Name: Daniela Marie D. Rabang	**Date Performed:** 09/18/2023
Course/Section: CPE232/CPE31S4	**Date Submitted:** 09/19/2023
Instructor: Dr. Jonathan V. Taylar	**Semester and SY:** 1st sem 2023-24

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?

```
daniela@workstation:~/CPE232_Rabang_HOA5$ git pull
Already up to date.
```

When the command git pull is run, the message “already up to date.” is being updated. Therefore, the directory is already pulled from the github.

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 2.9.3                                inventory

192.168.56.110 ansible_python_interpreter=/usr/bin/python3
192.168.56.112 ansible_python_interpreter=/usr/bin/python
```

```
daniela@workstation:~/CPE232_Rabang_HOA5$ ansible-playbook --ask-become-pass install_apache.yml
SUDO password:

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

TASK [Gathering Facts] *********************************************************************************
ok: [192.168.56.110]
ok: [192.168.56.112]

TASK [update repository index] *************************************************************************
fatal: [192.168.56.112]: FAILED! => {"changed": false, "cmd": "apt-get update", "msg": "[Errno 2] No suc
h file or directory", "rc": 2}
changed: [192.168.56.110]

TASK [install apache2 package] *************************************************************************
ok: [192.168.56.110]

TASK [add PHP support for apache] **********************************************************************
ok: [192.168.56.110]
        to retry, use: --limit @/home/daniela/CPE232_Rabang_HOA5/install_apache.retry

PLAY RECAP *********************************************************************************************
192.168.56.110             : ok=4    changed=1    unreachable=0    failed=0
192.168.56.112             : ok=1    changed=0    unreachable=0    failed=1
```

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.

```
 GNU nano 2.9.3                          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
    when: ansible_distribution == "Ubuntu"

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

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

```
daniela@workstation:~/CPE232_Rabang_HOA5$ ansible-playbook --ask-become-pass install_apache.yml
SUDO password:

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

TASK [Gathering Facts] *********************************************************************************
ok: [192.168.56.110]
ok: [192.168.56.112]

TASK [update repository index] *************************************************************************
skipping: [192.168.56.112]
changed: [192.168.56.110]

TASK [install apache2 package] *************************************************************************
skipping: [192.168.56.112]
ok: [192.168.56.110]

TASK [add PHP support for apache] **********************************************************************
skipping: [192.168.56.112]
ok: [192.168.56.110]

PLAY RECAP *********************************************************************************************
192.168.56.110             : ok=4    changed=1    unreachable=0    failed=0
192.168.56.112             : ok=1    changed=0    unreachable=0    failed=0
```

After running the command above, it outputs skipping since I had put the condition when ansible_distribution == “Ubuntu”. So when the system detects that it is not Ubuntu then the process skips.

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.

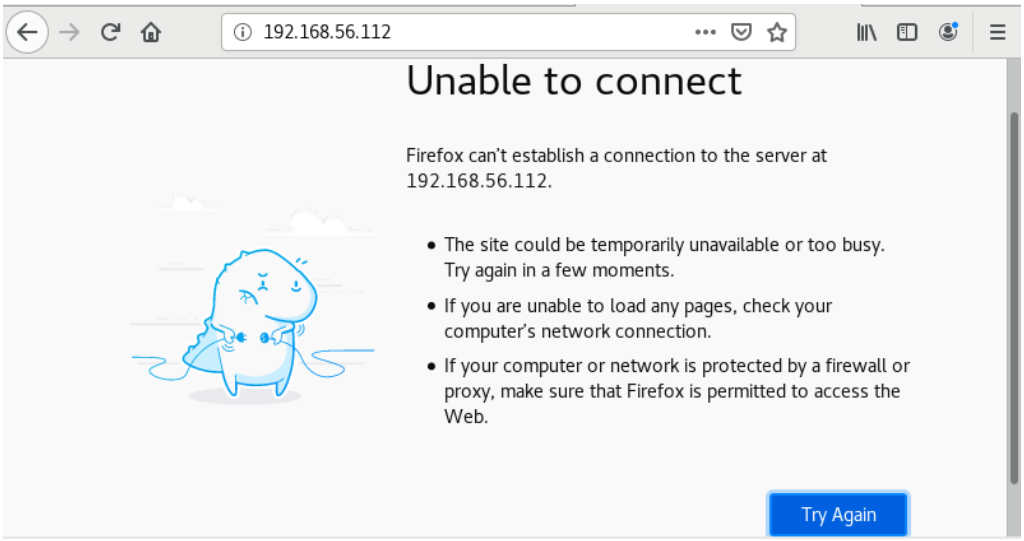
```
 GNU nano 2.9.3                                              install_apache.yml

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

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

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

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

  - name: update repository index
    yum:
      name: "*"
      update_cache: yes
    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"
```

```
daniela@workstation:~/CPE232_Rabang_HOA5$ ansible-playbook --ask-become-pass install_apache.yml
SUDO password:

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

TASK [Gathering Facts] *********************************************************************
ok: [192.168.56.112]
ok: [192.168.56.110]

TASK [update repository index] *************************************************************
skipping: [192.168.56.112]
changed: [192.168.56.110]

TASK [install apache2 package] *************************************************************
skipping: [192.168.56.112]
ok: [192.168.56.110]

TASK [add PHP support for apache] **********************************************************
skipping: [192.168.56.112]
ok: [192.168.56.110]

TASK [update repository index] *************************************************************
skipping: [192.168.56.110]
ok: [192.168.56.112]

TASK [install apache2 package] *************************************************************
skipping: [192.168.56.110]
ok: [192.168.56.112]

TASK [add php support for apache] **********************************************************
skipping: [192.168.56.110]
ok: [192.168.56.112]

PLAY RECAP *********************************************************************************
192.168.56.110             : ok=4    changed=1    unreachable=0    failed=0
192.168.56.112             : ok=4    changed=0    unreachable=0    failed=0
```

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

The results show that the following tasks are done and changed. It updated the repository index on both Ubuntu and CentOS same as the installation of apache2, and added PHP support for apache.

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.



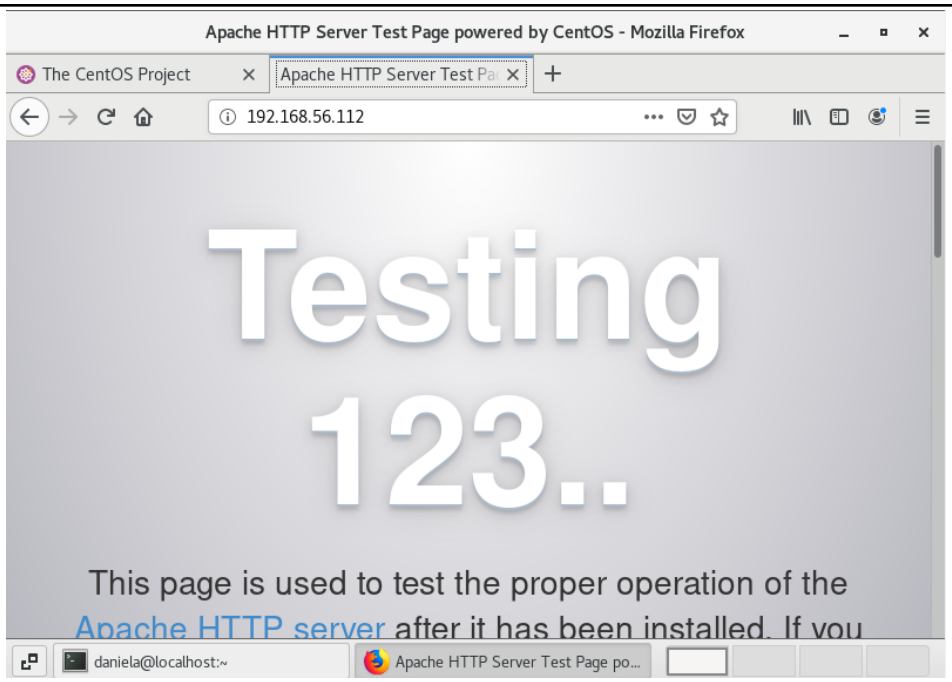
5.1 To activate, go to the CentOS VM terminal and enter the following:
systemctl status httpd
The result of this command tells you that the service is inactive.

```
[daniela@localhost ~]$ 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)
```

5.2 Issue the following command to start the service:
sudo systemctl start httpd
(When prompted, enter the sudo password)
sudo firewall-cmd --add-port=80/tcp
(The result should be a success)

```
[daniela@localhost ~]$ sudo systemctl start httpd
[daniela@localhost ~]$ 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)



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 2.9.3                    install_apache.yml                    Modified

#    when: ansible_distribution == "CentOS"

---
- 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
    yum:
      name: "*"
      update_cache: yes
    when: ansible_distribution == "CentOS"

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

```
daniela@workstation:~/CPE232_Rabang_HOA5$ ansible-playbook --ask-become-pass install_apache.yml
SUDO password:

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

TASK [Gathering Facts] *********************************************************************************************
ok: [192.168.56.110]
ok: [192.168.56.112]

TASK [update repository index Ubuntu] ******************************************************************************
skipping: [192.168.56.112]
changed: [192.168.56.110]

TASK [install apache2 and php packages for Ubuntu] *****************************************************************
skipping: [192.168.56.112]
ok: [192.168.56.110]

TASK [update repository index for CentOS] **************************************************************************
skipping: [192.168.56.110]
ok: [192.168.56.112]

TASK [install apache and php packages for CentOS] ******************************************************************
skipping: [192.168.56.110]
ok: [192.168.56.112]

PLAY RECAP *********************************************************************************************************
192.168.56.110             : ok=3    changed=1    unreachable=0    failed=0
192.168.56.112             : ok=3    changed=0    unreachable=0    failed=0
```

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

The results gave the list of tasks that are being run and it is completed like the last one.

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"
```

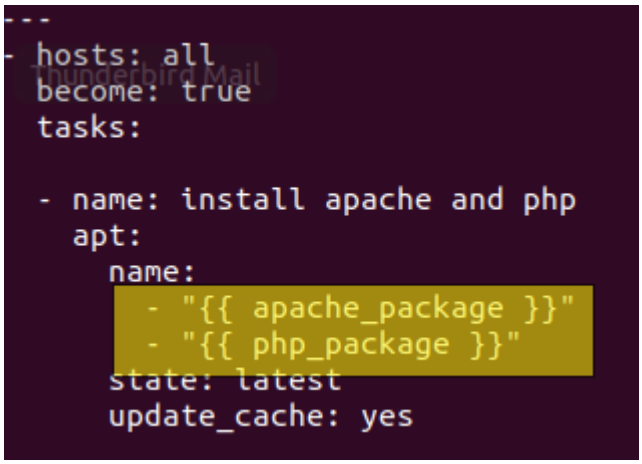
Make sure to save the file and exit.

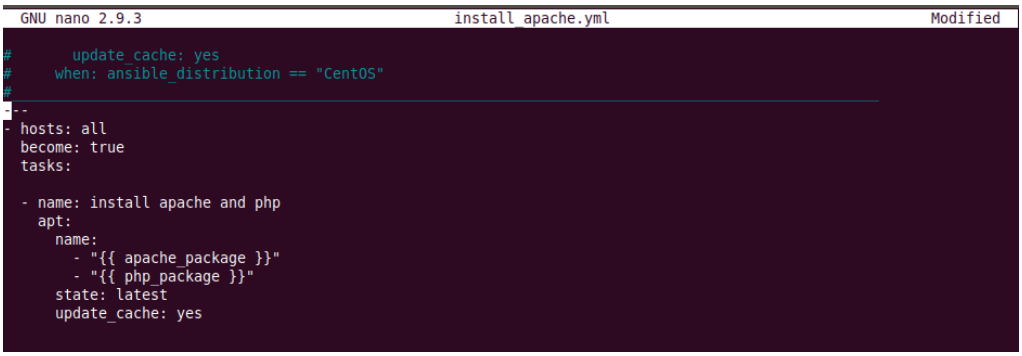
```
 GNU nano 2.9.3                    install_apache.yml                    Modified


---
- 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
     yum:
       name:
         - httpd
         - php
       state: latest
       update_cache: yes
     when: ansible_distribution == "CentOS"
```

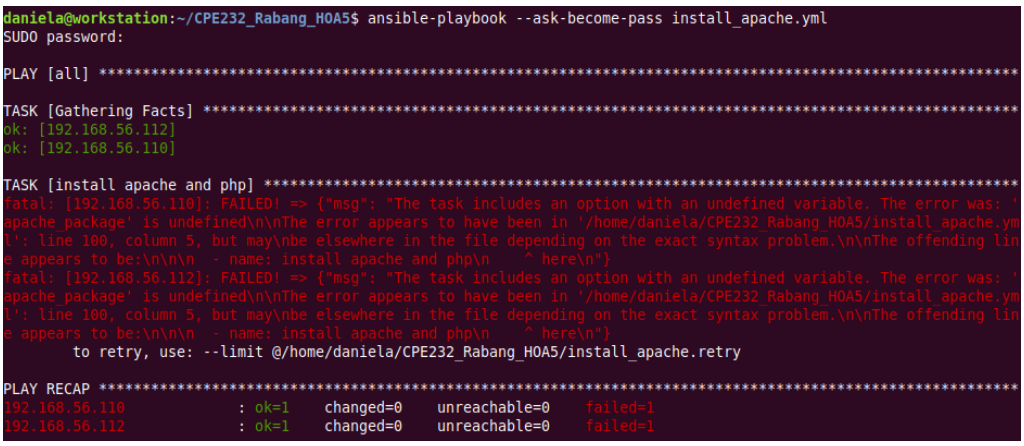
```
daniela@workstation:~/CPE232_Rabang_HOA5$ ansible-playbook --ask-become-pass install_apache.yml
SUDO password:

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

TASK [Gathering Facts] *********************************************************************************************
ok: [192.168.56.112]
ok: [192.168.56.110]

TASK [install apache2 and php packages for Ubuntu] *****************************************************************
skipping: [192.168.56.112]
ok: [192.168.56.110]

TASK [install apache and php packages for CentOS] ******************************************************************
skipping: [192.168.56.110]
ok: [192.168.56.112]

PLAY RECAP *********************************************************************************************************
192.168.56.110             : ok=2    changed=0    unreachable=0    failed=0
192.168.56.112             : ok=2    changed=0    unreachable=0    failed=0
```

Run *ansible-playbook --ask-become-pass install_apache.yml* and describe the result. After running the command above, It outputs that the following actions and tasks are being done when it detects the same distribution.

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.







Run *ansible-playbook --ask-become-pass install_apache.yml* and describe the result. When the command has finished running the output shows that there are errors and it is an expected error. This is because there are modifications that need to be done inside the inventory file.

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.

```
  GNU nano 2.9.3                                inventory

192.168.56.110 ansible_python_interpreter=/usr/bin/python3
#192.168.56.112 ansible_python_interpreter=/usr/bin/python

192.168.56.110 apache_package=apache2 php_package=libapache2-mod-php
192.168.56.112 apache_package=httpd php_package=php
```

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]

```
daniela@workstation:~/CPE232_Rabang_HOA5$ ansible-playbook --ask-become-pass install_apache.yml
SUDO password:

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

TASK [Gathering Facts] *********************************************************************************************
ok: [192.168.56.112]
ok: [192.168.56.110]

TASK [install apache and php] **************************************************************************************
ok: [192.168.56.112]
ok: [192.168.56.110]

PLAY RECAP *********************************************************************************************************
192.168.56.110             : ok=2    changed=0    unreachable=0    failed=0
192.168.56.112             : ok=2    changed=0    unreachable=0    failed=0
```

Run *ansible-playbook --ask-become-pass install_apache.yml* and describe the result. The following tasks are being installed since I already had changed something from the inventory file.

Supplementary Activity:

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

Reflections:

Answer the following:

1. Why do you think refactoring of playbook codes is important?
   - Refactoring code of playbook is important since it is used for improving the performance of the playbook when it is run. This makes the performance better by reducing the complexity of it.
2. When do we use the “when” command in the playbook?
   - The command is used in the playbook when the person wants to check if the tasks that are being runned is required. This checks it before it executes.

Conclusion:

In this activity, which is all about consolidating playbooks, this is where we are asked to run playbook codes from the yml file. Then every task is being asked to be run, from the start it needs to run the basic tasks for Ubuntu. When the system detects that the distribution of the control node is an Ubuntu, the tasks that are listed to be runned will be successful. Then we added the tasks that detect for CentOS. This is the same as the one for Ubuntu. But along the way of doing this activity I had encountered a lot of errors from all the codes that I inputted. Then there are some minor changes that need to be done in the inventory file, and yaml file. So that it can run successfully. Therefore, I had learned that most of the time the dnl is not supported for the CentOS rather I had to use the yum for it. I also learned that just small mistakes in the input will make an error.