1-17/17 (END)
```

```
[kazuki@centos ~]$ hostname -I
10.0.2.15 192.168.56.127 192.168.122.1
[kazuki@centos ~]$ systemctl status mariadb
● mariadb.service - MariaDB database server
   Loaded: loaded (/usr/lib/systemd/system/mariadb.service; disabled; vendor preset: d
sabled)
   Active: active (running) since Sat 2023-10-14 21:35:07 PST; 12min ago
  Process: 16188 ExecStartPost=/usr/libexec/mariadb-wait-ready $MAINPID (code=exited, 
tatus=0/SUCCESS)
  Process: 16102 ExecStartPre=/usr/libexec/mariadb-prepare-db-dir %n (code=exited, sta
us=0/SUCCESS)
 Main PID: 16187 (mysqld_safe)
    Tasks: 20
   CGroup: /system.slice/mariadb.service
           ├─16187 /bin/sh /usr/bin/mysqld_safe --basedir=/usr
           └─16352 /usr/libexec/mysqld --basedir=/usr --datadir=/var/lib/mysql --plu..
```

Figure 3.1.5 - Mariadb status for both Ubuntu (server 3) and CentOS

It shows that it is currently active after its installation in the playbook.

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: isntall samba package
    package:
      name: samba
      state: latest
```

   Make sure to save the file and exit.

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

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

TASK [Gathering Facts] *********************************************************
ok: [192.168.56.127]

TASK [isntall samba package] ***************************************************
changed: [192.168.56.127]

PLAY RECAP *********************************************************************
192.168.56.123             : ok=4    changed=2    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
192.168.56.125             : ok=5    changed=3    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
192.168.56.126             : ok=4    changed=2    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
192.168.56.127             : ok=4    changed=2    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
```

Figure 3.1.6 - Output of file_servers play

It successfully installed the latest samba package for CentOS (the IP address of CentOS is the only one inside the “file_server”) the “change” output verified that it was truly successfully executed.

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"
```

```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
    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
    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: isntall 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.

```
kazuki@workstation:~/CPE232_HOA6.1$ ansible-playbook --ask-become-pass site.yml
BECOME password:

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

TASK [Gathering Facts] *********************************************************
ok: [192.168.56.125]
ok: [192.168.56.126]
ok: [192.168.56.127]
ok: [192.168.56.123]

TASK [install updates (CentOS)] ************************************************
skipping: [192.168.56.126]
skipping: [192.168.56.123]
skipping: [192.168.56.125]
changed: [192.168.56.127]

TASK [install updates (Ubuntu)] ************************************************
skipping: [192.168.56.127]
changed: [192.168.56.126]
changed: [192.168.56.123]
changed: [192.168.56.125]

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

TASK [Gathering Facts] *********************************************************
ok: [192.168.56.123]
ok: [192.168.56.126]

TASK [install apache and php for Ubuntu servers] *******************************
changed: [192.168.56.123]
changed: [192.168.56.126]

TASK [install apache and php for CentOS servers] *******************************
skipping: [192.168.56.126]
skipping: [192.168.56.123]
```

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

TASK [Gathering Facts] *********************************************************
ok: [192.168.56.125]

TASK [install mariadb package (CentOS)] ****************************************
skipping: [192.168.56.125]

TASK [Mariadb- Restarting/Enabling] ********************************************
changed: [192.168.56.125]

TASK [install mariadb package (Ubuntu)] ****************************************
changed: [192.168.56.125]

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

TASK [Gathering Facts] *********************************************************
ok: [192.168.56.127]

TASK [isntall samba package] ***************************************************
changed: [192.168.56.127]

PLAY RECAP *********************************************************************
192.168.56.123             : ok=4    changed=2    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
192.168.56.125             : ok=5    changed=3    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
192.168.56.126             : ok=4    changed=2    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
192.168.56.127             : ok=4    changed=2    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
```

Figure 3.2.1 - Output of site.yml

In this playbook, we used the “tags” command where it is used to allow us to categorize the tasks and selectively run or skip them. Basically we can control which tasks to be executed when we run the playbook.

2. On the local machine, try to issue the following commands and describe each result:
   2.1 *ansible-playbook --list-tags site.yml*

```
kazuki@workstation:~/CPE232_HOA6.1$ 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]
```

- This command only displays a list of all the tags inside the playbook “site.yml”.

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

```
kazuki@workstation:~/CPE232_HOA6.1$ ansible-playbook --tags centos --ask-become-pass site.yml
BECOME password:

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

TASK [Gathering Facts] *********************************************************************
ok: [192.168.56.126]
ok: [192.168.56.127]
ok: [192.168.56.125]
ok: [192.168.56.123]

TASK [install updates (CentOS)] ************************************************************
skipping: [192.168.56.126]
skipping: [192.168.56.123]
skipping: [192.168.56.125]
changed: [192.168.56.127]

TASK [install updates (Ubuntu)] ************************************************************
skipping: [192.168.56.127]
changed: [192.168.56.123]
changed: [192.168.56.126]
changed: [192.168.56.125]

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

TASK [Gathering Facts] *********************************************************************
ok: [192.168.56.123]
ok: [192.168.56.126]

TASK [install apache and php for CentOS servers] *******************************************
skipping: [192.168.56.126]
skipping: [192.168.56.123]

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

TASK [Gathering Facts] *********************************************************************
ok: [192.168.56.125]

TASK [install mariadb package (CentOS)] ****************************************************
skipping: [192.168.56.125]

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

TASK [Gathering Facts] *********************************************************************
ok: [192.168.56.127]

PLAY RECAP *********************************************************************************
192.168.56.123             : ok=3    changed=1    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
192.168.56.125             : ok=3    changed=1    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
192.168.56.126             : ok=3    changed=1    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
192.168.56.127             : ok=3    changed=1    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
```

- This command executed all the tasks with a tag “centos” with it.

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

```
kazuki@workstation:~/CPE232_HOA6.1$ ansible-playbook --tags db --ask-become-pass site.yml
BECOME password:

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

TASK [Gathering Facts] *********************************************************************
ok: [192.168.56.126]
ok: [192.168.56.123]
ok: [192.168.56.125]
ok: [192.168.56.127]

TASK [install updates (CentOS)] ************************************************************
skipping: [192.168.56.126]
skipping: [192.168.56.123]
skipping: [192.168.56.125]
changed: [192.168.56.127]

TASK [install updates (Ubuntu)] ************************************************************
skipping: [192.168.56.127]
changed: [192.168.56.126]
changed: [192.168.56.123]
changed: [192.168.56.125]

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

TASK [Gathering Facts] *********************************************************************
ok: [192.168.56.126]
ok: [192.168.56.123]

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

TASK [Gathering Facts] *********************************************************************
ok: [192.168.56.125]

TASK [install mariadb package (CentOS)] ****************************************************
skipping: [192.168.56.125]

TASK [install mariadb package (Ubuntu)] ****************************************************
changed: [192.168.56.125]

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

TASK [Gathering Facts] *********************************************************************
ok: [192.168.56.127]

PLAY RECAP *********************************************************************************
192.168.56.123             : ok=3    changed=1    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
192.168.56.125             : ok=4    changed=2    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
192.168.56.126             : ok=3    changed=1    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
192.168.56.127             : ok=3    changed=1    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
```

- This command executed all the tasks with a tag “db” with it.

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

```
kazuki@workstation:~/CPE232_HOA6.1$ ansible-playbook --tags apache --ask-become-pass site.yml
BECOME password:

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

TASK [Gathering Facts] *********************************************************************************
ok: [192.168.56.123]
ok: [192.168.56.125]
ok: [192.168.56.126]
ok: [192.168.56.127]

TASK [install updates (CentOS)] ************************************************************************
skipping: [192.168.56.126]
skipping: [192.168.56.123]
skipping: [192.168.56.125]
changed: [192.168.56.127]

TASK [install updates (Ubuntu)] ************************************************************************
skipping: [192.168.56.127]
changed: [192.168.56.123]
changed: [192.168.56.125]
changed: [192.168.56.126]

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

TASK [Gathering Facts] *********************************************************************************
ok: [192.168.56.123]
ok: [192.168.56.126]

TASK [install apache and php for Ubuntu servers] *******************************************************
changed: [192.168.56.123]
changed: [192.168.56.126]

TASK [install apache and php for CentOS servers] *******************************************************
skipping: [192.168.56.126]
skipping: [192.168.56.123]

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

TASK [Gathering Facts] *********************************************************************************
ok: [192.168.56.125]

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

TASK [Gathering Facts] *********************************************************************************
ok: [192.168.56.127]

PLAY RECAP *********************************************************************************************
192.168.56.123             : ok=4    changed=2    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
192.168.56.125             : ok=3    changed=1    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
192.168.56.126             : ok=4    changed=2    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
192.168.56.127             : ok=3    changed=1    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
```

- This command executed all the tasks with a tag “apache” with it.

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

```
kazuki@workstation:~/CPE232_HOA6.1$ ansible-playbook --tags "apache,db" --ask-become-pass site.yml
BECOME password:

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

TASK [Gathering Facts] *********************************************************************************
ok: [192.168.56.123]
ok: [192.168.56.125]
ok: [192.168.56.126]
ok: [192.168.56.127]

TASK [install updates (CentOS)] ************************************************************************
skipping: [192.168.56.126]
skipping: [192.168.56.125]
skipping: [192.168.56.123]
changed: [192.168.56.127]

TASK [install updates (Ubuntu)] ************************************************************************
skipping: [192.168.56.127]
changed: [192.168.56.123]
changed: [192.168.56.126]
changed: [192.168.56.125]
PLAY [web_servers] *************************************************************************************

TASK [Gathering Facts] *********************************************************************************
ok: [192.168.56.123]
ok: [192.168.56.126]

TASK [install apache and php for Ubuntu servers] *******************************************************
changed: [192.168.56.123]
changed: [192.168.56.126]

TASK [install apache and php for CentOS servers] *******************************************************
skipping: [192.168.56.126]
skipping: [192.168.56.123]

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

TASK [Gathering Facts] *********************************************************************************
ok: [192.168.56.125]

TASK [install mariadb package (CentOS)] ****************************************************************
skipping: [192.168.56.125]

TASK [install mariadb package (Ubuntu)] ****************************************************************
changed: [192.168.56.125]

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

TASK [Gathering Facts] *********************************************************************************
ok: [192.168.56.127]

PLAY RECAP *********************************************************************************************
192.168.56.123             : ok=4    changed=2    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
192.168.56.125             : ok=4    changed=2    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
192.168.56.126             : ok=4    changed=2    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
192.168.56.127             : ok=3    changed=1    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
```

- This command executed all the tasks with a tag “apache” and “db” with it.

Task 3: Managing Services

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

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

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

Figure 3.1.1

Make sure to save the file and exit.

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

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

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

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

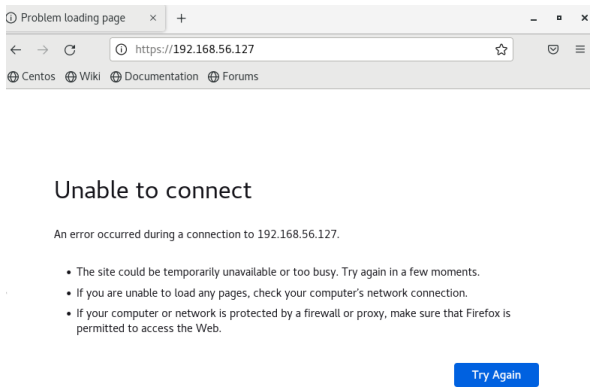
Figure 3.1.2

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.

```
[kazuki@centos ~]$ systemctl stop httpd
[kazuki@centos ~]$ systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; vendor preset: disa
bled)
   Active: inactive (dead)
     Docs: man:httpd(8)
           man:apachectl(8)
```

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.



- There is no display since the httpd service is stopped so there is no web server running in web pages.

Reflections:

Answer the following:

1. What is the importance of putting our remote servers into groups?
   - The importance of putting remote servers into groups is for convenient and efficient remote server management. We can easily apply different tasks/configurations to a specific group of remote servers. With this, we can easily handle, maintain, update, debug, or troubleshoot multiple remote servers simultaneously with similar objectives. For example, organizing your things in specific places. A cabinet with 5 drawers, in drawer 1, I will put all of my clothes (top), in drawer 2, I will put all of my pants/shorts (bottom), in drawer 3, I will put all of my underwear and socks, and etc. With this, I can easily locate them and it will save me time in choosing my outfit for the day.

2. What is the importance of tags in playbooks?
   - I actually don't know what tags are, so I research it. Based on my research, tags are used to control which plays/tasks to be executed when we run playbooks. So for me, the importance of tags is it allows us to target a task within the tasks/plays which can save our time and might also reduce errors. For example, if you have multiple tasks for CPE, ECE, and CE, you can use tags to execute only one or two of them, like "tags: CPE" or "tags: CPE,ECE", it will save you time and possibly avoid more errors.

3. Why do you think some services need to be managed automatically in playbooks?
   - I think some services need to be managed automatically in playbooks for efficiency and consistency. With automation, it ensures that the same procedure will always be the same every time it updates, reducing potential errors. Besides efficiency and consistency, with automation, it can also make the job faster (conveniency).