Name: Cruz, Patrick Danielle C	**Date Performed: August 28, 2025**
Course/Section: CPE212 – CPE31S4	**Date Submitted: August 28,2025**
Instructor: Engr. Robin Valenzuela	**Semester and SY: 1st semester**

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**:

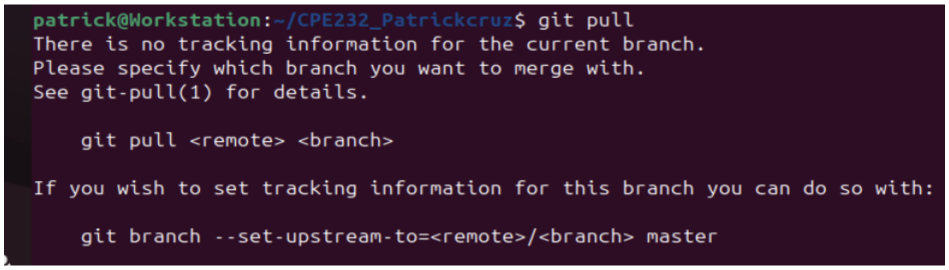
We are going to look at a way that we can differentiate a playbook by a host in terms of which distribution the host is running. It's very common in most Linux shops to run multiple distributions, for example, Ubuntu shop or Debian shop and you need a different distribution for a one off-case or perhaps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

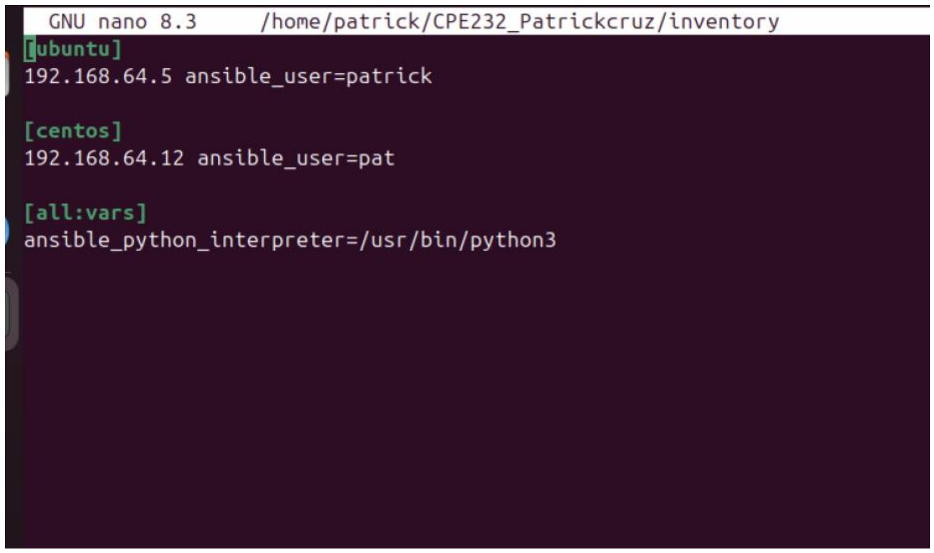
1. In the local machine, make sure you are in the local repository directory (*CPE232_yourname*). Issue the command git pull. When prompted, enter the correct passphrase or password. Describe what happened when you issue this command. Did something happen? Why?



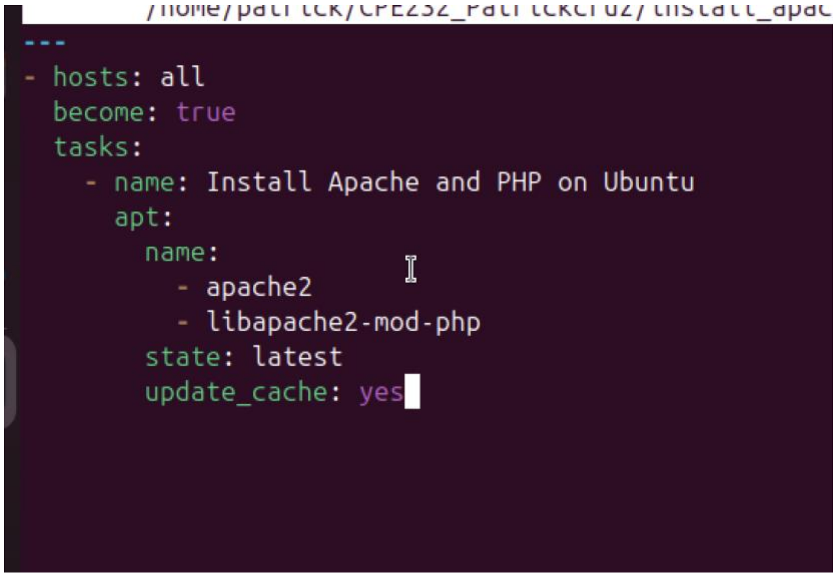
```
patrick@Workstation:~/CPE232_Patrickcruz$ git pull
There is no tracking information for the current branch.
Please specify which branch you want to merge with.
See git-pull(1) for details.

    git pull <remote> <branch>

If you wish to set tracking information for this branch you can do so with:

    git branch --set-upstream-to=<remote>/<branch> master
```

- **The git pull command failed with an error because the local branch exists but has not been linked, or set to track, a corresponding branch on the remote repository such as origin/main or origin/master. Without this link, Git does not know which remote branch it should pull changes from.**

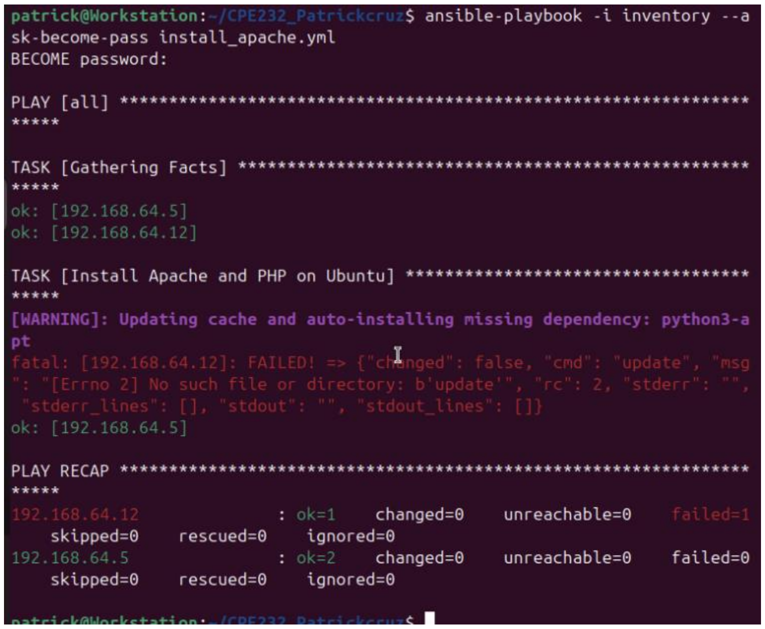
2. Edit the inventory file and add the IP address of the Centos VM. Issue the command we used to execute the playbook (the one we used in the last activity): *ansible-playbook --ask-become-pass install_apache.yml*. After executing this command, you may notice that it did not become successful in the 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 8.3       /home/patrick/CPE232_Patrickcruz/inventory
[ubuntu]
192.168.64.5 ansible_user=patrick

[centos]
192.168.64.12 ansible_user=pat

[all:vars]
ansible_python_interpreter=/usr/bin/python3
```

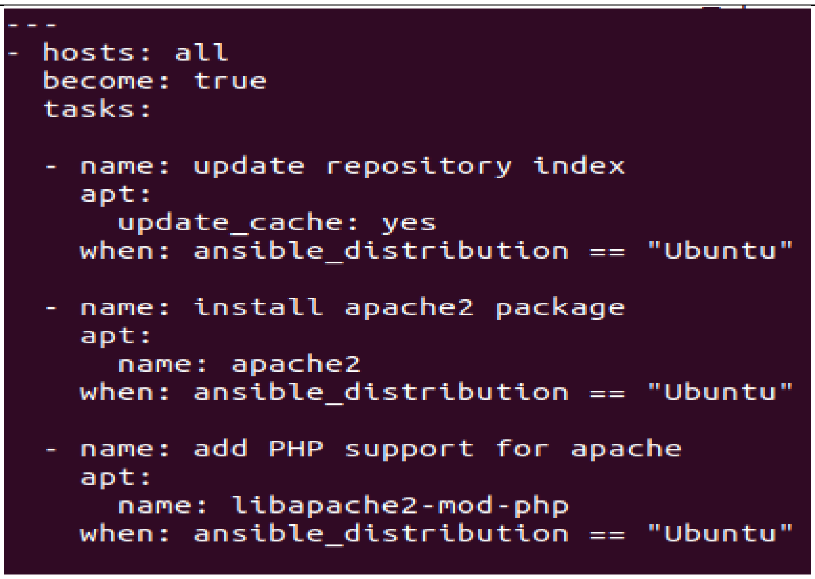


```
/home/patrick/CPE232_Patrickcruz/install_apac
---
- hosts: all
  become: true
  tasks:
    - name: Install Apache and PHP on Ubuntu
      apt:
        name:
          - apache2
          - libapache2-mod-php
        state: latest
        update_cache: yes
```

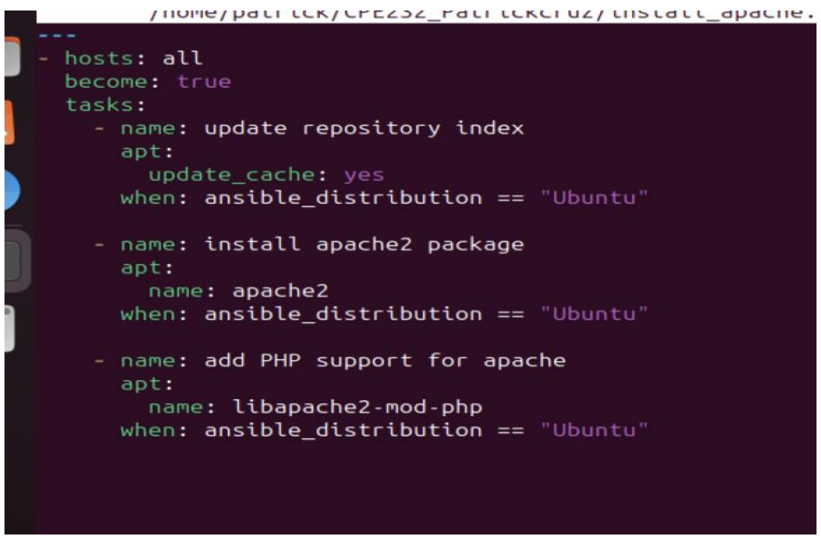


```
patrick@Workstation:~/CPE232_Patrickcruz$ ansible-playbook -i inventory --a
sk-become-pass install_apache.yml
BECOME password:

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

TASK [Gathering Facts] *********************************************************
*****
ok: [192.168.64.5]
ok: [192.168.64.12]

TASK [Install Apache and PHP on Ubuntu] ****************************************
*****
[WARNING]: Updating cache and auto-installing missing dependency: python3-a
pt
fatal: [192.168.64.12]: FAILED! => {"changed": false, "cmd": "update", "msg
": "[Errno 2] No such file or directory: b'update'", "rc": 2, "stderr": "",
 "stderr_lines": [], "stdout": "", "stdout_lines": []}
ok: [192.168.64.5]

PLAY RECAP *********************************************************************
*****
192.168.64.12              : ok=1    changed=0    unreachable=0    failed=1
    skipped=0    rescued=0    ignored=0
192.168.64.5               : ok=2    changed=0    unreachable=0    failed=0
    skipped=0    rescued=0    ignored=0

patrick@Workstation:~/CPE232_Patrickcruz$
```

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



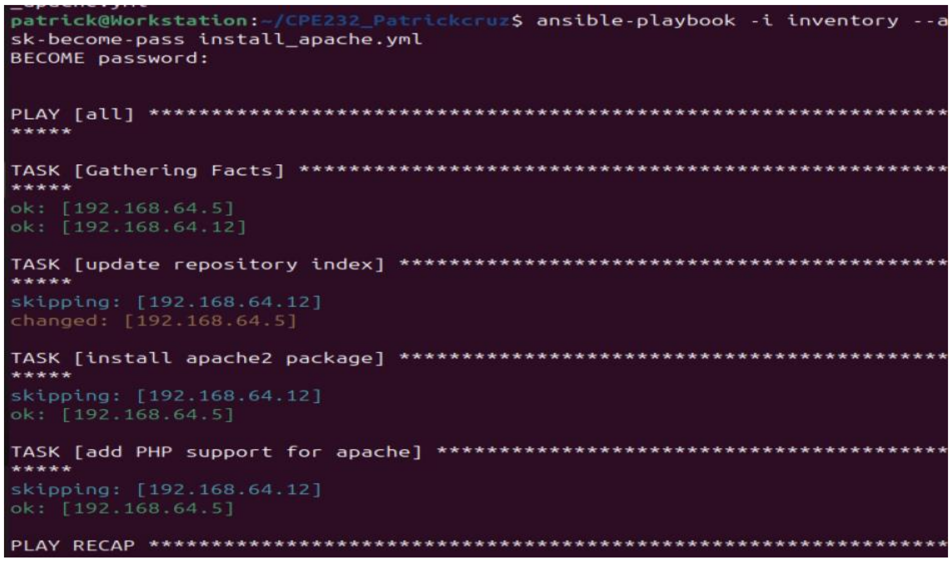
```
---
- hosts: all
  become: true
  tasks:

  - 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"
```



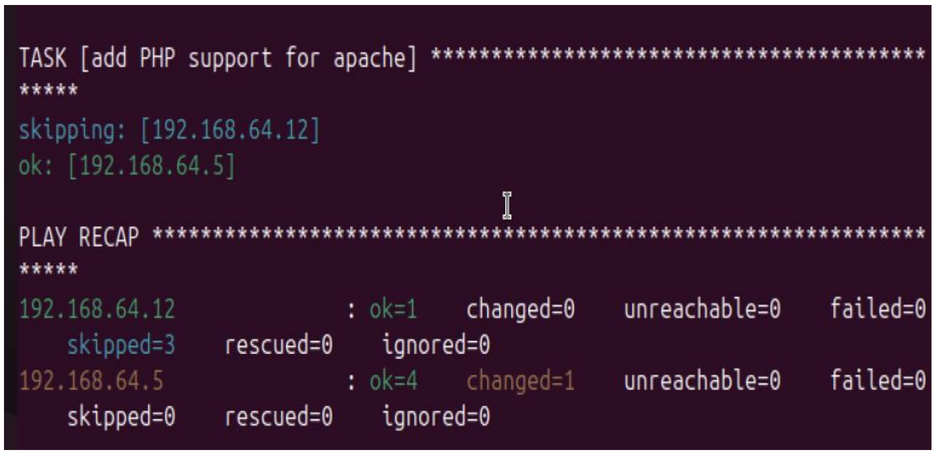
```
/home/patrick/CPE232_PatrickCruz/install_apache.
---
- hosts: all
  become: true
  tasks:
    - name: update repository index
      apt:
        update_cache: yes
      when: ansible_distribution == "Ubuntu"

    - 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.



```
patrick@Workstation:~/CPE232_Patrickcruz$ ansible-playbook -i inventory --a
sk-become-pass install_apache.yml
BECOME password:

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

TASK [Gathering Facts] ******************************************************
*****
ok: [192.168.64.5]
ok: [192.168.64.12]

TASK [update repository index] **********************************************
*****
skipping: [192.168.64.12]
changed: [192.168.64.5]

TASK [install apache2 package] **********************************************
*****
skipping: [192.168.64.12]
ok: [192.168.64.5]

TASK [add PHP support for apache] *******************************************
*****
skipping: [192.168.64.12]
ok: [192.168.64.5]

PLAY RECAP ******************************************************************
```



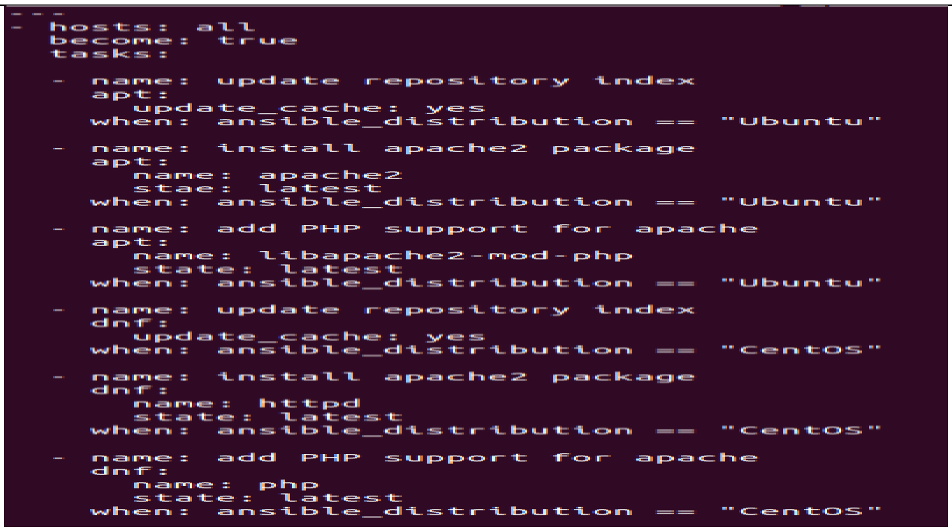
```
TASK [add PHP support for apache] *******************************************
*****
skipping: [192.168.64.12]
ok: [192.168.64.5]

PLAY RECAP ******************************************************************
*****
192.168.64.12              : ok=1    changed=0    unreachable=0    failed=0
    skipped=3    rescued=0    ignored=0
192.168.64.5               : ok=4    changed=1    unreachable=0    failed=0
    skipped=0    rescued=0    ignored=0
```

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

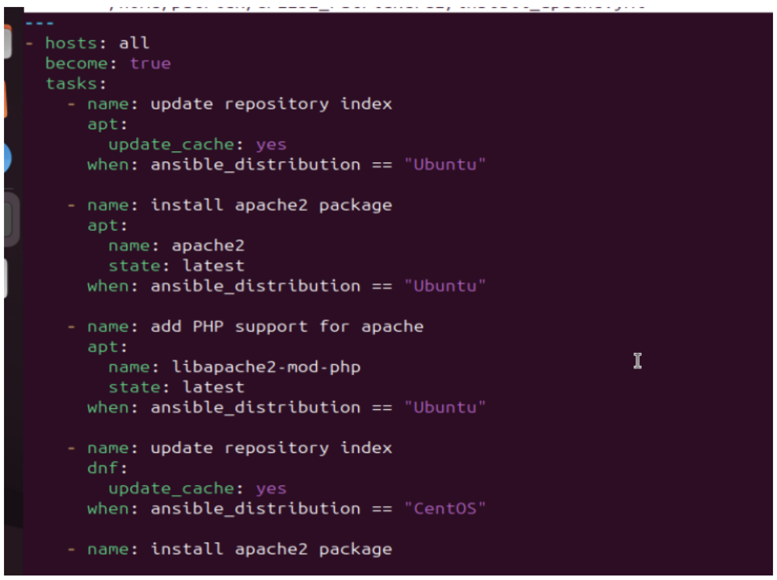
```
-   name: update repository index
    apt:
      update_cache: yes
    when: ansible_distribution in [“Debian”, “Ubuntu]
```

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

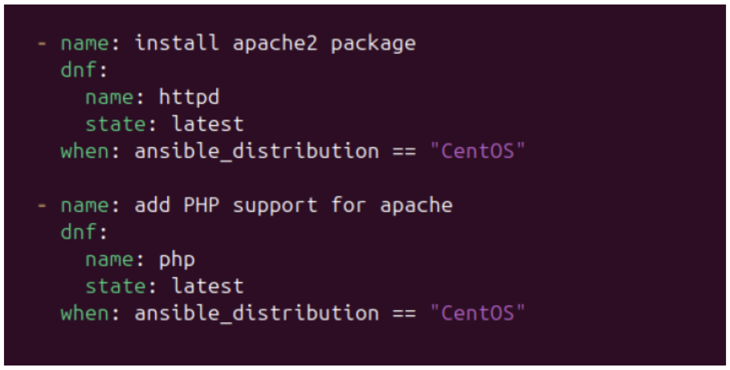
4. Edit the *install_apache.yml* file and insert the lines shown below.



```
---
- 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"
```



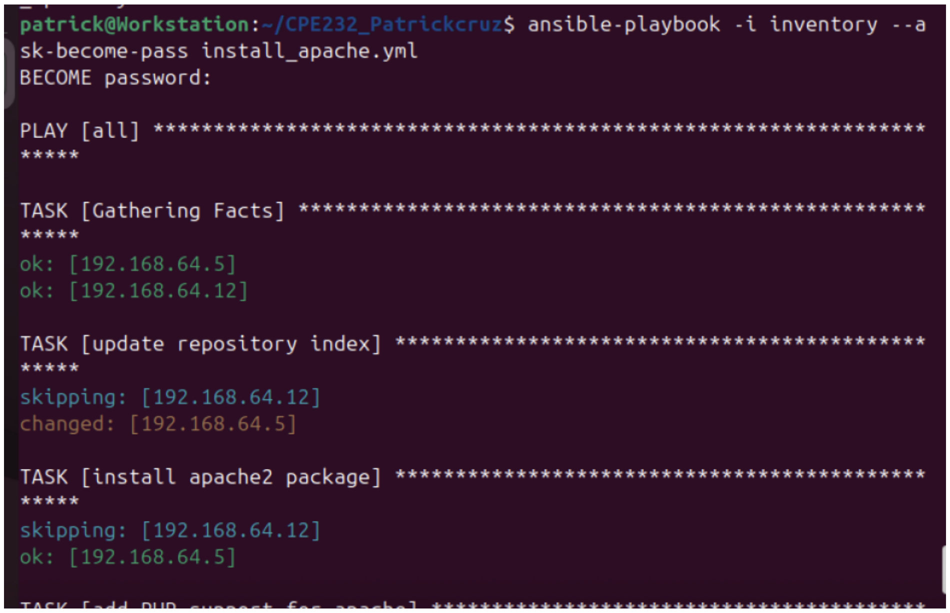
```
---
- hosts: all
  become: true
  tasks:
    - name: update repository index
      apt:
        update_cache: yes
      when: ansible_distribution == "Ubuntu"

    - 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
```



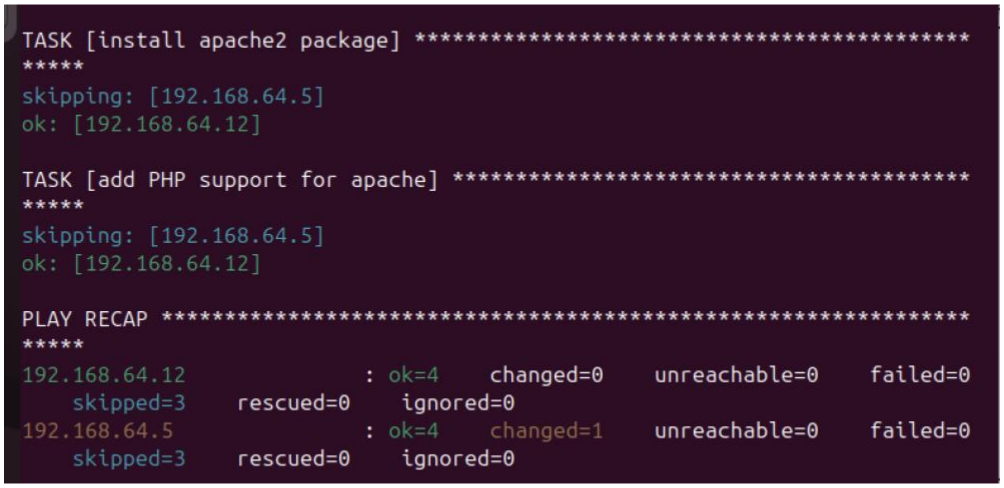
```
    - name: install apache2 package
      dnf:
        name: httpd
        state: latest
      when: ansible_distribution == "CentOS"

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

Make sure to save and exit.

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



```
patrick@Workstation:~/CPE232_Patrickcruz$ ansible-playbook -i inventory --a
sk-become-pass install_apache.yml
BECOME password:

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

TASK [Gathering Facts] *********************************************************
*****
ok: [192.168.64.5]
ok: [192.168.64.12]

TASK [update repository index] *************************************************
*****
skipping: [192.168.64.12]
changed: [192.168.64.5]

TASK [install apache2 package] *************************************************
*****
skipping: [192.168.64.12]
ok: [192.168.64.5]
```

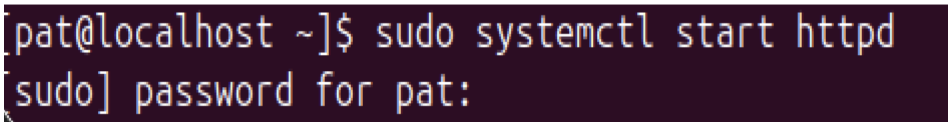


```
TASK [install apache2 package] ***********************************************
*****
skipping: [192.168.64.5]
ok: [192.168.64.12]

TASK [add PHP support for apache] ********************************************
*****
skipping: [192.168.64.5]
ok: [192.168.64.12]

PLAY RECAP *******************************************************************
*****
192.168.64.12              : ok=4    changed=0    unreachable=0    failed=0
    skipped=3    rescued=0    ignored=0
192.168.64.5               : ok=4    changed=1    unreachable=0    failed=0
    skipped=3    rescued=0    ignored=0
```

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.

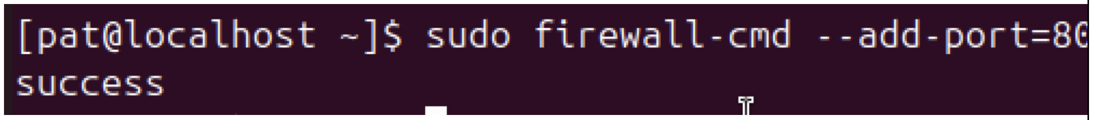
5.1 To activate, go to the CentOS VM terminal and enter the following:
systemctl status httpd
The result of this command tells you that the service is inactive.

5.2 Issue the following command to start the service:
sudo systemctl start httpd
(When prompted, enter the sudo password)



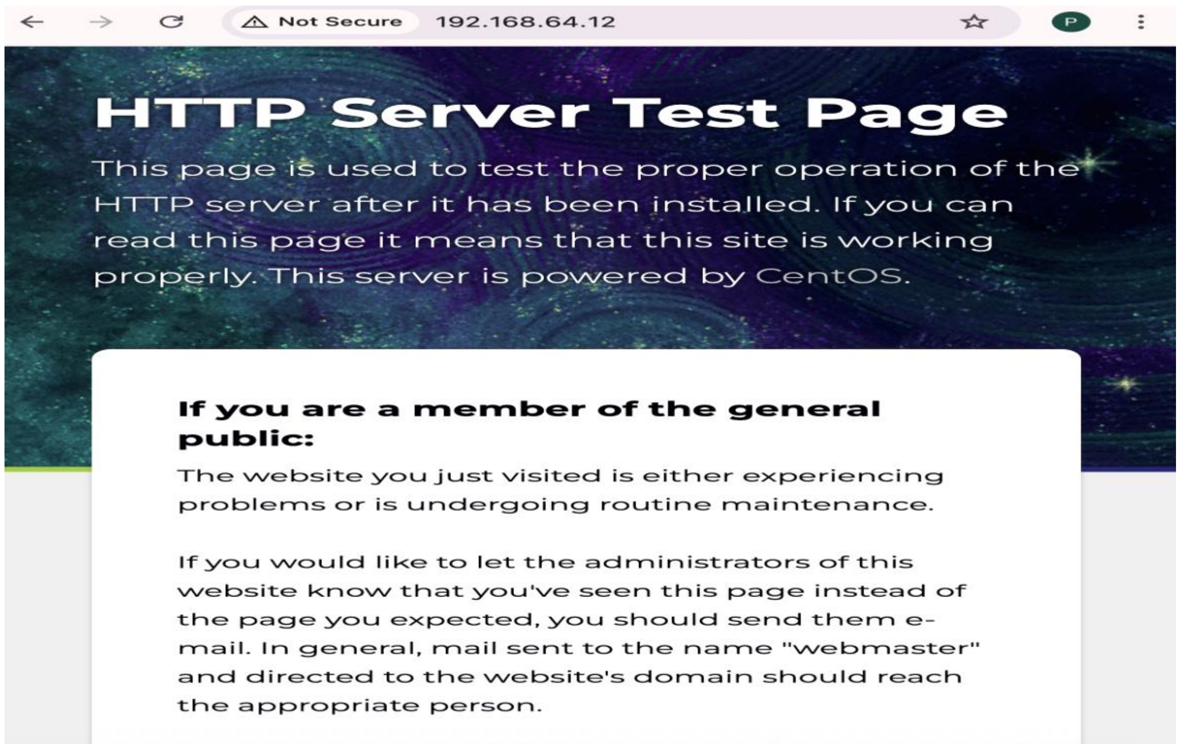
```
[pat@localhost ~]$ sudo systemctl start httpd
[sudo] password for pat:
```

sudo firewall-cmd --add-port=80/tcp
(The result should be a success)



```
[pat@localhost ~]$ sudo firewall-cmd --add-port=80
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)



HTTP Server Test Page

This page is used to test the proper operation of the HTTP server after it has been installed. If you can read this page it means that this site is working properly. This server is powered by CentOS.

If you are a member of the general public:

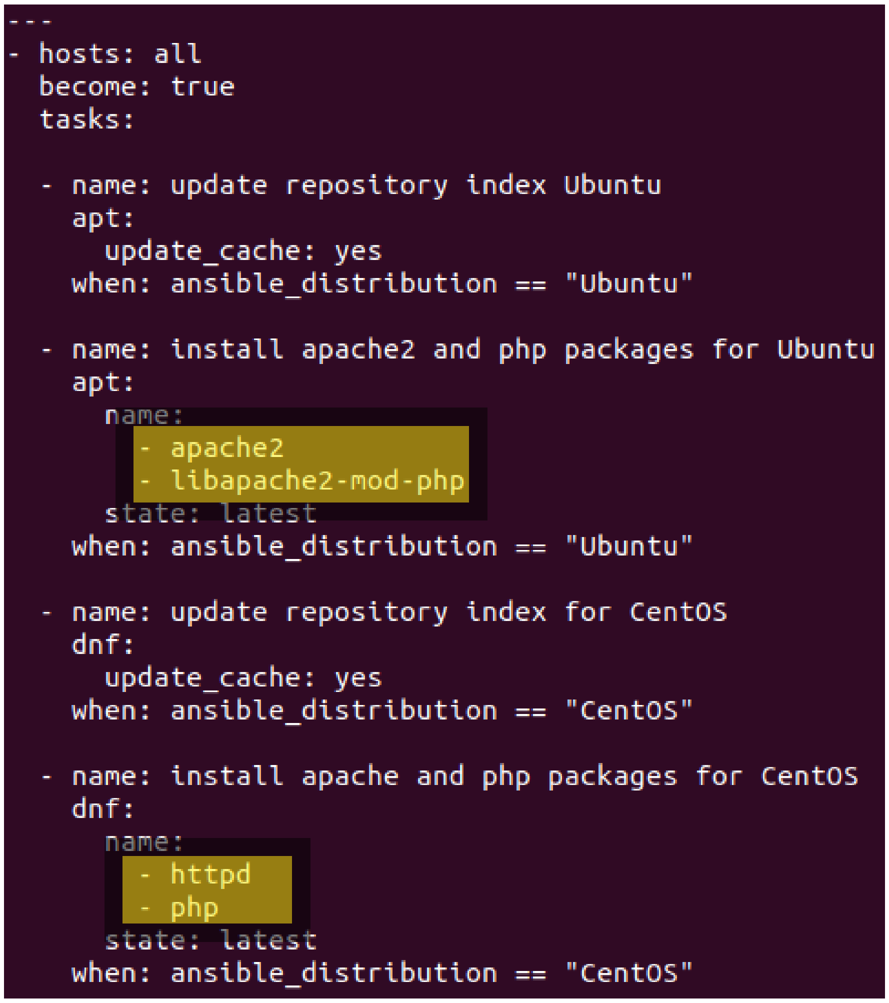
The website you just visited is either experiencing problems or is undergoing routine maintenance.

If you would like to let the administrators of this website know that you've seen this page instead of the page you expected, you should send them e-mail. In general, mail sent to the name "webmaster" and directed to the website's domain should reach the appropriate person.

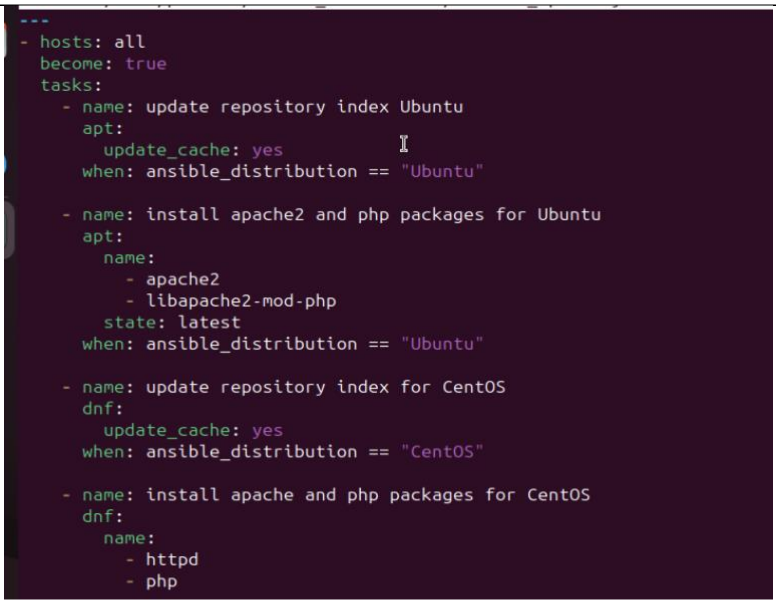
Task 2: Refactoring playbook

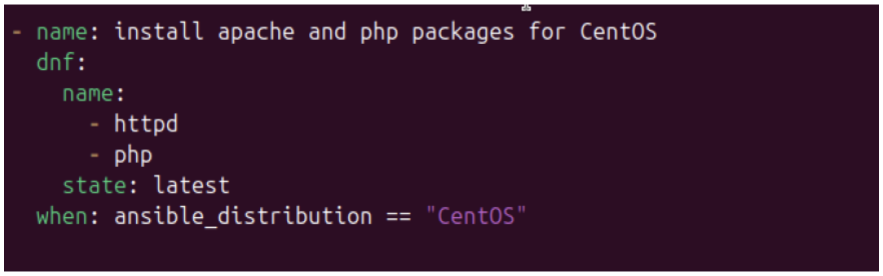
This time, we want to make sure that our playbook is efficient and that the codes are easier to read. This will also makes run ansible more quickly if it has to execute fewer tasks to do the same thing.

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:



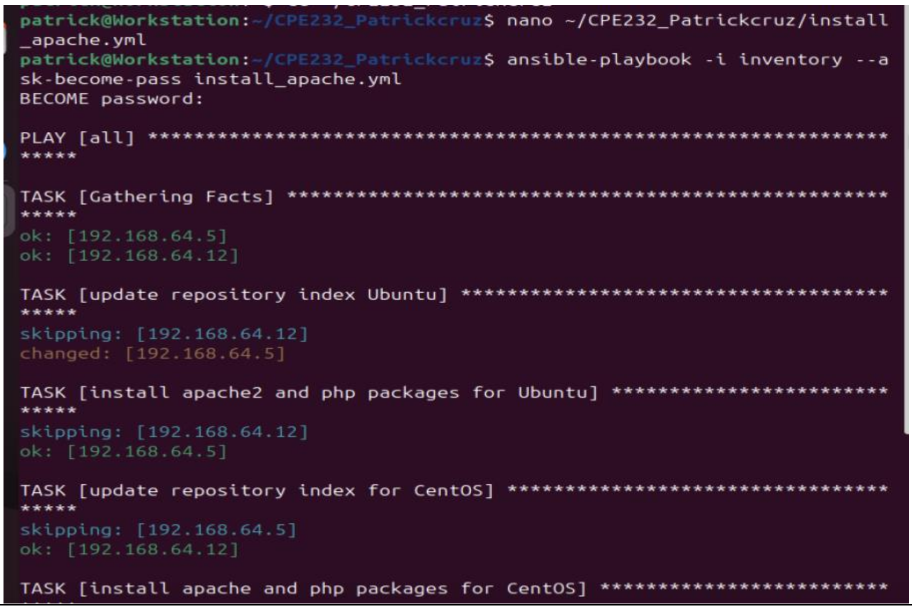
```
---
- hosts: all
  become: true
  tasks:

  - 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"
```

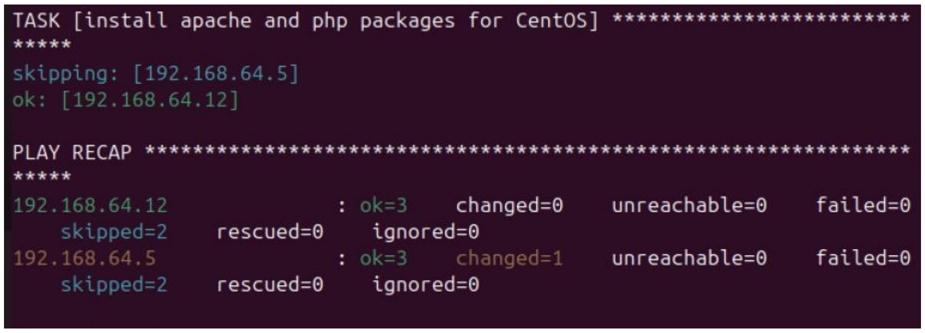

```
---
- hosts: all
  become: true
  tasks:
    - name: update repository index Ubuntu
      apt:
        update_cache: yes
      when: ansible_distribution == "Ubuntu"

    - 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
```


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

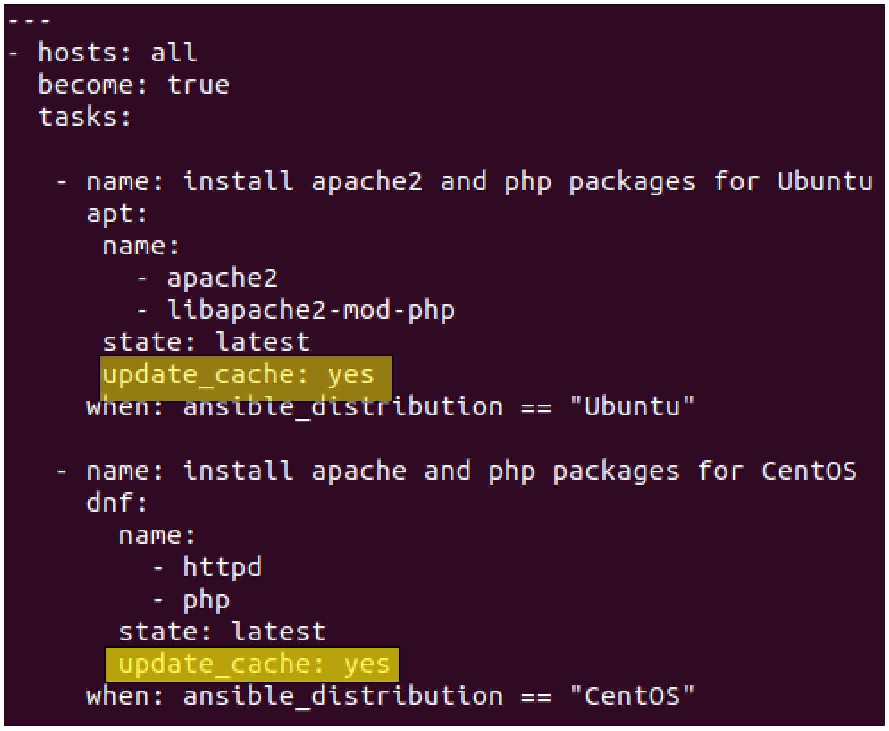
Make sure to save the file and exit.

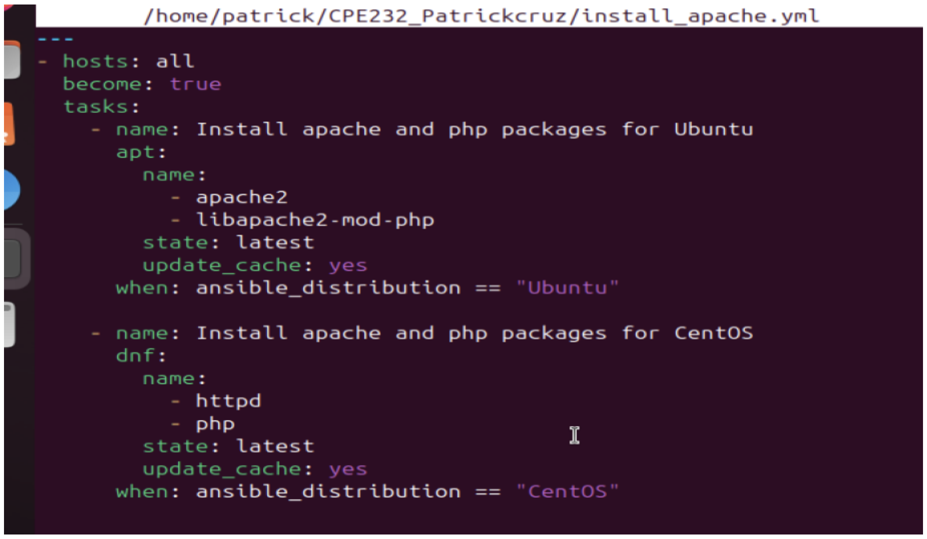
Run *ansible-playbook --ask-become-pass install_apache.yml* and describe the result.


```
patrick@Workstation:~/CPE232_Patrickcruz$ nano ~/CPE232_Patrickcruz/install
_apache.yml
patrick@Workstation:~/CPE232_Patrickcruz$ ansible-playbook -i inventory --a
sk-become-pass install_apache.yml
BECOME password:

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

TASK [Gathering Facts] *****************************************************
*****
ok: [192.168.64.5]
ok: [192.168.64.12]

TASK [update repository index Ubuntu] **************************************
*****
skipping: [192.168.64.12]
changed: [192.168.64.5]

TASK [install apache2 and php packages for Ubuntu] *************************
*****
skipping: [192.168.64.12]
ok: [192.168.64.5]

TASK [update repository index for CentOS] **********************************
*****
skipping: [192.168.64.5]
ok: [192.168.64.12]

TASK [install apache and php packages for CentOS] **************************
```


```
TASK [install apache and php packages for CentOS] *************************
*****
skipping: [192.168.64.5]
ok: [192.168.64.12]

PLAY RECAP ****************************************************************
*****
192.168.64.12              : ok=3    changed=0    unreachable=0    failed=0
    skipped=2    rescued=0    ignored=0
192.168.64.5               : ok=3    changed=1    unreachable=0    failed=0
    skipped=2    rescued=0    ignored=0
```

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:

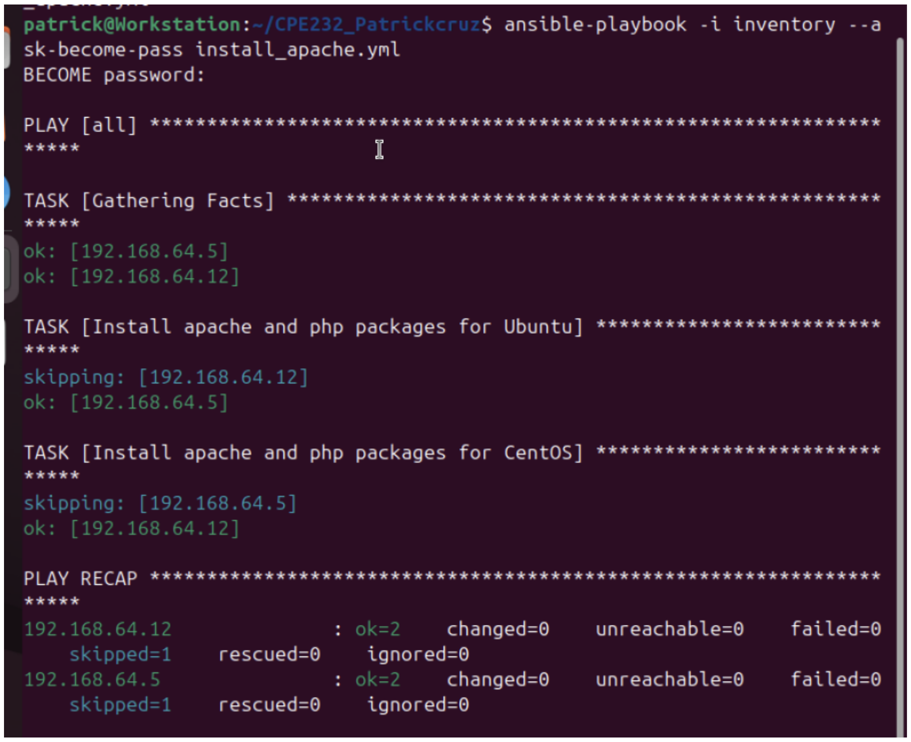

```
---
- hosts: all
  become: true
  tasks:

    - 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"
```



```
/home/patrick/CPE232_Patrickcruz/install_apache.yml
---
- hosts: all
  become: true
  tasks:
    - name: Install apache 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"
```

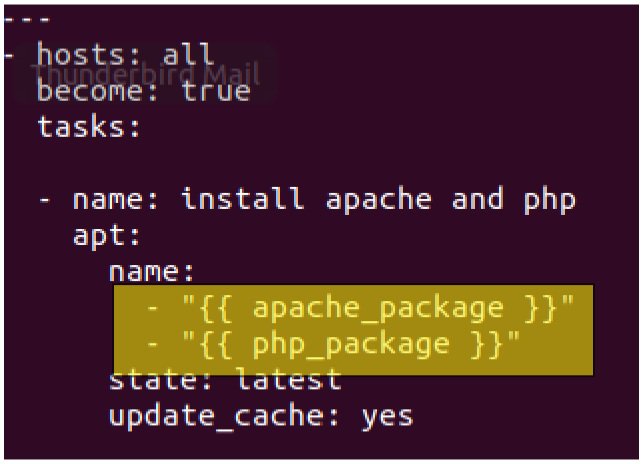
Make sure to save the file and exit.

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

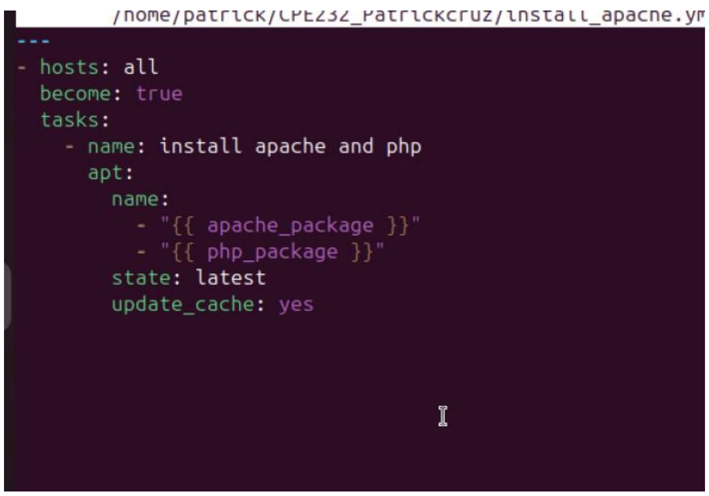


```
patrick@Workstation:~/CPE232_Patrickcruz$ ansible-playbook -i inventory --a
sk-become-pass install_apache.yml
BECOME password:

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

TASK [Gathering Facts] ***************************************************
*****
ok: [192.168.64.5]
ok: [192.168.64.12]

TASK [Install apache and php packages for Ubuntu] *************************
*****
skipping: [192.168.64.12]
ok: [192.168.64.5]

TASK [Install apache and php packages for CentOS] *************************
*****
skipping: [192.168.64.5]
ok: [192.168.64.12]

PLAY RECAP ***************************************************************
*****
192.168.64.12              : ok=2    changed=0    unreachable=0    failed=0
    skipped=1    rescued=0    ignored=0
192.168.64.5               : ok=2    changed=0    unreachable=0    failed=0
    skipped=1    rescued=0    ignored=0
```

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.

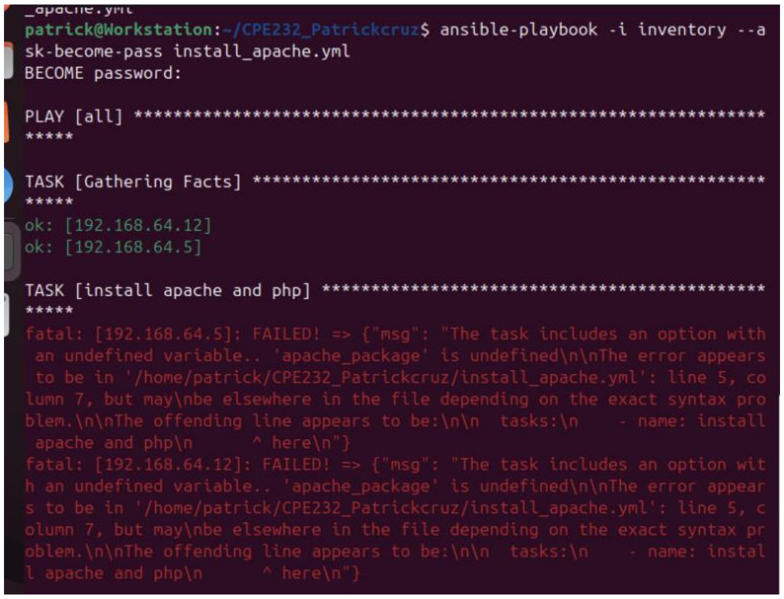


```
---
- hosts: all
  become: true
  tasks:

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

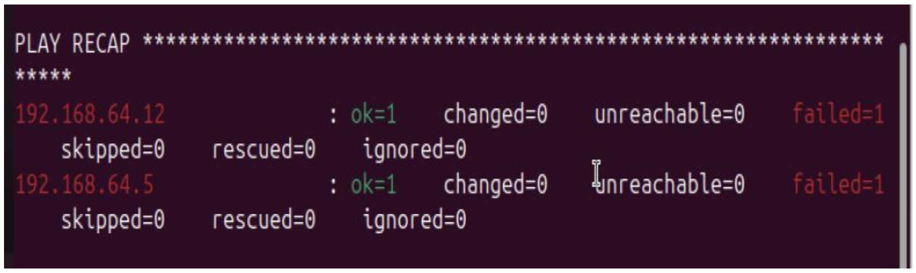


```
/home/patrick/CPE232_PatrickCruz/install_apache.ym
---
- hosts: all
  become: true
  tasks:
    - name: install apache and php
      apt:
        name:
          - "{{ apache_package }}"
          - "{{ php_package }}"
        state: latest
        update_cache: yes
```

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



```
_apache.yml
patrick@Workstation:~/CPE232_Patrickcruz$ ansible-playbook -i inventory --a
sk-become-pass install_apache.yml
BECOME password:

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

TASK [Gathering Facts] *****************************************************
*****
ok: [192.168.64.12]
ok: [192.168.64.5]

TASK [install apache and php] **********************************************
*****
fatal: [192.168.64.5]: FAILED! => {"msg": "The task includes an option with
 an undefined variable.. 'apache_package' is undefined\n\nThe error appears
 to be in '/home/patrick/CPE232_Patrickcruz/install_apache.yml': line 5, co
lumn 7, but may\nbe elsewhere in the file depending on the exact syntax pro
blem.\n\nThe offending line appears to be:\n\n  tasks:\n    - name: install
 apache and php\n      ^ here\n"}
fatal: [192.168.64.12]: FAILED! => {"msg": "The task includes an option wit
h an undefined variable.. 'apache_package' is undefined\n\nThe error appear
s to be in '/home/patrick/CPE232_Patrickcruz/install_apache.yml': line 5, c
olumn 7, but may\nbe elsewhere in the file depending on the exact syntax pr
oblem.\n\nThe offending line appears to be:\n\n  tasks:\n    - name: instal
l apache and php\n      ^ here\n"}
```



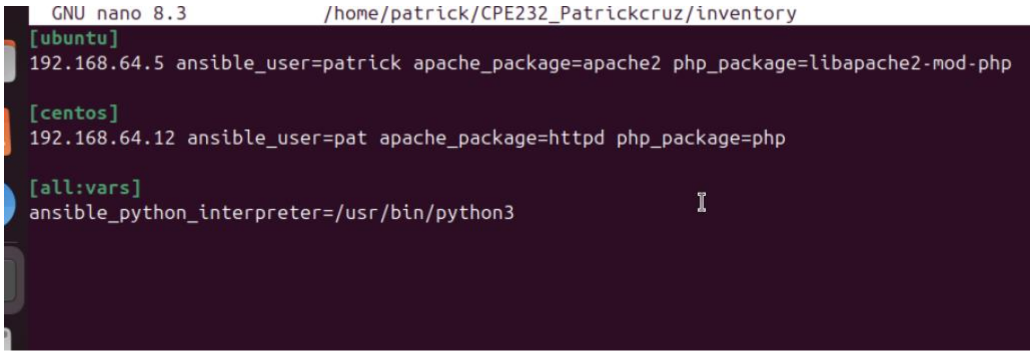
```
PLAY RECAP *****************************************************************
*****
192.168.64.12              : ok=1    changed=0    unreachable=0    failed=1
    skipped=0    rescued=0    ignored=0
192.168.64.5               : ok=1    changed=0    unreachable=0    failed=1
    skipped=0    rescued=0    ignored=0
```

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
```

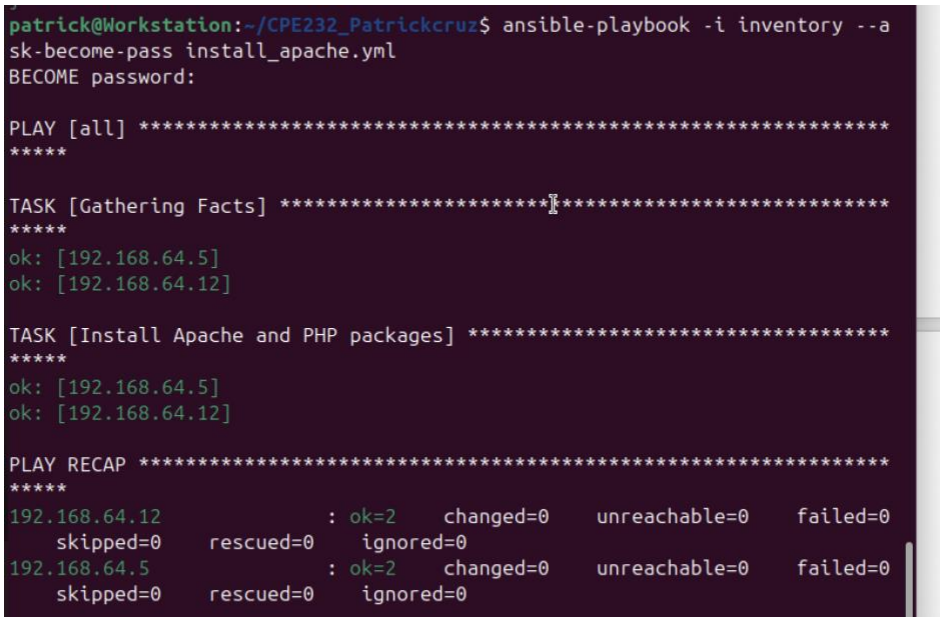
Make sure to save the *inventory* file and exit.

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]()



```
GNU nano 8.3              /home/patrick/CPE232_Patrickcruz/inventory
[ubuntu]
192.168.64.5 ansible_user=patrick apache_package=apache2 php_package=libapache2-mod-php

[centos]
192.168.64.12 ansible_user=pat apache_package=httpd php_package=php

[all:vars]
ansible_python_interpreter=/usr/bin/python3
```

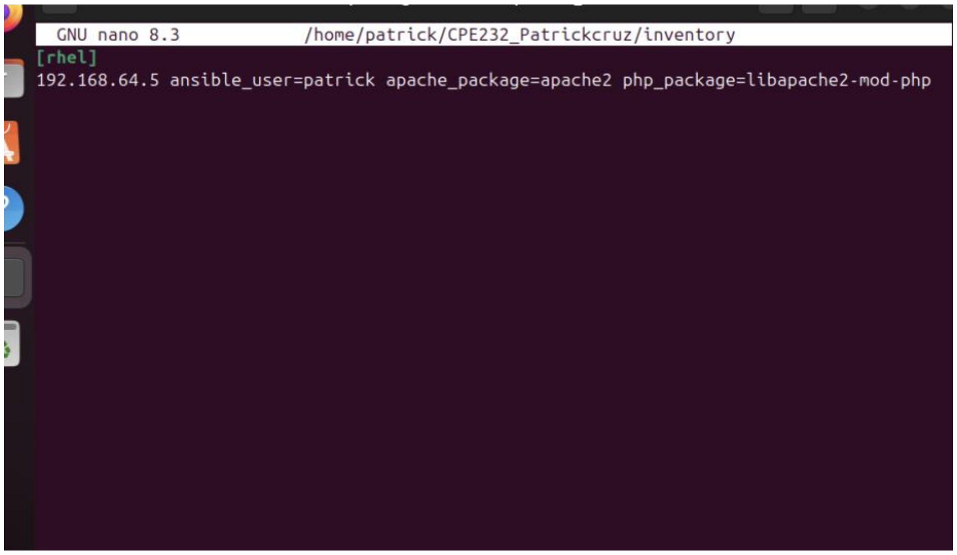
Run *ansible-playbook --ask-become-pass install_apache.yml* and describe the result.



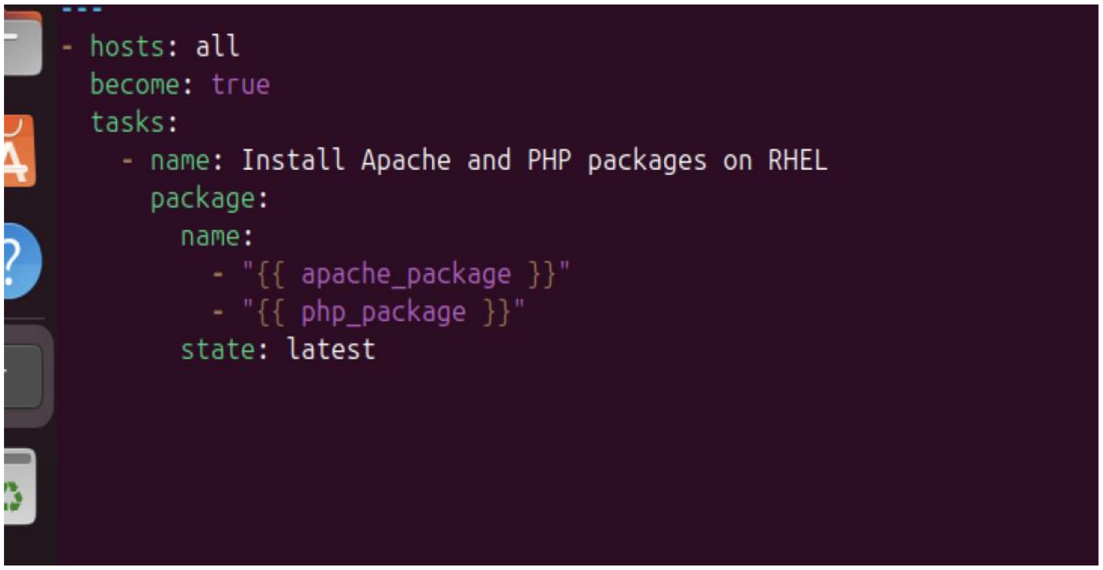
```
patrick@Workstation:~/CPE232_Patrickcruz$ ansible-playbook -i inventory --a
sk-become-pass install_apache.yml
BECOME password:

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

TASK [Gathering Facts] ****************************************************
*****
ok: [192.168.64.5]
ok: [192.168.64.12]

TASK [Install Apache and PHP packages] ************************************
*****
ok: [192.168.64.5]
ok: [192.168.64.12]

PLAY RECAP ****************************************************************
*****
192.168.64.12              : ok=2    changed=0    unreachable=0    failed=0
    skipped=0    rescued=0    ignored=0
192.168.64.5               : ok=2    changed=0    unreachable=0    failed=0
    skipped=0    rescued=0    ignored=0
```

Supplementary Activity:

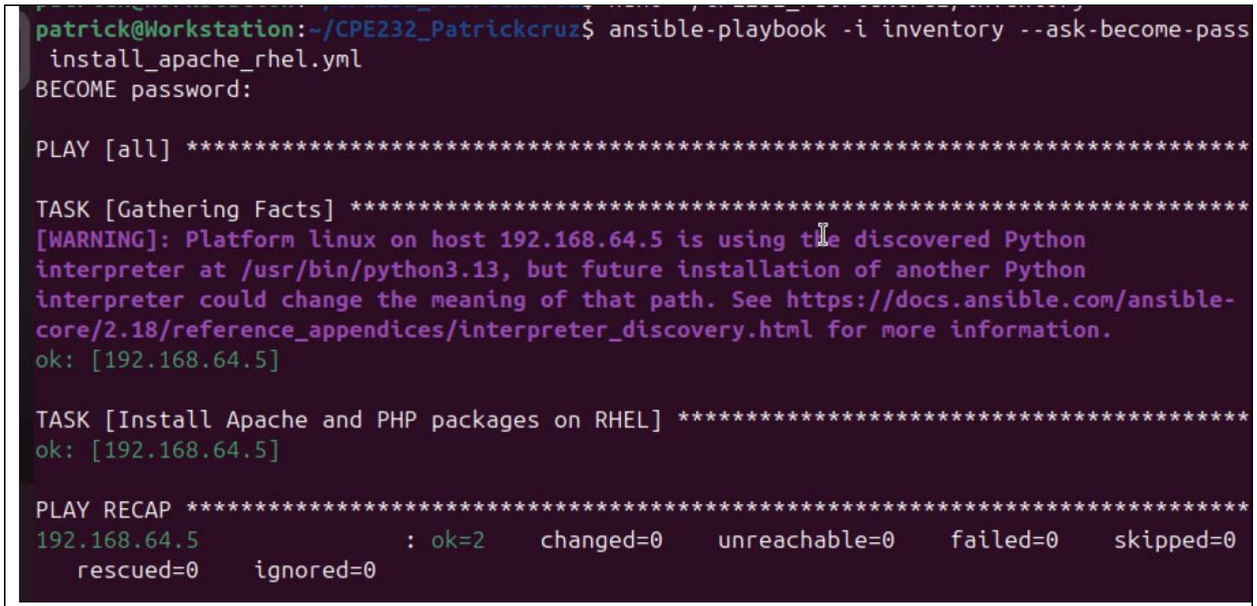
1. Create a playbook that could do the previous tasks in Red Hat OS.



```
GNU nano 8.3              /home/patrick/CPE232_Patrickcruz/inventory
[rhel]
192.168.64.5 ansible_user=patrick apache_package=apache2 php_package=libapache2-mod-php
```



```
---
- hosts: all
  become: true
  tasks:
    - name: Install Apache and PHP packages on RHEL
      package:
        name:
          - "{{ apache_package }}"
          - "{{ php_package }}"
        state: latest
```



```
patrick@Workstation:~/CPE232_Patrickcruz$ ansible-playbook -i inventory --ask-become-pass
 install_apache_rhel.yml
BECOME password:

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

TASK [Gathering Facts] *********************************************************
[WARNING]: Platform linux on host 192.168.64.5 is using the discovered Python
interpreter at /usr/bin/python3.13, but future installation of another Python
interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-
core/2.18/reference_appendices/interpreter_discovery.html for more information.
ok: [192.168.64.5]

TASK [Install Apache and PHP packages on RHEL] *********************************
ok: [192.168.64.5]

PLAY RECAP *********************************************************************
192.168.64.5               : ok=2    changed=0    unreachable=0    failed=0    skipped=0
   rescued=0    ignored=0
```

Reflections:

Answer the following:

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

   - Refactoring is important because it makes playbooks more efficient, easier to read, and simpler to maintain. It reduces code duplication, which minimizes errors. Using variables and generic modules allows one playbook to manage many different operating systems, making the code reusable and adaptable.

2. When do we use the “when” command in playbook?

   - The when command is used to run a task conditionally. It is primarily used to execute tasks on specific operating systems, for example, to only use the apt module when the target system is Ubuntu when ansible distribution ==Ubuntu. It can also be used to run tasks based on custom variables or the outcome of previous tasks.