

Name: Dela cruz, Ivan Kenneth B.	Date Performed: Oct 12, 2023
Course/Section: CPE31S5	Date Submitted: Oct 14, 2023
Instructor: Engr. Roman Richard	Semester and SY: 1st Sem (2023 - 2024)

Activity 6: Targeting Specific Nodes and Managing Services

1. Objectives:

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

2. Discussion:

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

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

Requirement:

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

Task 1: Targeting Specific Nodes

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

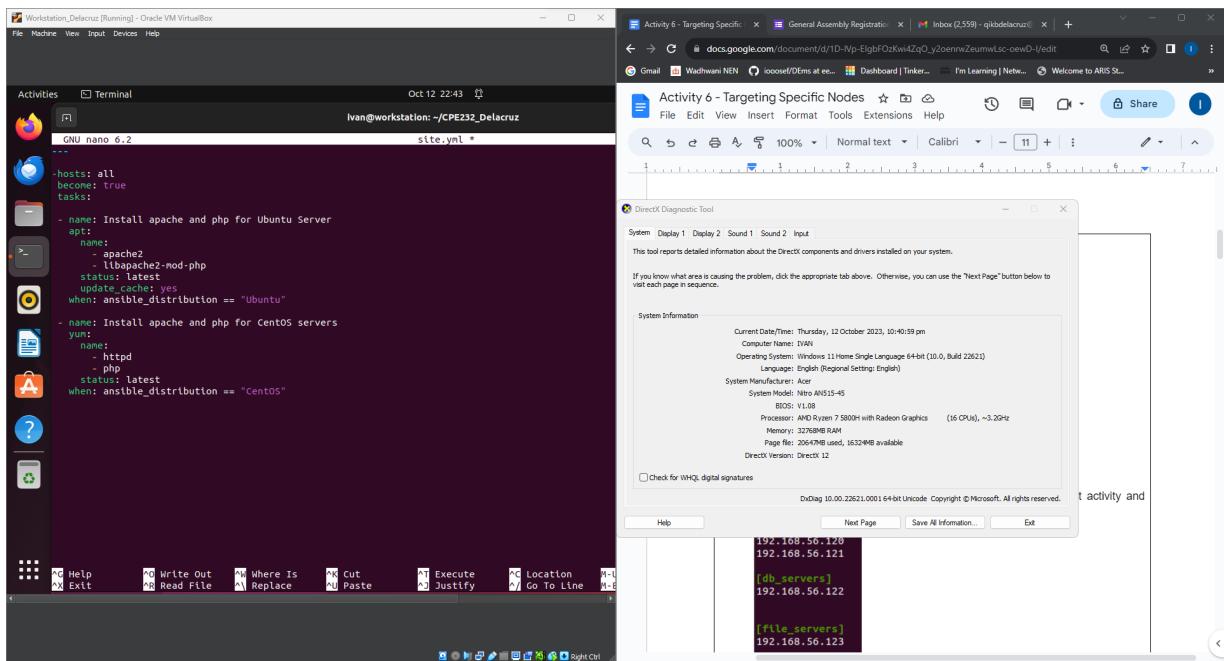
```

---
- 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"

```

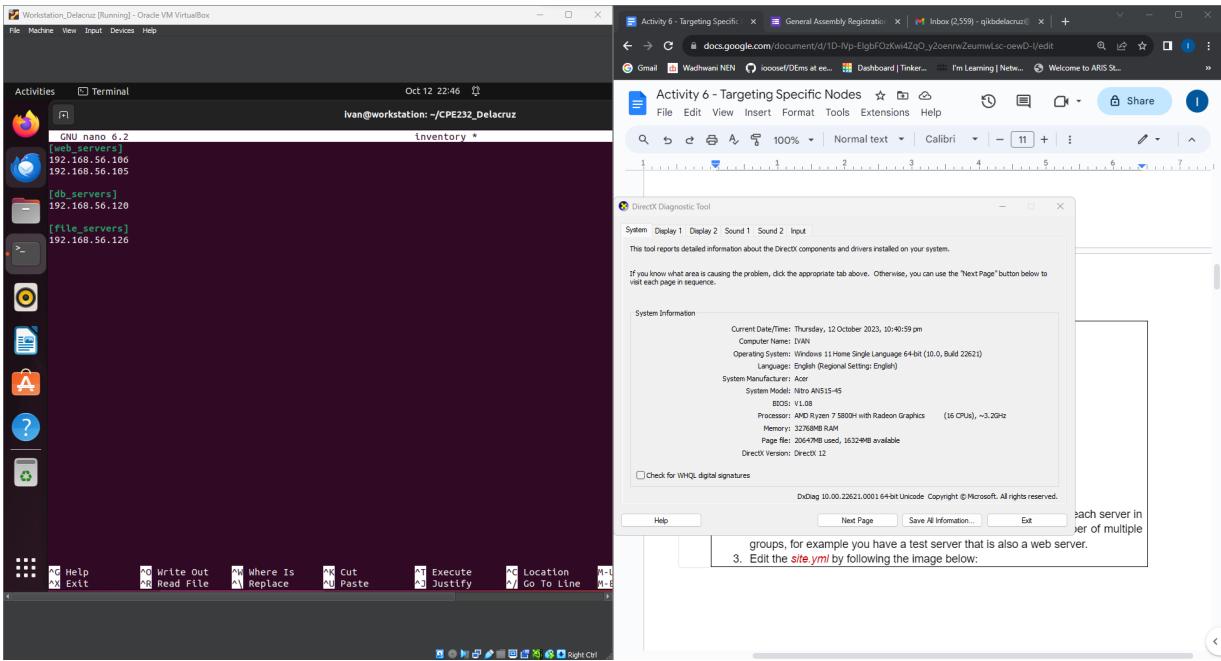


2. Edit the inventory file. Remove the variables we put in our last activity and group according to the image shown below:

```
[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.

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"

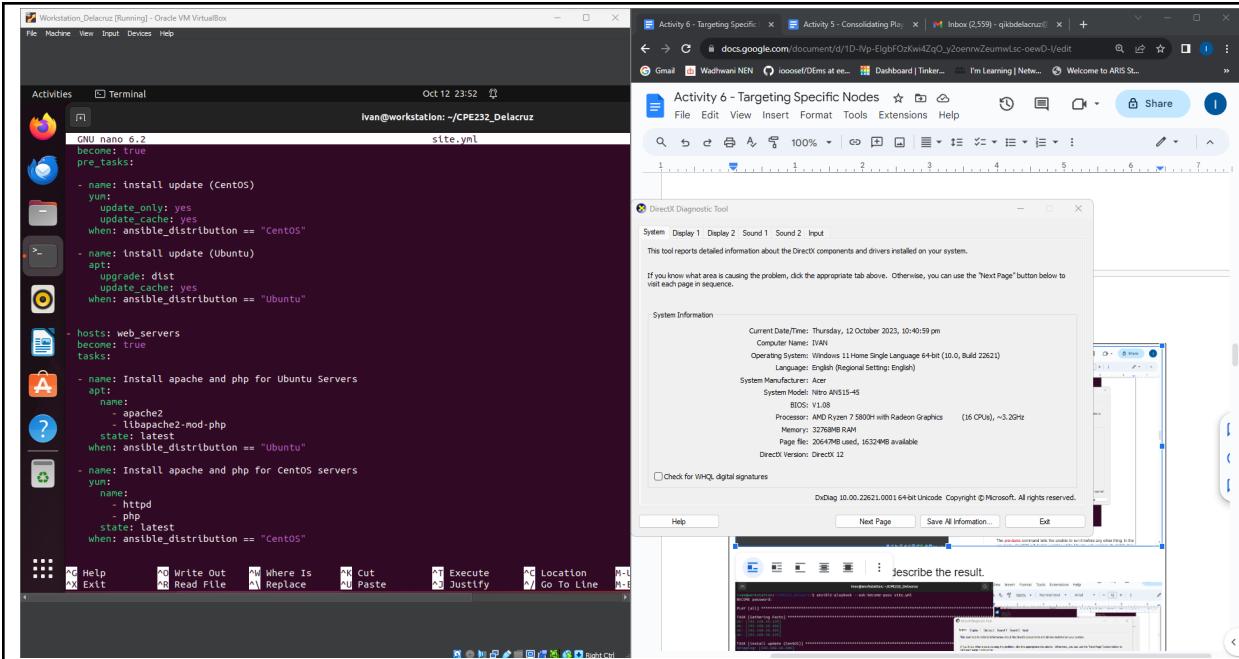
    - name: install updates (Ubuntu)
      apt:
        upgrade: dist
        update_cache: yes
      when: ansible_distribution == "Ubuntu"
```

```
- 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"

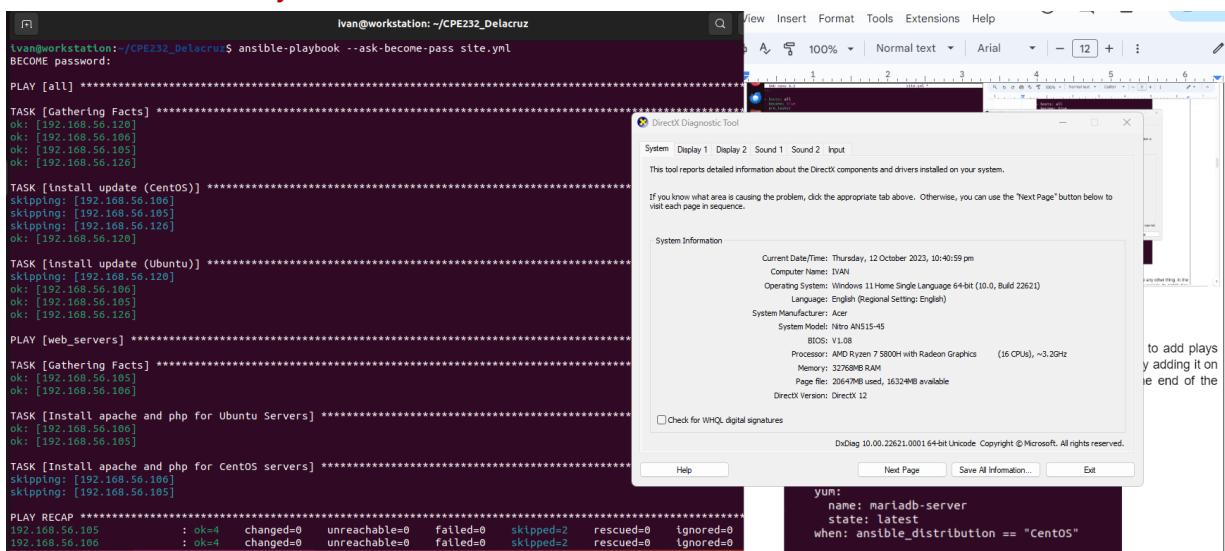
    - name: install apache and php for CentOS servers
      dnf:
        name:
          - httpd
          - php
        state: latest
      when: ansible_distribution == "CentOS"
```

Make sure to save the file and exit.

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.



- 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:

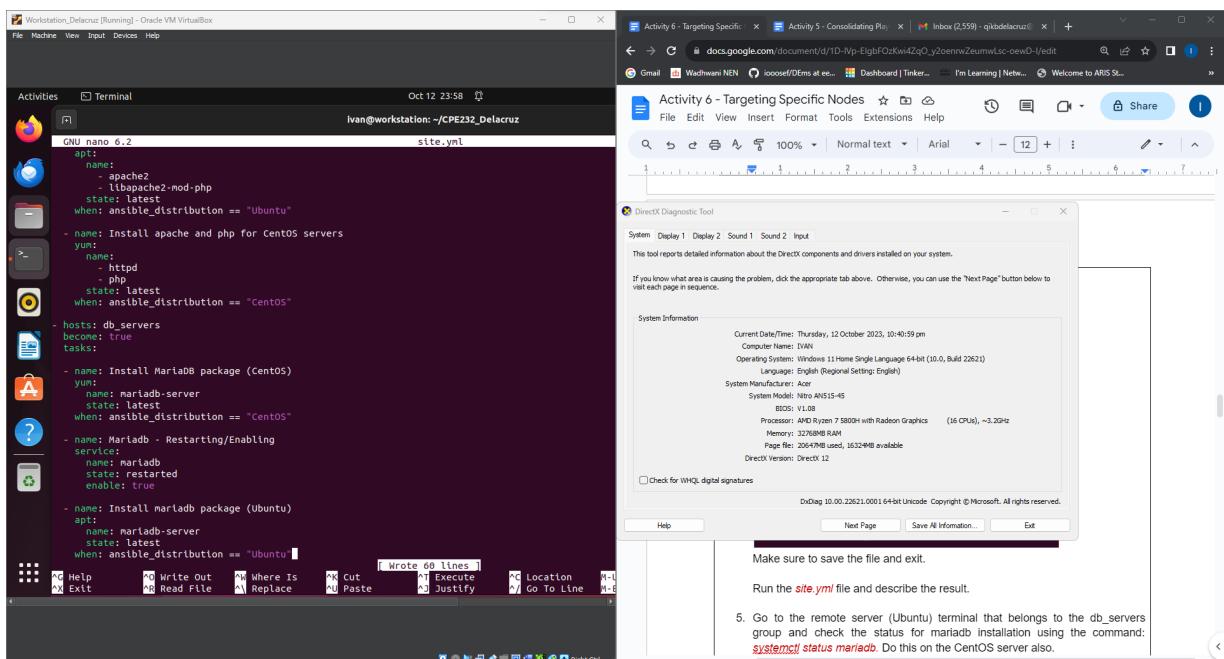
    - name: install mariadb package (CentOS)
      yum:
        name: mariadb-server
        state: latest
      when: ansible_distribution == "CentOS"

    - 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.

```

Oct 13 00:06 ivan@workstation: ~/CPE232_Delacruz
PLAY [web_servers] *****
TASK [Gathering Facts] *****
ok: [192.168.56.105]
ok: [192.168.56.106]
ok: [192.168.56.107]

TASK [Install apache and php for Ubuntu Servers] *****
ok: [192.168.56.105]
ok: [192.168.56.106]
ok: [192.168.56.107]

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

PLAY [db_servers] *****
TASK [Gathering Facts] *****
ok: [192.168.56.128]

TASK [Install MariaDB package (CentOS)] *****
ok: [192.168.56.128]

TASK [Install MariaDB package (Ubuntu)] *****
skipping: [192.168.56.128]

TASK [Mariadb - Restarting/Enabling] *****
changed: [192.168.56.128]

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
192.168.56.107 : ok=5    changed=1   unreachable=0   failed=0    skipped=2   rescued=0   ignored=0
192.168.56.128 : ok=2    changed=0   unreachable=0   failed=0    skipped=1   rescued=0   ignored=0
ivan@workstation: ~/CPE232_Delacruz:$

```

- On this part I successfully run the playbook and Install mariadb and also no error being encountered so far.

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.

```

Oct 13 00:09 ivan@localhost:~
File Edit View Search Terminal Help
File Machine View Input Devices Help
Applications Places Terminal Fri 00:08 ivan@localhost:~ - Directx Diagnostic Tool
System Display 1 Display 2 Sound 1 Sound 2 Input
This tool reports detailed information about the DirectX components and drivers installed on your system.
If you know what area is causing the problem, click the appropriate tab above. Otherwise, you can use the "Next Page" button below to visit each page in sequence.

System Information
Current Date/Time: Thursday, 12 October 2023, 10:40:59 pm
Computer Name: i5-10200H
Operating System: Windows 11 Home Single Language (4-bit) (10.0, Build 22621)
Language: English (Regional Setting: English)
System Manufacturer: Acer
System Model: Nitro AN515-45
BIOS: V1.08
Processor: AMD Ryzen 7 5800H with Radeon Graphics (16 CPUs), ~3.2GHz
Memory: 32768MB RAM
Page file: 2047MB used, 16324MB available
DirectX Version: DirectX 12
Check for WHQL digital signatures
DxDiag 10.0.22621.0001 64-bit Unicode Copyright © Microsoft. All rights reserved.
Help Next Page Save All Information... Exit
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.

```

```

Oct 13 00:23 localhost.localdomain systemd[1]: Starting MariaDB database server...
Oct 13 00:23 localhost.localdomain mariadb-prepare-db-dir[8773]: database MariaDB...
Oct 13 00:23 localhost.localdomain mysqld_safe[8808]: 231013 00:23 mysqld safe...
Oct 13 00:23 localhost.localdomain mysqld_safe[8808]: 231013 00:23 mysqld safe...
Oct 13 00:23 localhost.localdomain mysqld_safe[8808]: 231013 00:23 mysqld safe...
Oct 13 00:23 localhost.localdomain systemd[1]: Started MariaDB database server.
Hint: Some lines were ellipsized, use -l to show in full.
ivan@localhost:~$ state: latest
Make sure to save the file and exit.
Run the site.yml file and describe the result.
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.

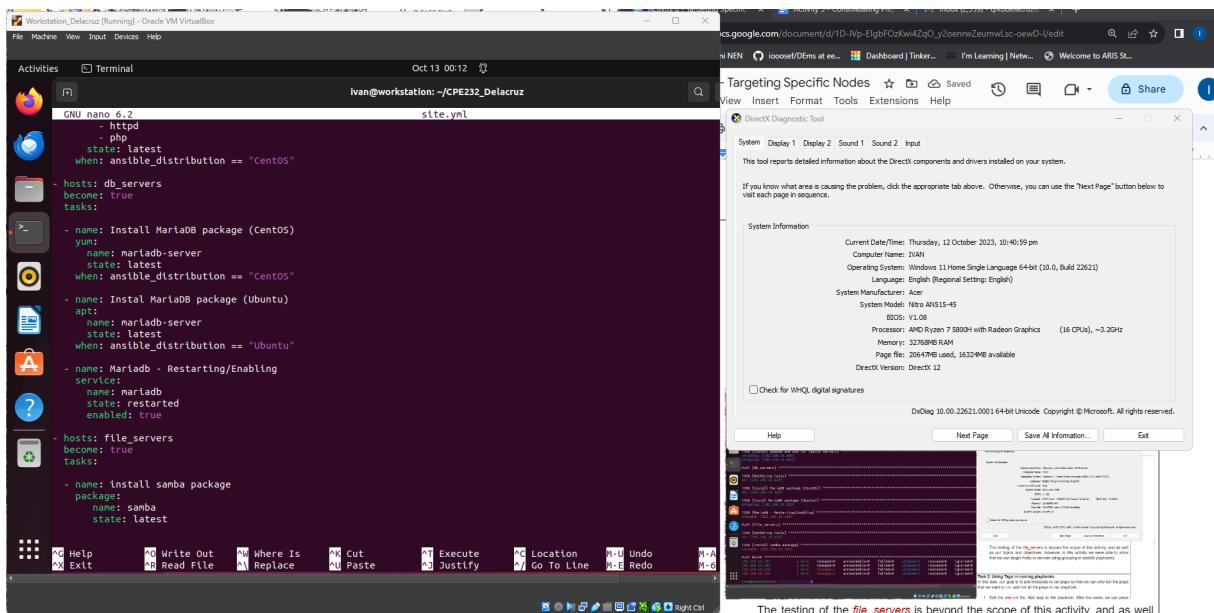
```

6. Edit the `site.yml` again. This time we will append the code to configure installation on the `file_servers` group. We can add the following on our file.

```
- hosts: file_servers
  become: true
  tasks:

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

Make sure to save the file and exit.



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

The screenshot shows a Windows desktop environment. On the left, a terminal window titled 'Workstation_Delacruz [Running] - Oracle VM VirtualBox' displays the output of an Ansible playbook. The output shows tasks being run on hosts with IP addresses 192.168.56.106, 192.168.56.105, and 192.168.56.105. The tasks include installing apache and php for Ubuntu servers, gathering facts, installing MariaDB packages, and restarting MySQL. On the right, a 'DirectX Diagnostic Tool' window is open, showing system information such as the operating system (Windows 11 Home Single Language 64-bit), processor (AMD Ryzen 7 5800H), and memory (32768MB RAM). Below the terminal window, a task bar shows icons for File Explorer, Task View, Start, and other system utilities.

- I was able to run the playbook successful and also able to Install samba on the file server which is server 3.

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.

```

ok: [192.168.56.106]
ok: [192.168.56.105]

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

PLAY [db_servers] ****
.
.
.

TASK [Gathering Facts] ****
ok: [192.168.56.120]

TASK [Install MariaDB package (Centos)] ****
ok: [192.168.56.120]

TASK [Mariadb - Restarting/Enabling] ****
changed: [192.168.56.120]

TASK [Install Marlab8 package (Ubuntu)] ****
skipping: [192.168.56.120]

PLAY [file_servers] ****
.
.
.

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

TASK [Install samba package] ****
ok: [192.168.56.126]

PLAY RECAP ****
192.168.56.105 : ok=4    changed=0   unreachable=0   failed=0    skipped=2   res
192.168.56.106 : ok=4    changed=0   unreachable=0   failed=0    skipped=2   res
cued@# ignored=0
192.168.56.120 : ok=5    changed=1   unreachable=0   failed=0    skipped=2   res
cued@# ignored=0
192.168.56.126 : ok=4    changed=0   unreachable=0   failed=0    skipped=1   res
cued@# ignored=0

ivan@workstation:/~CPE232_Delacruz$ 

```

- On this part , I was able to run the playbook successfully without encountering an error.

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

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

```

TASK [Mariadb - Restarting/Enabling] ****
changed: [192.168.56.120]

TASK [Install Marlab8 package (Ubuntu)] ****
skipping: [192.168.56.120]

PLAY [file_servers] ****
.
.
.

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

TASK [Install samba package] ****
ok: [192.168.56.126]

PLAY RECAP ****
192.168.56.105 : ok=4    changed=0   unreachable=0   failed=0    skipped=2   res
192.168.56.106 : ok=4    changed=0   unreachable=0   failed=0    skipped=2   res
cued@# ignored=0
192.168.56.120 : ok=5    changed=1   unreachable=0   failed=0    skipped=2   res
cued@# ignored=0
192.168.56.126 : ok=4    changed=0   unreachable=0   failed=0    skipped=1   res
cued@# ignored=0

ivan@workstation:/~CPE232_Delacruz$ ansible-playbook --list-tags 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]

ivan@workstation:/~CPE232_Delacruz$ ansible-playbook --tags centos --ask-become-pass site.yml

```

- On this command it displays all the list tags that have been in use in the file.

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

```

play # (file_servers): file_servers TAGS: []
  TASK TAGS: [samba]
ivan@workstation:~/CPE232_Delacruz$ ansible-playbook --tags centos --ask-become-pass site.yml
BECOME password:

PLAY [all] *****
  TASK [Gathering Facts] *****
    ok: [192.168.56.105]
    ok: [192.168.56.106]
    ok: [192.168.56.120]
    ok: [192.168.56.126]

  TASK [Install update (CentOS)] *****
    skipping: [192.168.56.105]
    skipping: [192.168.56.126]
    ok: [192.168.56.120]

  TASK [Install update (Ubuntu)] *****
    skipping: [192.168.56.120]
    ok: [192.168.56.105]
    ok: [192.168.56.106]
    ok: [192.168.56.126]

  TASK [Install apache and php for CentOS servers] *****
    skipping: [192.168.56.106]
    skipping: [192.168.56.105]
    ok: [192.168.56.120]

  PLAY [db_servers] *****
    TASK [Gathering Facts] *****
      ok: [192.168.56.126]

```

- On this part it only run the playbook with the tag of centos on it and script the other.

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

```

play # (file_servers): file_servers TAGS: []
  TASK TAGS: [samba]
ivan@workstation:~/CPE232_Delacruz$ ansible-playbook --tags db --ask-become-pass site.yml
BECOME password:

PLAY [all] *****
  TASK [Gathering Facts] *****
    ok: [192.168.56.105]
    ok: [192.168.56.106]
    ok: [192.168.56.120]
    ok: [192.168.56.126]

  TASK [Install update (CentOS)] *****
    skipping: [192.168.56.105]
    skipping: [192.168.56.126]
    ok: [192.168.56.120]

  TASK [Install update (Ubuntu)] *****
    skipping: [192.168.56.120]
    ok: [192.168.56.106]
    ok: [192.168.56.126]
    ok: [192.168.56.105]

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

  PLAY [db_servers] *****
    TASK [Gathering Facts] *****
      ok: [192.168.56.120]

    TASK [Install MariaDB package (CentOS)] *****
      ok: [192.168.56.120]

    TASK [Install MariaDB package (Ubuntu)] *****
      skipping: [192.168.56.120]

```

- On this part, It run the playbook with the tags of db on it and skip the other servers.

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

```

Ivan@workstation:~/CPE232_Delacruz$ ansible-playbook --tags apache --ask-become-pass site.yml
BECOME password:
PLAY [all] *****
TASK [Gathering Facts] *****
ok: [192.168.56.126]
ok: [192.168.56.105]
ok: [192.168.56.120]
ok: [192.168.56.106]

TASK [Install update (CentOS)] *****
skipping: [192.168.56.106]
skipping: [192.168.56.105]
skipping: [192.168.56.120]
ok: [192.168.56.126]

TASK [Install update (Ubuntu)] *****
skipping: [192.168.56.126]
ok: [192.168.56.120]
ok: [192.168.56.105]
ok: [192.168.56.106]

TASK [Install apache and php for Ubuntu Servers] *****
ok: [192.168.56.106]
ok: [192.168.56.105]

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

PLAY [db_servers] *****

```

System Information

Current Date/Time: Friday, 13 October 2023, 3:02:11 pm
Computer Name: IVAN
Operating System: Windows 11 Home Single Language 64-bit (10.0, Build 22621)
Language: English (Regional Setting: English)
System Manufacturer: Acer
System Model: Nitro AN515-45
BIOS: V1.08
Processor: AMD Ryzen 7 5800H with Radeon Graphics (16 CPUs), ~3.2GHz
Memory: 32768MB RAM
Page file: 30514MB used, 6457MB available
DirectX Version: DirectX 12

Task 3: Managing Services

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

- On this part, It run the playbook with the tags of apache on it and skip the other servers.

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

```

Ivan@workstation:~/CPE232_Delacruz$ ansible-playbook --tags "apache,db" --ask-become-pass site.yml
BECOME password:
PLAY [all] *****
TASK [Gathering Facts] *****
ok: [192.168.56.126]
ok: [192.168.56.105]
ok: [192.168.56.120]
ok: [192.168.56.106]

TASK [Install update (CentOS)] *****
skipping: [192.168.56.106]
skipping: [192.168.56.105]
skipping: [192.168.56.120]
ok: [192.168.56.126]

TASK [Install update (Ubuntu)] *****
skipping: [192.168.56.126]
ok: [192.168.56.120]
ok: [192.168.56.105]
ok: [192.168.56.106]

TASK [Install apache and php for Ubuntu Servers] *****
ok: [192.168.56.106]
ok: [192.168.56.105]

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

PLAY [db_servers] *****

```

System Information

Current Date/Time: Friday, 13 October 2023, 3:02:11 pm
Computer Name: IVAN
Operating System: Windows 11 Home Single Language 64-bit (10.0, Build 22621)
Language: English (Regional Setting: English)
System Manufacturer: Acer
System Model: Nitro AN515-45
BIOS: V1.08
Processor: AMD Ryzen 7 5800H with Radeon Graphics (16 CPUs), ~3.2GHz
Memory: 32768MB RAM
Page file: 30514MB used, 6457MB available
DirectX Version: DirectX 12

Task 3: Managing Services

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

- On this part, It run the playbook with the tags of apache and db on it and skip the other servers.

Task 3: Managing Services

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

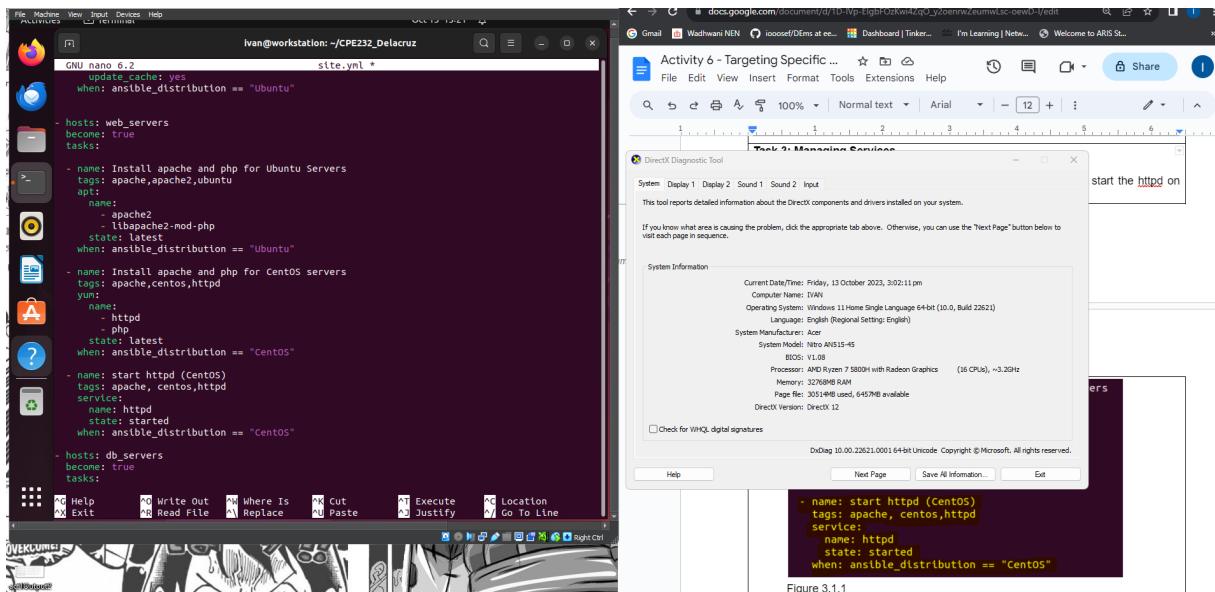
```

- 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
      dnf:
        name: mariadb-server
        state: latest
      when: ansible_distribution == "CentOS"

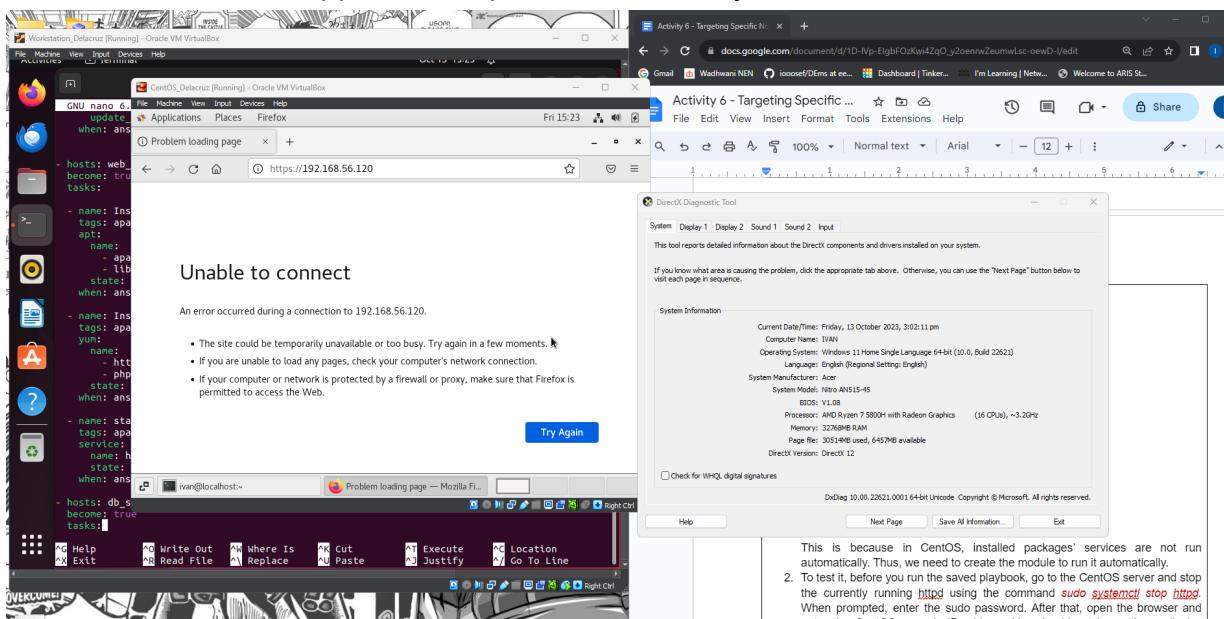
    - name: "Mariadb- Restarting/Enabling"
      service:
        name: mariadb
        state: restarted
        enabled: true

```

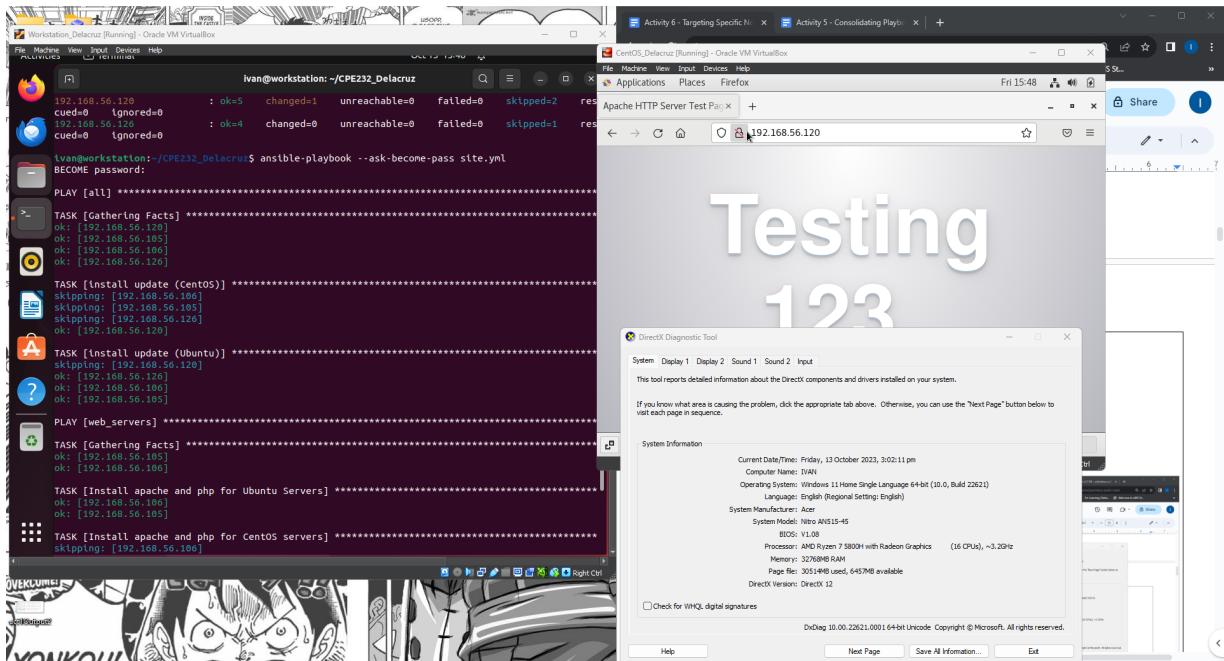
Figure 3.1.2

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.



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.

Reflections:

Answer the following:

1. What is the importance of putting our remote servers into groups?
 - The importance of putting the remote server on the group, it can actually fix some errors that can be encountered. Also, It is easy to update or install on a specific server.
2. What is the importance of tags in playbooks?
 - The tag was important especially on the long script wherein we don't want to edit it again or change some of it, by simply adding a tag on it we can choose which on script we want to run with running another part of the script.
3. Why do think some services need to be managed automatically in playbooks?
 - Services like automatically updating a managing node.

