Name: Bernice M. Peña	**Date Performed:** 09/20/2023
Course/Section: Managing Enterprise Servers / CPE31S5	**Date Submitted:** 09/23/2023
Instructor: Engr. Roman Richard	**Semester and SY:** 1st semester, SY 2023-2024

Activity 5: Consolidating Playbook plays

1. **Objectives:**

1.1 Use **when** command in playbook for different OS distributions

1.2 Apply refactoring techniques in cleaning up the playbook codes

2. **Discussion**:

We are going to look at a way that we can differentiate a playbook by a host in terms of which distribution the host is running. It's very common in most Linux shops to run multiple distributions, for example, Ubuntu shop or Debian shop and you need a different distribution for a one off-case or perhaps you want to run plays only on certain distributions.

It is a best practice in ansible when you are working in a collaborative environment to use the command git pull. git pull is a Git command used to update the local version of a repository from a remote. By default, git pull does two things. Updates the current local working branch (currently checked out branch) and updates the remote-tracking branches for all other branches. git pull essentially pulls down any changes that may have happened since the last time you worked on the repository.

Requirement:

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

Task 1: Use when command for different distributions

1. In the local machine, make sure you are in the local repository directory (*CPE232_yourname*). Issue the command git pull. When prompted, enter the correct passphrase or password. Describe what happened when you issue this command. Did something happen? Why?

```
bernice@Workstation-Pena:~$ cd CPE232_BernicePena
bernice@Workstation-Pena:~/CPE232_BernicePena$ git pull
Already up to date.
bernice@Workstation-Pena:~/CPE232_BernicePena$
```

After issuing the command git pull in my repository, the message "Already up-to-date" appeared, this means that there were no changes in the repository that needed to be fetched.

2. Edit the inventory file and add the IP address of the Centos VM. Issue the command we used to execute the playbook (the one we used in the last activity): *ansible-playbook --ask-become-pass install_apache.yml*. After executing this command, you may notice that it did not become successful in the Centos VM. You can see that the Centos VM has failed=1. Only the two remote servers have been changed. The reason is that Centos VM does not support "apt" as the package manager. The default package manager for Centos is "yum."

```
  GNU nano 6.2                          inventory.ini
[ubuntu_servers]
192.168.1.18
192.168.1.19

[centos_server]
192.168.56.106
```

This is where I added the IP address of my CentOS VM to inventory

```
bernice@Workstation-Pena:~/CPE232_BernicePena$ ansible-playbook --ask-become-pass install_apache.yml
BECOME password:

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

TASK [Gathering Facts] *****************************************************************************************
ok: [192.168.1.19]
ok: [192.168.1.18]
ok: [192.168.56.106]

TASK [Gathering Facts] *****************************************************************************************
ok: [192.168.1.18]
ok: [192.168.1.19]
ok: [192.168.56.106]

TASK [Gathering Facts] *****************************************************************************************
skipping: [192.168.1.18]
skipping: [192.168.1.19]
ok: [192.168.56.106]

TASK [fail] ****************************************************************************************************
skipping: [192.168.1.18]
skipping: [192.168.1.19]
fatal: [192.168.56.106]: FAILED! => {"changed": false, "msg": ""}

PLAY RECAP *****************************************************************************************************
192.168.1.18               : ok=2    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
192.168.1.19               : ok=2    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
192.168.56.106             : ok=3    changed=0    unreachable=0    failed=1    skipped=0    rescued=0    ignored=0

bernice@Workstation-Pena:~/CPE232_BernicePena$
```

After executing the command, the IP address of my CentOS has failed=1 since it doesn't support apt package manager.

3. Edit the *install_apache.yml* file and insert the lines shown below.

```
---
- hosts: all
  become: true
  tasks:

  - name: update repository index
    apt:
      update_cache: yes
    when: ansible_distribution == "Ubuntu"

  - name: install apache2 package
    apt:
      name: apache2
    when: ansible_distribution == "Ubuntu"

  - name: add PHP support for apache
    apt:
      name: libapache2-mod-php
    when: ansible_distribution == "Ubuntu"
```

Make sure to save the file and exit.

Run *ansible-playbook --ask-become-pass install_apache.yml* and describe the result.

```
bernice@Workstation-Pena:~/CPE232_BernicePena$ ansible-playbook --ask-become-pass install_apache.yml
BECOME password:

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

TASK [Gathering Facts] *********************************************************
ok: [192.168.1.18]
ok: [192.168.1.19]
ok: [192.168.56.106]

TASK [update repository index] *************************************************
skipping: [192.168.56.106]
changed: [192.168.1.19]
changed: [192.168.1.18]

TASK [install apache2 package] *************************************************
skipping: [192.168.56.106]
changed: [192.168.1.19]
changed: [192.168.1.18]

TASK [add PHP support for apache] **********************************************
skipping: [192.168.56.106]
changed: [192.168.1.19]
changed: [192.168.1.18]

PLAY RECAP *********************************************************************
192.168.1.18               : ok=4    changed=3    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
192.168.1.19               : ok=4    changed=3    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
192.168.56.106             : ok=1    changed=0    unreachable=0    failed=0    skipped=3    rescued=0    ignored=0
```

It successfully updated the repository for ubuntu servers, the apache2 installation was also successful for the servers resulting in "changed" status. Regarding with the CentOS, the status was "skipped".

If you have a mix of Debian and Ubuntu servers, you can change the configuration of your playbook like this.

- name: update repository index
  apt:
    update_cache: yes
  when: ansible_distribution in ["Debian", "Ubuntu]

Note: This will work also if you try. Notice the changes are highlighted.

4. Edit the *install_apache.yml* file and insert the lines shown below.

```
---
- hosts: all
  become: true
  tasks:

  - name: update repository index
    apt:
      update_cache: yes
    when: ansible_distribution == "Ubuntu"

  - name: install apache2 package
    apt:
      name: apache2
      stae: latest
    when: ansible_distribution == "Ubuntu"

  - name: add PHP support for apache
    apt:
      name: libapache2-mod-php
      state: latest
    when: ansible_distribution == "Ubuntu"

  - name: update repository index
    dnf:
      update_cache: yes
    when: ansible_distribution == "CentOS"

  - name: install apache2 package
    dnf:
      name: httpd
      state: latest
    when: ansible_distribution == "CentOS"

  - name: add PHP support for apache
    dnf:
      name: php
      state: latest
    when: ansible_distribution == "CentOS"
```

Make sure to save and exit.

```
  GNU nano 6.2                                        install_apache.yml
---
- hosts: all
  become: true
  tasks:

  - name: update repository index
    apt:
      update_cache: yes
    when: ansible_distribution == "Ubuntu"

  - name: install apache2 package
    apt:
      name: apache2
      state: latest
    when: ansible_distribution == "Ubuntu"

  - name: add PHP support for apache
    apt:
      name: libapache2-mod-php
      state: latest
    when: ansible_distribution == "Ubuntu"

  - name: update repository index
    dnf:
      update_cache: yes
    when: ansible_distribution == "CentOS"

  - name: install apache2 package
    dnf:
      name: httpd
      state: latest
    when: ansible_distribution == "CentOS"

  - name: add PHP support for apache
    dnf:
      name: php
                                                    [ Read 38 lines ]
```

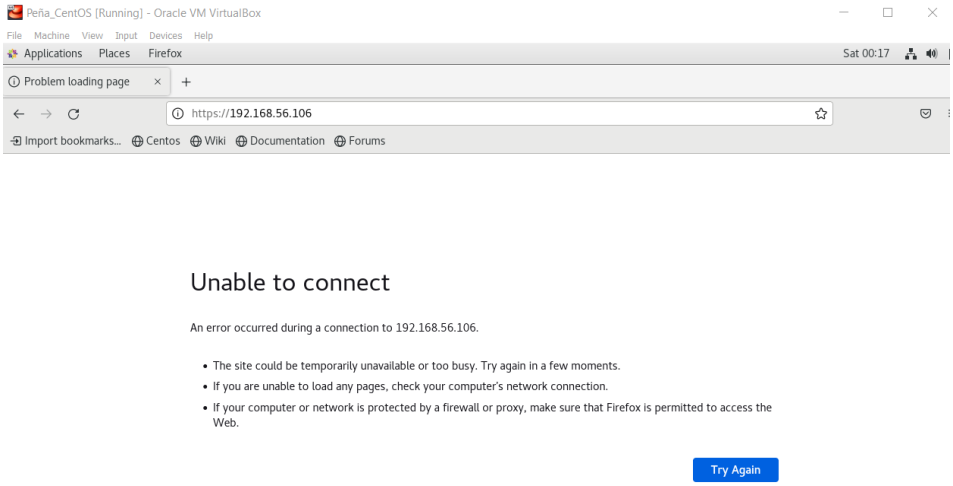
Run *ansible-playbook --ask-become-pass install_apache.yml* and describe the result.

```
ok: [192.168.1.18]
ok: [192.168.1.19]
ok: [192.168.56.106]

TASK [update repository index] ***************************************************************************************************
skipping: [192.168.56.106]
changed: [192.168.1.19]
changed: [192.168.1.18]

TASK [install apache2 package] ***************************************************************************************************
skipping: [192.168.56.106]
ok: [192.168.1.18]
ok: [192.168.1.19]

TASK [add PHP support for apache] ************************************************************************************************
skipping: [192.168.56.106]
ok: [192.168.1.18]
ok: [192.168.1.19]

TASK [update repository index] ***************************************************************************************************
skipping: [192.168.1.18]
skipping: [192.168.1.19]
ok: [192.168.56.106]

TASK [install apache2 package] ***************************************************************************************************
skipping: [192.168.1.18]
skipping: [192.168.1.19]
ok: [192.168.56.106]

TASK [add PHP support for apache] ************************************************************************************************
skipping: [192.168.1.18]
skipping: [192.168.1.19]
ok: [192.168.56.106]

PLAY RECAP ***********************************************************************************************************************
192.168.1.18               : ok=4    changed=1    unreachable=0    failed=0    skipped=3    rescued=0    ignored=0
192.168.1.19               : ok=4    changed=1    unreachable=0    failed=0    skipped=3    rescued=0    ignored=0
192.168.56.106             : ok=4    changed=0    unreachable=0    failed=0    skipped=3    rescued=0    ignored=0

bernice@Workstation-Pena:~/CPE232_BernicePena$
```

It successfully ran on the three hosts, but there were three skipped in the CentOS since those were meant for Ubuntu servers only. There is a conditional which is the “dnf”, this caused the skipped for the CentOS.

5. To verify the installations, go to CentOS VM and type its IP address on the browser. Was it successful? The answer is no. It’s because the httpd service or

the Apache HTTP server in the CentOS is not yet active. Thus, you need to activate it first.



5.1 To activate, go to the CentOS VM terminal and enter the following:
systemctl status httpd
The result of this command tells you that the service is inactive.

```
[bernice@pena_Workstation ~]$ systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; vendor prese
t: disabled)
   Active: inactive (dead)
     Docs: man:httpd(8)
           man:apachectl(8)
[bernice@pena_Workstation ~]$
```

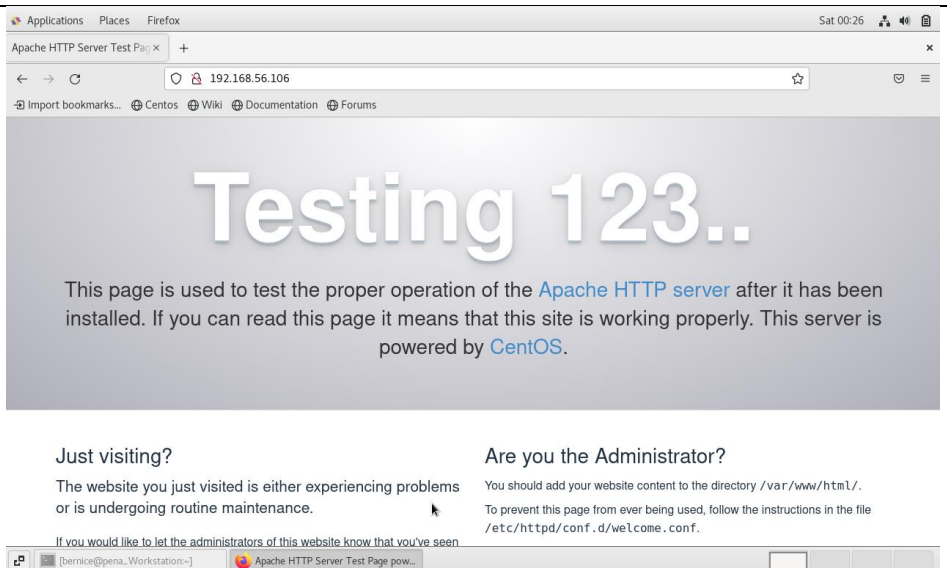
5.2 Issue the following command to start the service:
sudo systemctl start httpd

```
[bernice@pena Workstation ~]$ sudo systemctl start httpd
```

(When prompted, enter the sudo password)
sudo firewall-cmd --add-port=80/tcp
(The result should be a success)

```
[bernice@pena_Workstation ~]$ sudo sudo firewall-cmd --add-port=80/tcp
success
```

5.3 To verify the service is already running, go to CentOS VM and type its IP address on the browser. Was it successful? (Screenshot the browser)



After the activation, I was able to run the ip address successfully.

Task 2: Refactoring playbook

This time, we want to make sure that our playbook is efficient and that the codes are easier to read. This will also makes run ansible more quickly if it has to execute fewer tasks to do the same thing.

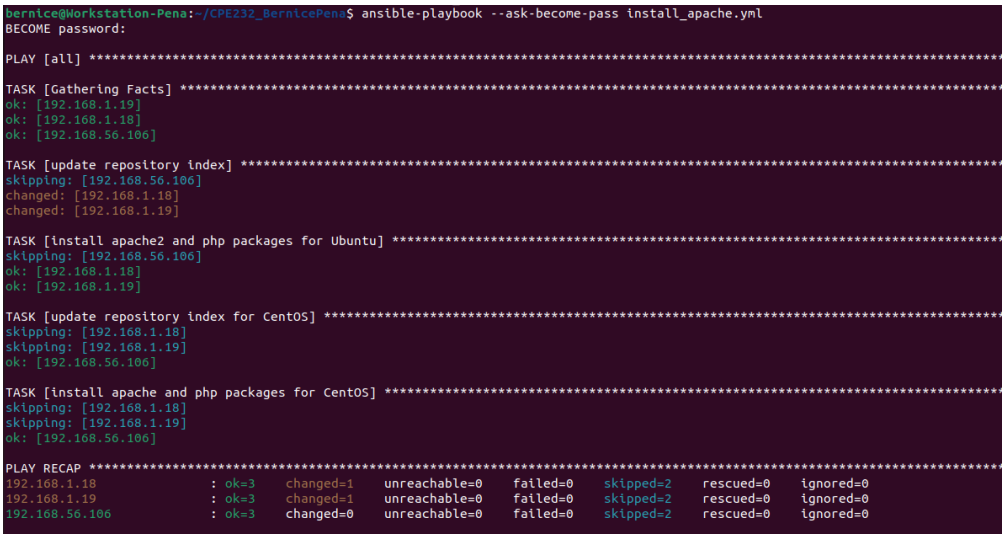
1. Edit the playbook *install_apache.yml*. Currently, we have three tasks targeting our Ubuntu machines and 3 tasks targeting our CentOS machine. Right now, we try to consolidate some tasks that are typically the same. For example, we can consolidate two plays that install packages. We can do that by creating a list of installation packages as shown below:

```
---
- hosts: all
  become: true
  tasks:

  - name: update repository index Ubuntu
    apt:
      update_cache: yes
    when: ansible_distribution == "Ubuntu"

  - name: install apache2 and php packages for Ubuntu
    apt:
      name:
        - apache2
        - libapache2-mod-php
      state: latest
    when: ansible_distribution == "Ubuntu"

  - name: update repository index for CentOS
    dnf:
      update_cache: yes
    when: ansible_distribution == "CentOS"

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

Make sure to save the file and exit.

```
  GNU nano 6.2
---
- hosts: all
  become: true
  tasks:

  - name: update repository index
    apt:
      update_cache: yes
    when: ansible_distribution == "Ubuntu"

  - name: install apache2 and php packages for Ubuntu
    apt:
      name:
        - apache2
        - libapache2-mod-php
      state: latest
    when: ansible_distribution == "Ubuntu"

  - name: update repository index for CentOS
    dnf:
      update_cache: yes
    when: ansible_distribution == "CentOS"

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

Run *ansible-playbook --ask-become-pass install_apache.yml* and describe the result.



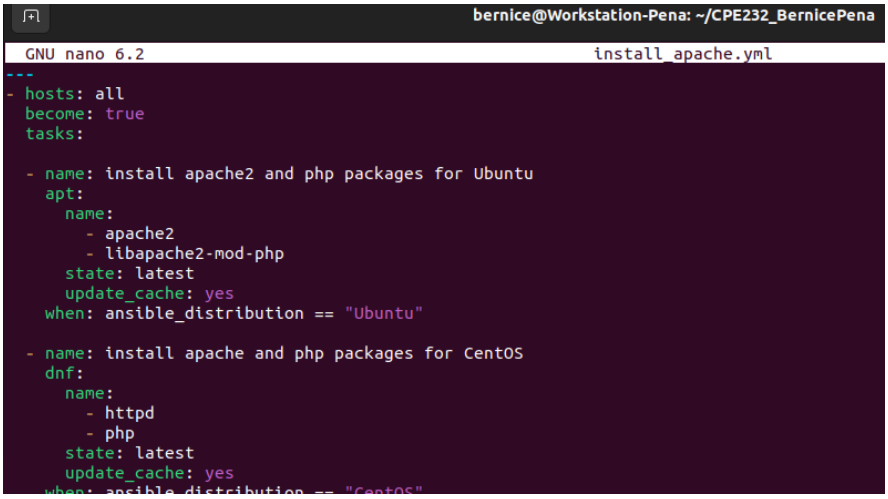
There were changes made on the Ubuntu servers but no changes was made on the CentOS

2. Edit the playbook *install_apache.yml* again. In task 2.1, we consolidated the plays into one play. This time we can actually consolidated everything in just 2 plays. This can be done by removing the update repository play and putting the

command *update_cache: yes* below the command *state: latest*. See below for reference:

```
---
- hosts: all
  become: true
  tasks:

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

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

Make sure to save the file and exit.



Run *ansible-playbook --ask-become-pass install_apache.yml* and describe the result.

```
bernice@Workstation-Pena:~/CPE232_BerniePena$ ansible-playbook --ask-become-pass install_apache.yml
BECOME password:

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

TASK [Gathering Facts] *********************************************************
ok: [192.168.1.19]
ok: [192.168.1.18]
ok: [192.168.56.106]

TASK [install apache2 and php packages for Ubuntu] *****************************
skipping: [192.168.56.106]
ok: [192.168.1.18]
ok: [192.168.1.19]

TASK [install apache and php packages for CentOS] ******************************
skipping: [192.168.1.18]
skipping: [192.168.1.19]
ok: [192.168.56.106]

PLAY RECAP *********************************************************************
192.168.1.18               : ok=2    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
192.168.1.19               : ok=2    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
192.168.56.106             : ok=2    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
```

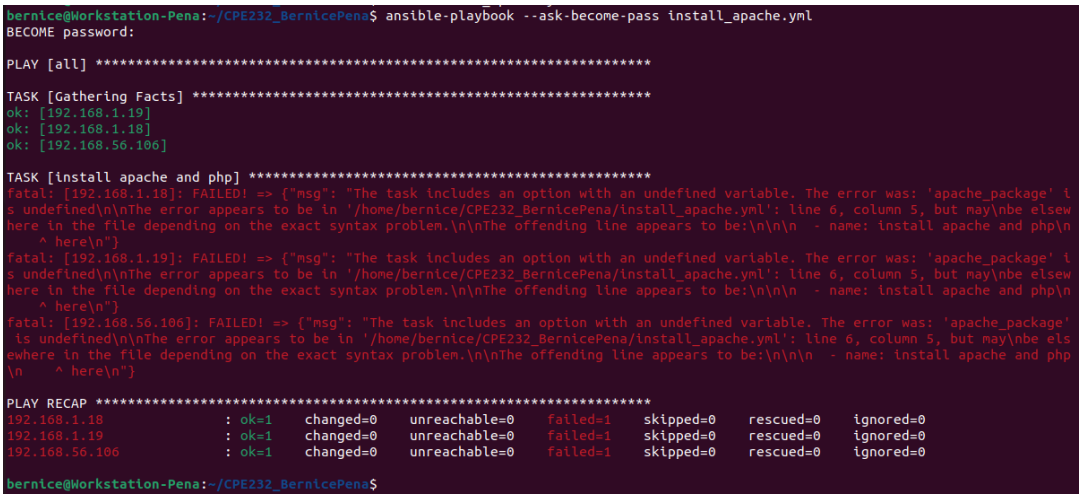
This shows the installation of apache and php for ubuntu servers and CentOS, the IP of my CentOS was once again skipped, it seems like it doesn't match the criteria specified in the playbook.

3. Finally, we can consolidate these 2 plays in just 1 play. This can be done by declaring variables that will represent the packages that we want to install. Basically, the apache_package and php_package are variables. The names are arbitrary, which means we can choose different names. We also take out the line when: ansible_distribution. Edit the playbook *install_apache.yml* again and make sure to follow the below image. Make sure to save the file and exit.

```
---
- hosts: all
  become: true
  tasks:

  - name: install apache and php
    apt:
      name:
        - "{{ apache_package }}"
        - "{{ php_package }}"
      state: latest
      update_cache: yes
```

```
 GNU nano 6.2                    install_apache.yml
---
- hosts: all
  become: true
  tasks:

  - name: install apache and php
    apt:
      name:
        - "{{ apache_package }}"
        - "{{ php_package }}"
      state: latest
      update_cache: yes
```

Run *ansible-playbook --ask-become-pass install_apache.yml* and describe the result.

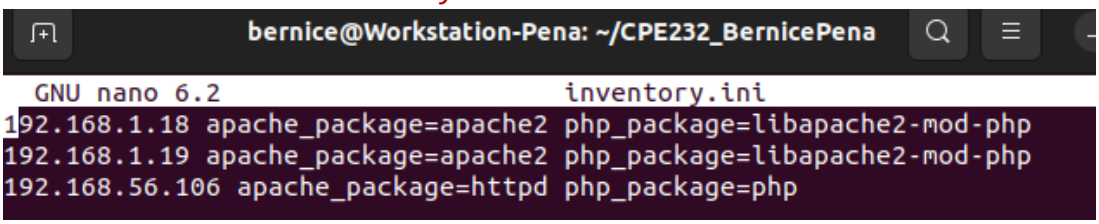


Failure occurred for the Ubuntu servers and CentOS

4. Unfortunately, task 2.3 was not successful. It's because we need to change something in the inventory file so that the variables we declared will be in place. Edit the *inventory* file and follow the below configuration:

```
192.168.56.120 apache_package=apache2 php_package=libapache2-mod-php
192.168.56.121 apache_package=apache2 php_package=libapache2-mod-php
192.168.56.122 apache_package=httpd php_package=php
```

Make sure to save the *inventory* file and exit.



Finally, we still have one more thing to change in our *install_apache.yml* file. In task 2.3, you may notice that the package is assign as apt, which will not run in CentOS. Replace the *apt* with *package*. Package is a module in ansible that is generic, which is going to use whatever package manager the underlying host or the target server uses. For Ubuntu it will automatically use *apt*, and for CentOS it will automatically use *dnf*. Make sure to save the file and exit. For more details about the ansible package, you may refer to this documentation: ansible.builtin.package – Generic OS package manager — Ansible Documentation

```
bernice@Workstation-Pena: ~/CPE232_BernicePena
 GNU nano 6.2                    install_apache.yml *
---
- hosts: all
  become: true
  tasks:

  - name: install apache and php
    package:
      name:
        - "{{ apache_package }}"
        - "{{ php_package }}"
      state: latest
      update_cache: yes
```

Run *ansible-playbook --ask-become-pass install_apache.yml* and describe the result.

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

TASK [Gathering Facts] *********************************************************
ok: [192.168.1.19]
ok: [192.168.1.18]
ok: [192.168.56.106]

TASK [install apache and php] **************************************************
ok: [192.168.1.19]
ok: [192.168.1.18]
ok: [192.168.56.106]

PLAY RECAP *********************************************************************
192.168.1.18               : ok=2    changed=0    unreachable=0    failed=0    s
kipped=0    rescued=0    ignored=0
192.168.1.19               : ok=2    changed=0    unreachable=0    failed=0    s
kipped=0    rescued=0    ignored=0
192.168.56.106             : ok=2    changed=0    unreachable=0    failed=0    s
kipped=0    rescued=0    ignored=0
```

After modifying the install_apache, all three hosts successfully installed the packages.

Supplementary Activity:

1. Create a playbook that could do the previous tasks in Red Hat OS.

```
bernice@pena_Workstation:~/CPE232_BernicePena
File  Edit  View  Search  Terminal  Help
 GNU nano 2.3.1              File: inventory

192.168.1.19
192.168.1.18
192.168.56.106
```

This is where I created the inventory file on my CentOS and then I added the IP of my ubuntu servers and IP of my CentOS.

```
bernice@pena_Workstation:~/CPE232_BernicePena
File  Edit  View  Search  Terminal  Help
 GNU nano 2.3.1                          File: install_apache.yml

---
- hosts: all
  become: yes
  tasks:
    - name: update repository index
      yum:
        name: '*'
        state: latest
      when: ansible_distribution == "CentOS"

    - name: install apache2 package
      yum:
        name: httpd
        state: latest
      when: ansible_distribution == "CentOS"

    - name: add PHP support for apache
      yum:
        name: php
        state: latest
      when: ansible_distribution == "CentOS"
```

I created a playbook on my CentOS VM that will perform the previous tasks I did in my Ubuntu workstation.

```
BECOME password:

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

TASK [Gathering Facts] *********************************************************
ok: [192.168.1.18]
ok: [192.168.1.19]
ok: [192.168.56.106]

TASK [update repository index] *************************************************
skipping: [192.168.1.18]
skipping: [192.168.1.19]
changed: [192.168.56.106]

TASK [install apache2 package] *************************************************
skipping: [192.168.1.18]
skipping: [192.168.1.19]
ok: [192.168.56.106]

TASK [add PHP support for apache] **********************************************
skipping: [192.168.1.18]
skipping: [192.168.1.19]
ok: [192.168.56.106]

PLAY RECAP *********************************************************************
192.168.1.18               : ok=1    changed=0    unreachable=0    failed=0    skipped=3    rescued=0    ignored=0
192.168.1.19               : ok=1    changed=0    unreachable=0    failed=0    skipped=3    rescued=0    ignored=0
192.168.56.106             : ok=4    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

[bernice@pena_Workstation CPE232_BernicePena]$
```

This was the result after executing it, there was a change in my CentOS IP since the task in my playbook is meant to perform it on my CentOS VM.

Reflections:

Answer the following:

1. Why do you think refactoring of playbook codes is important?

   Refactoring of playbook offers a lot of benefits such as improving the code readability and reducing the complexity, it also makes the having better scalability. In the previous tasks, I was able to break down the codes into pieces reducing the complexity while doing it, this makes it easier for the user to modify or analyze the code since it is organized, troubleshooting will

be much easier if needed, refactoring makes an improvement and efficient management in Ansible playbooks.

2. When do we use the “when” command in playbook?
   The “when” command in playbook is used for conditionals when controlling task executions, it is used to identify what conditions a certain task should run or sometimes skipped. The purpose of this is to manage and control the tasks that are running on the playbook according to user’s desired function on a certain host making it more efficient and manageable, this offers flexibility especially when troubleshooting, adding, or adjusting something from the code.