Name:Zamora, Denzel	**Date Performed: 9 / 18 / 2023**
Course/Section:CPE232 - CPE31S6	**Date Submitted: 9 / 21 / 2023**
Instructor: Dr. Jonathan Taylar	**Semester and SY: 1st Sem S.Y. 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.

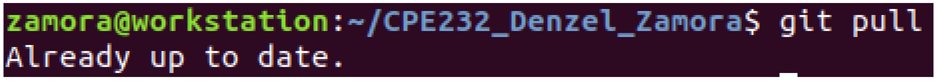
Requirement:

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 installation. 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

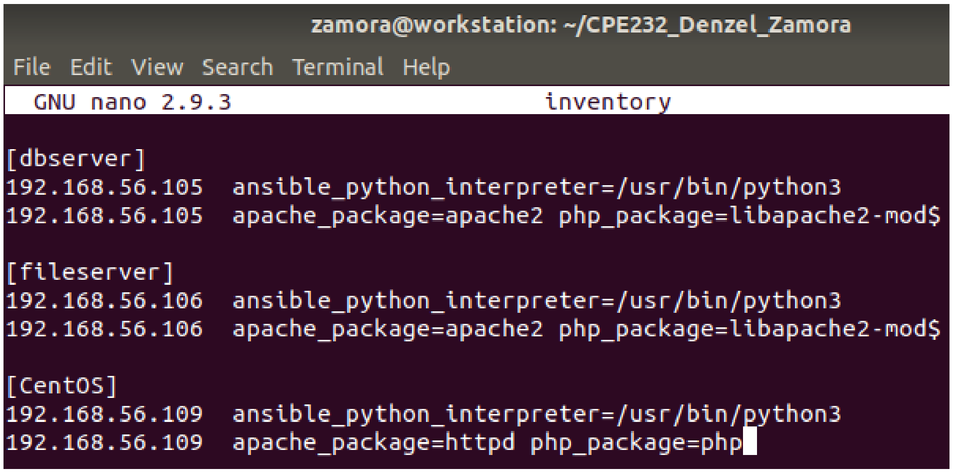
1. In the local machine, make sure you are in the local repository directory (*CPE232_yourname*). Issue the command git pull. When prompted, enter the correct passphrase or password. Describe what happens when you issue this command. Did something happen? Why?

No, because the local repository was already up to date



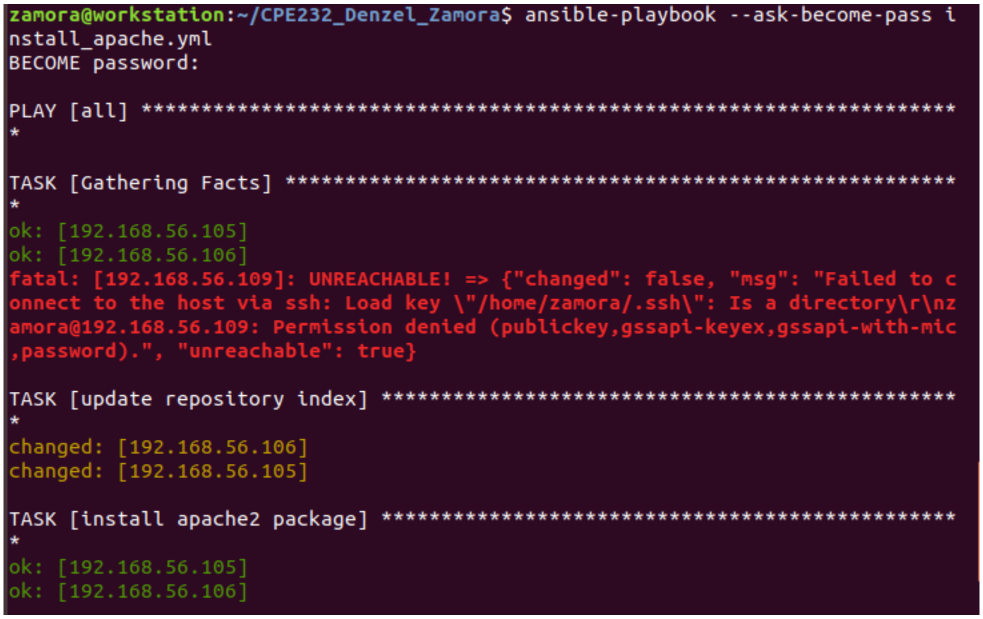
```
zamora@workstation:~/CPE232_Denzel_Zamora$ git pull
Already up to date.
```

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*.


zamora@workstation: ~/CPE232_Denzel_Zamora
File Edit View Search Terminal Help

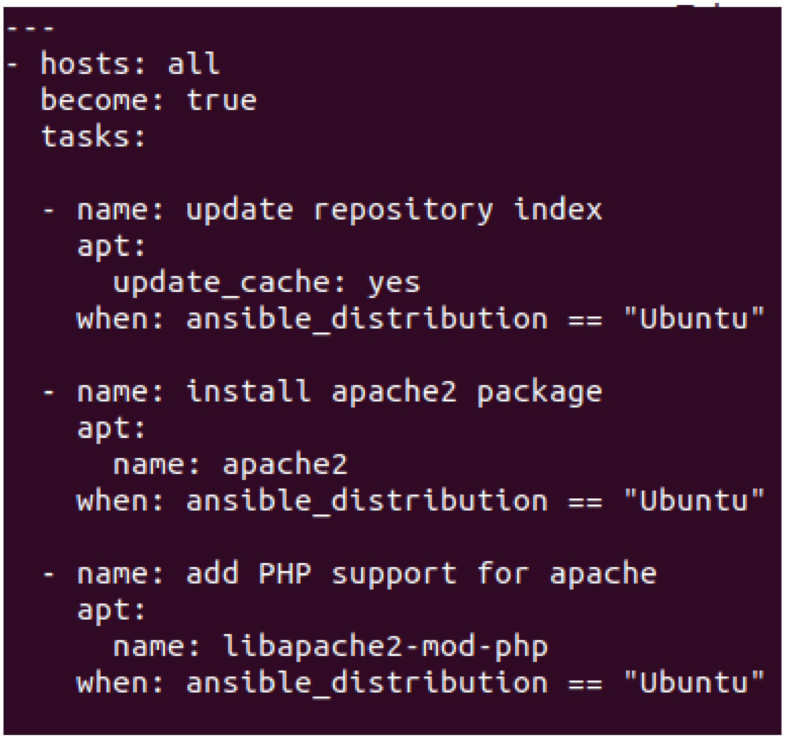
```
  GNU nano 2.9.3                          inventory

[dbserver]
192.168.56.105  ansible_python_interpreter=/usr/bin/python3
192.168.56.105  apache_package=apache2 php_package=libapache2-mod$

[fileserver]
192.168.56.106  ansible_python_interpreter=/usr/bin/python3
192.168.56.106  apache_package=apache2 php_package=libapache2-mod$

[CentOS]
192.168.56.109  ansible_python_interpreter=/usr/bin/python3
192.168.56.109  apache_package=httpd php_package=php
```

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."



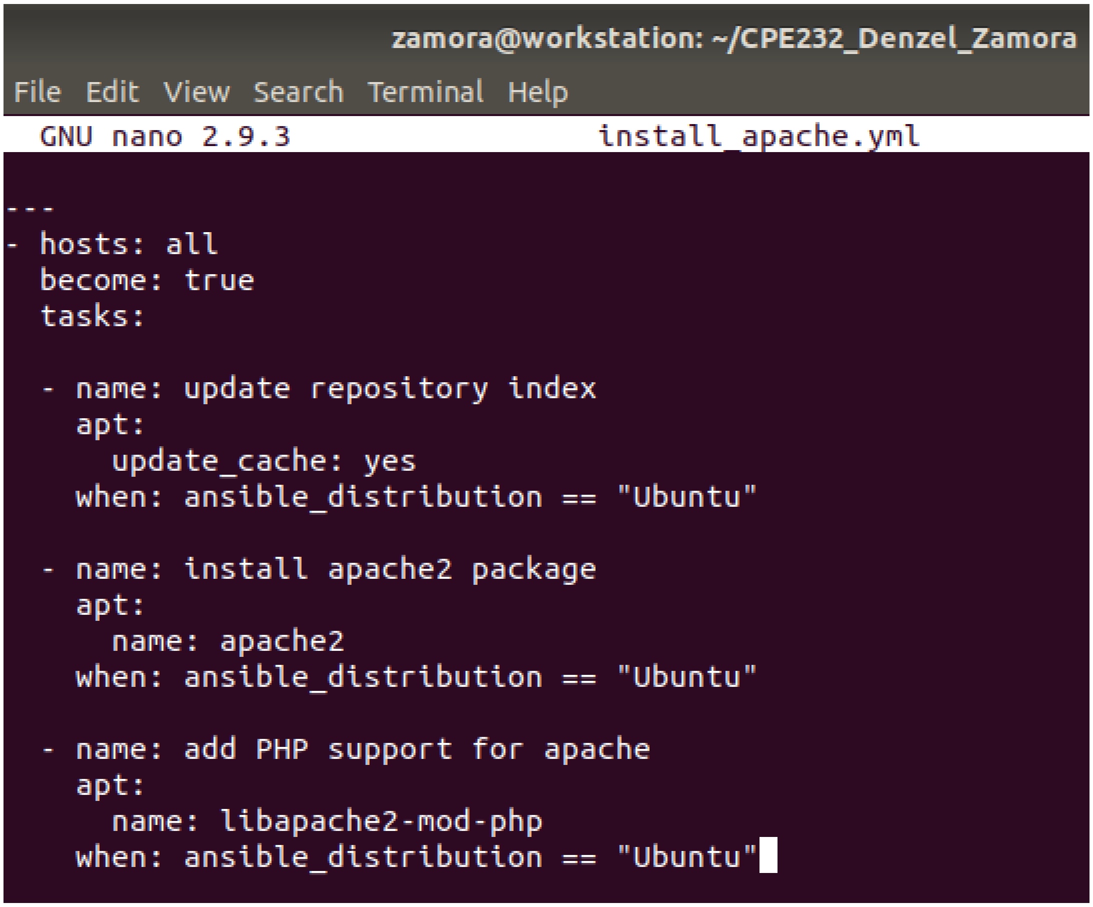
```
zamora@workstation:~/CPE232_Denzel_Zamora$ ansible-playbook --ask-become-pass i
nstall_apache.yml
BECOME password:

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

TASK [Gathering Facts] *********************************************************
*
ok: [192.168.56.105]
ok: [192.168.56.106]
fatal: [192.168.56.109]: UNREACHABLE! => {"changed": false, "msg": "Failed to c
onnect to the host via ssh: Load key \"/home/zamora/.ssh\": Is a directory\r\nz
amora@192.168.56.109: Permission denied (publickey,gssapi-keyex,gssapi-with-mic
,password).", "unreachable": true}

TASK [update repository index] *************************************************
*
changed: [192.168.56.106]
changed: [192.168.56.105]

TASK [install apache2 package] *************************************************
*
ok: [192.168.56.105]
ok: [192.168.56.106]
```

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.

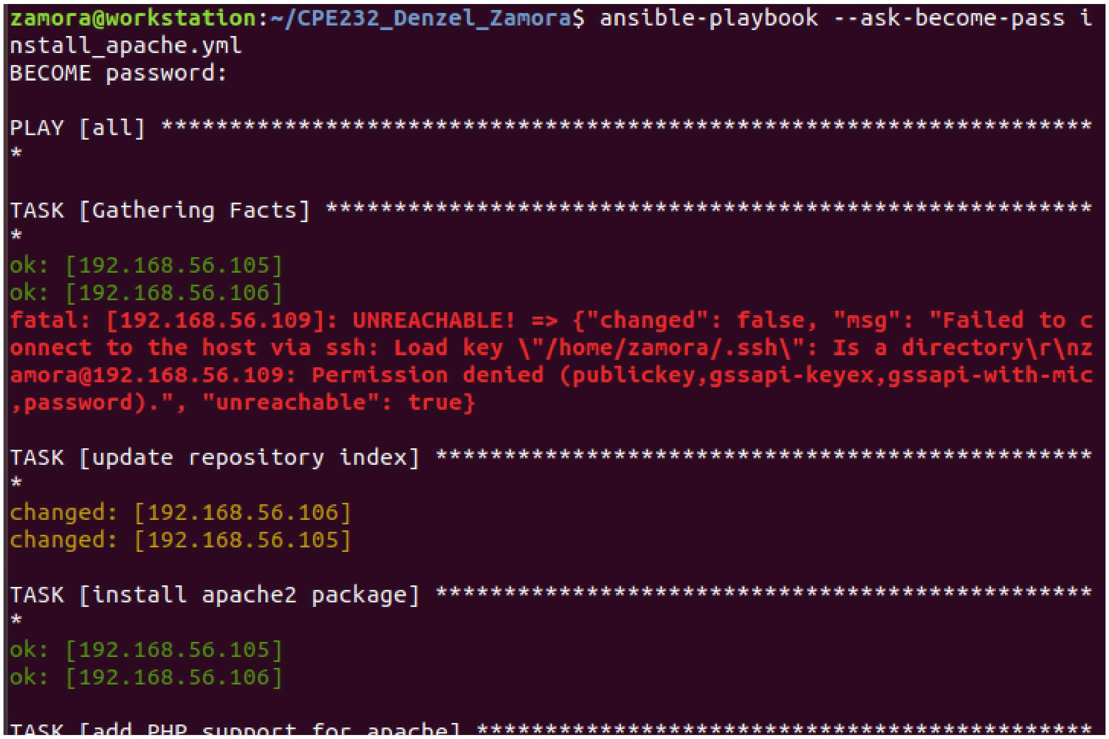


zamora@workstation: ~/CPE232_Denzel_Zamora
File Edit View Search Terminal Help
GNU nano 2.9.3
install_apache.yml

```
---
- 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.

<u>**The playbook ran and updated the server 1 and server 2 since the playbook mentioned that only Ubuntu servers will be updated.**</u>



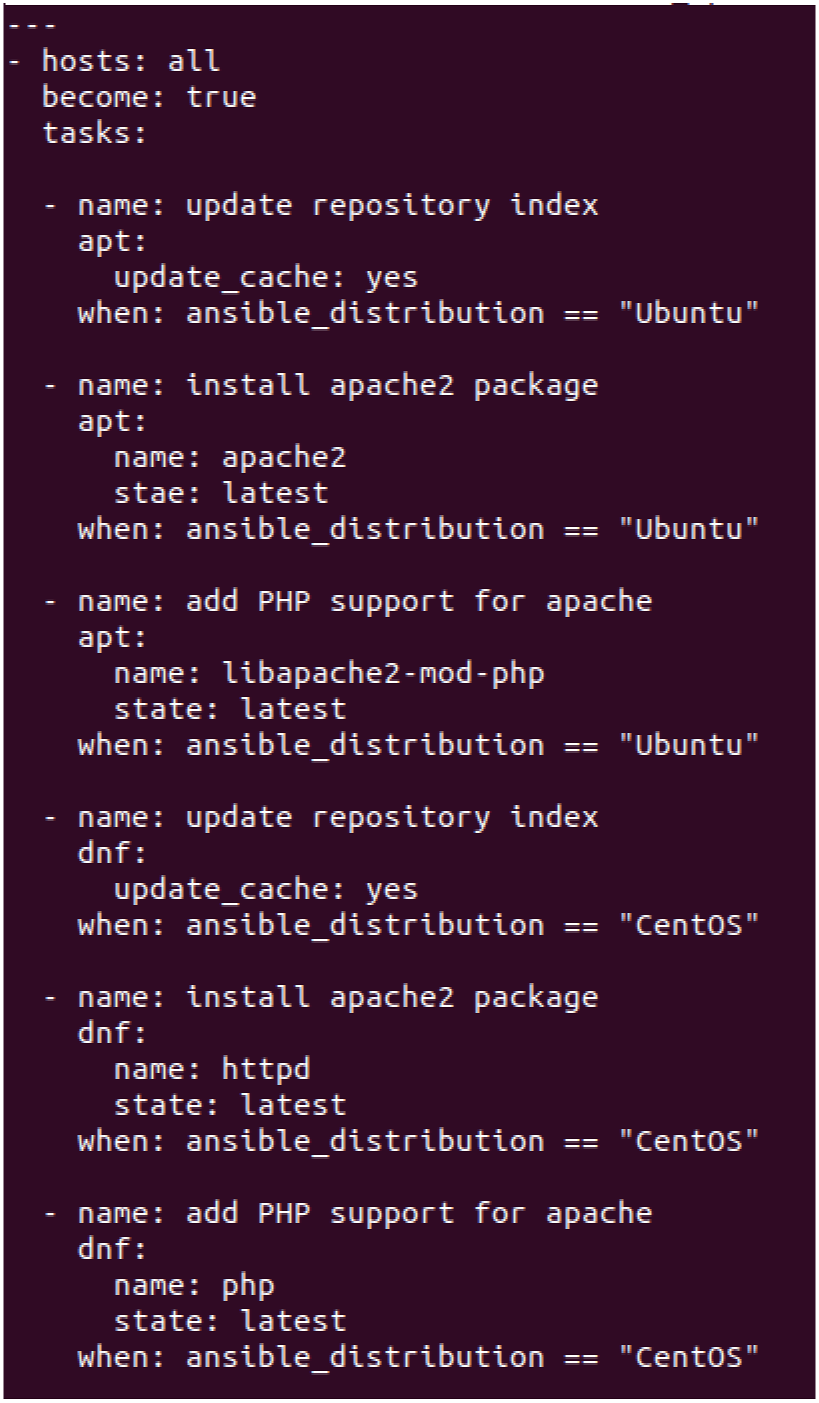
```
zamora@workstation:~/CPE232_Denzel_Zamora$ ansible-playbook --ask-become-pass i
nstall_apache.yml
BECOME password:

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

TASK [Gathering Facts] ********************************************************
*
ok: [192.168.56.105]
ok: [192.168.56.106]
fatal: [192.168.56.109]: UNREACHABLE! => {"changed": false, "msg": "Failed to c
onnect to the host via ssh: Load key \"/home/zamora/.ssh\": Is a directory\r\nz
amora@192.168.56.109: Permission denied (publickey,gssapi-keyex,gssapi-with-mic
,password).", "unreachable": true}

TASK [update repository index] ************************************************
*
changed: [192.168.56.106]
changed: [192.168.56.105]

TASK [install apache2 package] ************************************************
*
ok: [192.168.56.105]
ok: [192.168.56.106]

TASK [add PHP support for apache] *********************************************
```

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]
```

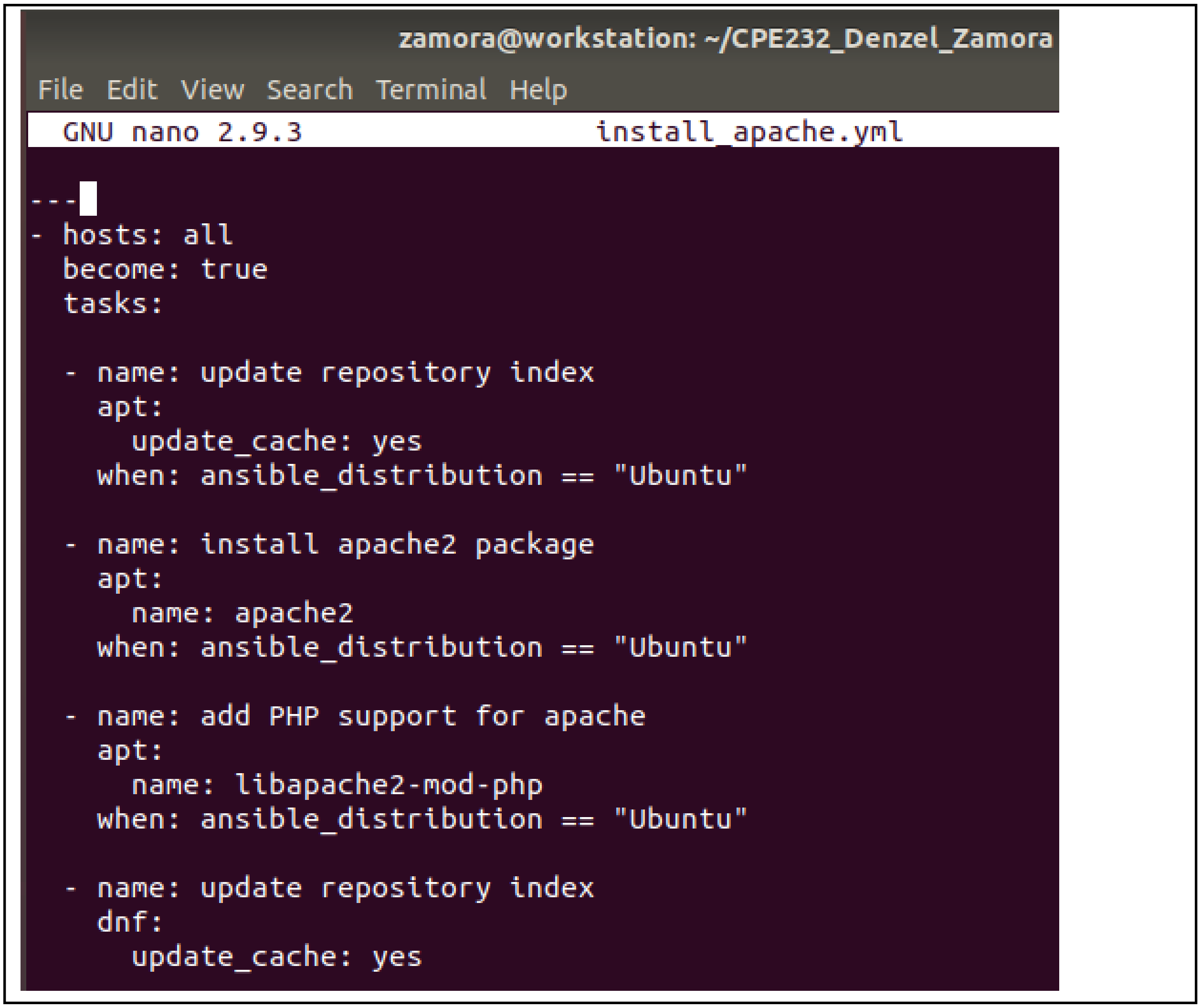
Note: This will work also if you try. Notice the changes are highlighted.

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



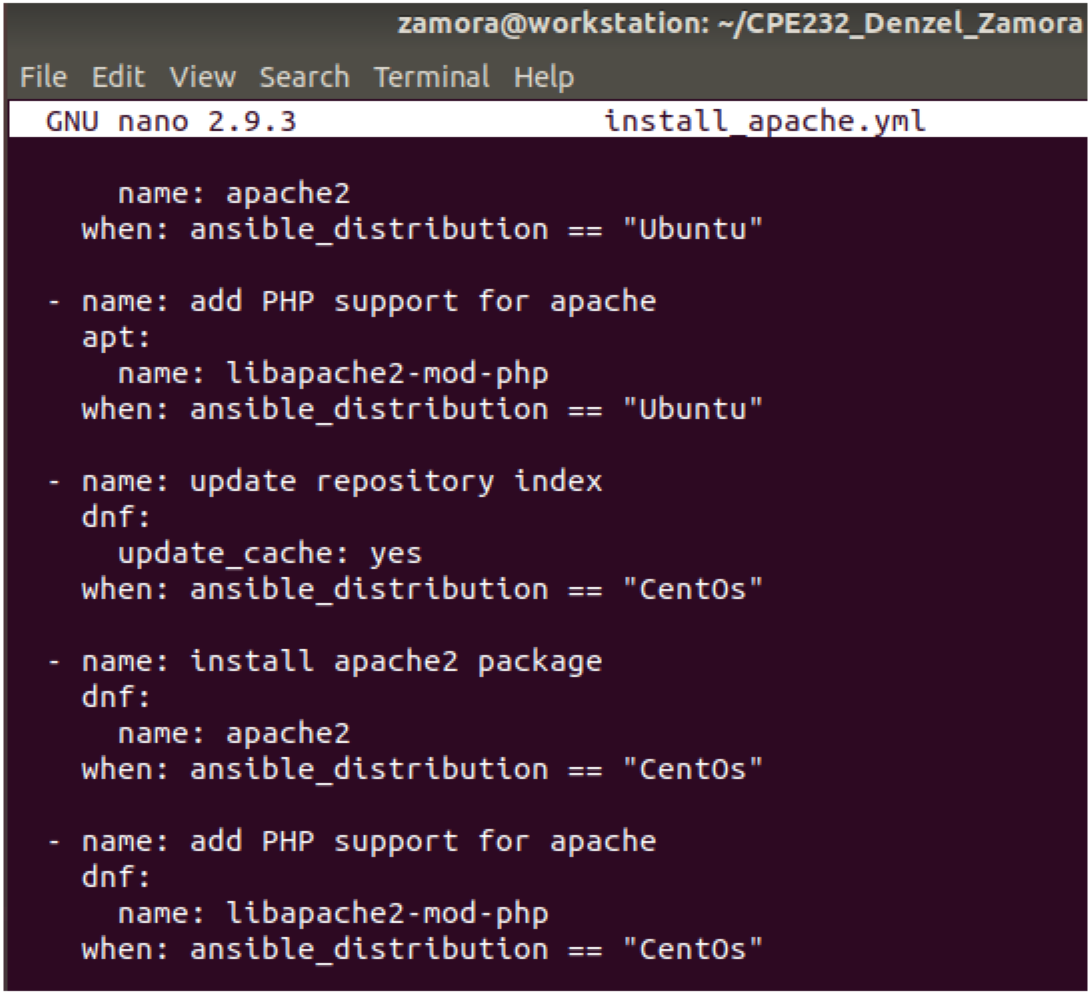
```
---
- hosts: all
  become: true
  tasks:

  - 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.



zamora@workstation: ~/CPE232_Denzel_Zamora
File Edit View Search Terminal Help
GNU nano 2.9.3 install_apache.yml

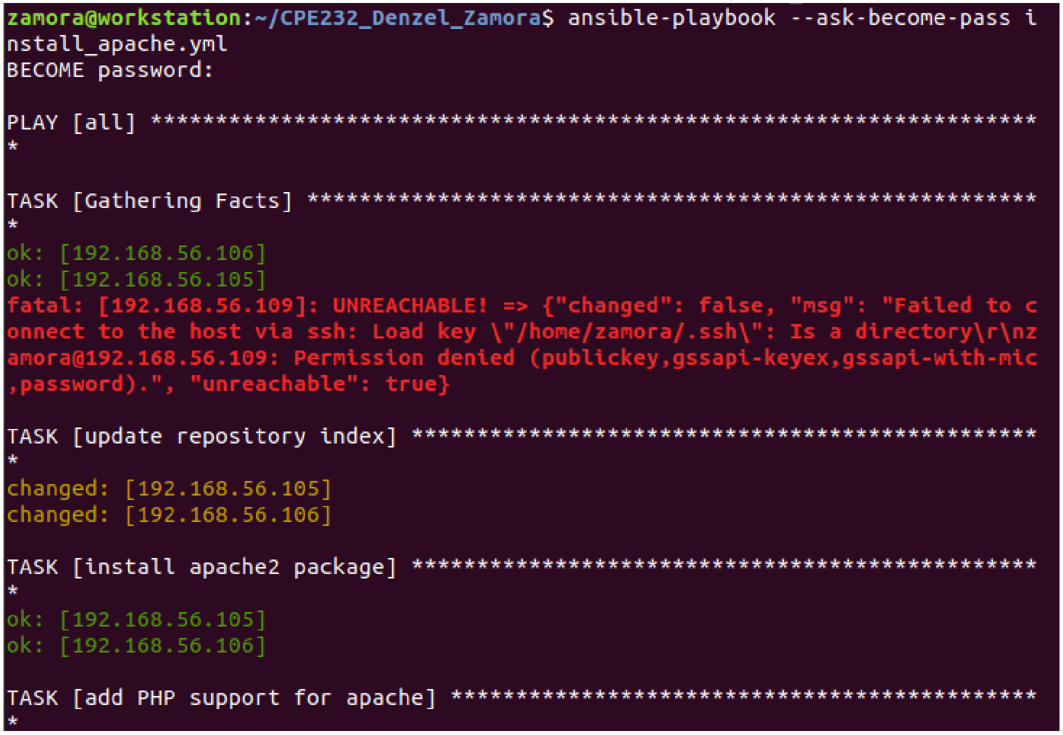
```
---
- hosts: all
  become: true
  tasks:

  - 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
    dnf:
      update_cache: yes
```


zamora@workstation: ~/CPE232_Denzel_Zamora
File Edit View Search Terminal Help

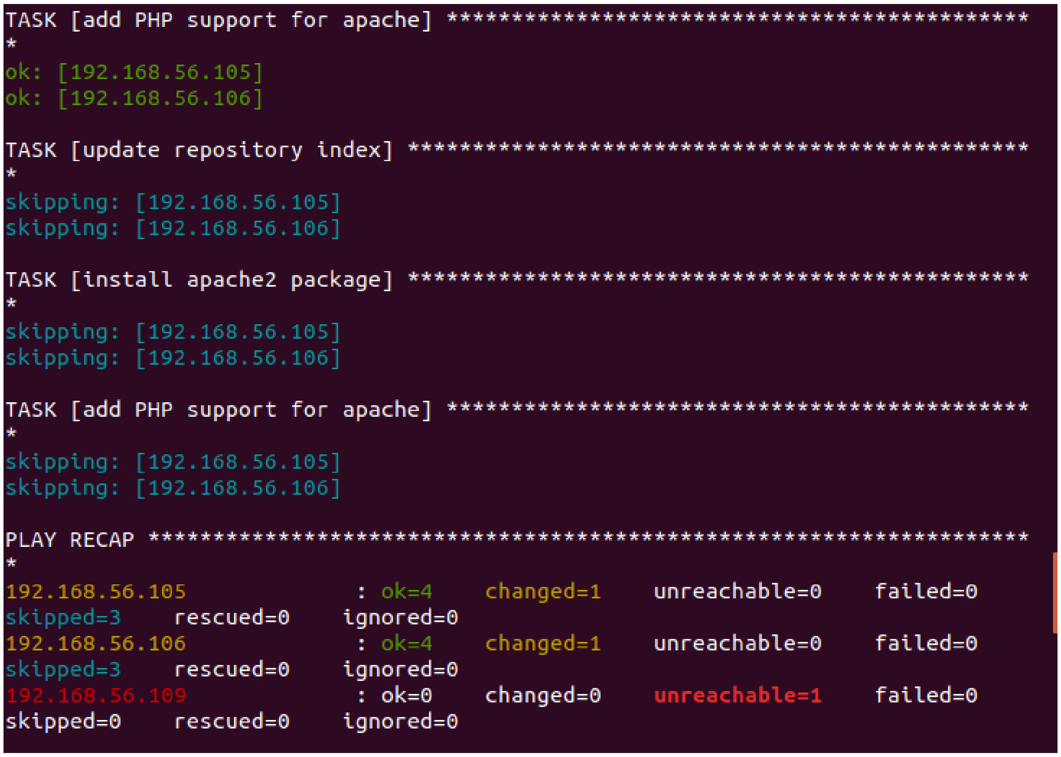
```
  GNU nano 2.9.3                     install_apache.yml

      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
    dnf:
      update_cache: yes
    when: ansible_distribution == "CentOs"

  - name: install apache2 package
    dnf:
      name: apache2
    when: ansible_distribution == "CentOs"

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

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

The playbook ran and updated the server 1 and server 2 but not the CentOS as the httpd has not been activated and the firewall on the CentOS has not been configured to allow connection from the Workstation.



```
zamora@workstation:~/CPE232_Denzel_Zamora$ ansible-playbook --ask-become-pass i
nstall_apache.yml
BECOME password:

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

TASK [Gathering Facts] *********************************************************
*
ok: [192.168.56.106]
ok: [192.168.56.105]
fatal: [192.168.56.109]: UNREACHABLE! => {"changed": false, "msg": "Failed to c
onnect to the host via ssh: Load key \"/home/zamora/.ssh\": Is a directory\r\nz
amora@192.168.56.109: Permission denied (publickey,gssapi-keyex,gssapi-with-mic
,password).", "unreachable": true}

TASK [update repository index] *************************************************
*
changed: [192.168.56.105]
changed: [192.168.56.106]

TASK [install apache2 package] *************************************************
*
ok: [192.168.56.105]
ok: [192.168.56.106]

TASK [add PHP support for apache] **********************************************
*
```



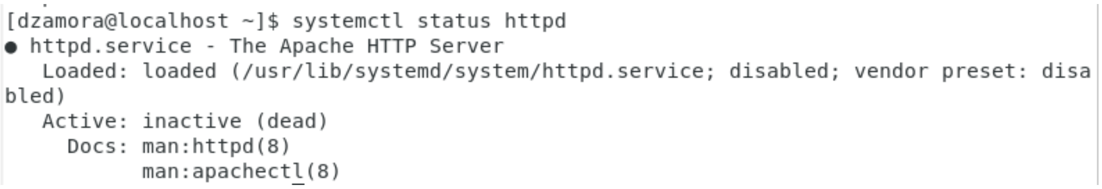
```
TASK [add PHP support for apache] **********************************************
*
ok: [192.168.56.105]
ok: [192.168.56.106]

TASK [update repository index] *************************************************
*
skipping: [192.168.56.105]
skipping: [192.168.56.106]

TASK [install apache2 package] *************************************************
*
skipping: [192.168.56.105]
skipping: [192.168.56.106]

TASK [add PHP support for apache] **********************************************
*
skipping: [192.168.56.105]
skipping: [192.168.56.106]

PLAY RECAP *********************************************************************
*
192.168.56.105             : ok=4    changed=1    unreachable=0    failed=0
skipped=3    rescued=0    ignored=0
192.168.56.106             : ok=4    changed=1    unreachable=0    failed=0
skipped=3    rescued=0    ignored=0
192.168.56.109             : ok=0    changed=0    unreachable=1    failed=0
skipped=0    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 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.



```
[dzamora@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)
```

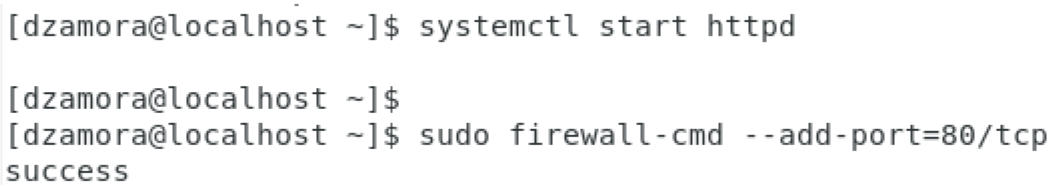
5.2 Issue the following command to start the service:

sudo systemctl start httpd

(When prompted, enter the sudo password)

sudo firewall-cmd --add-port=80/tcp

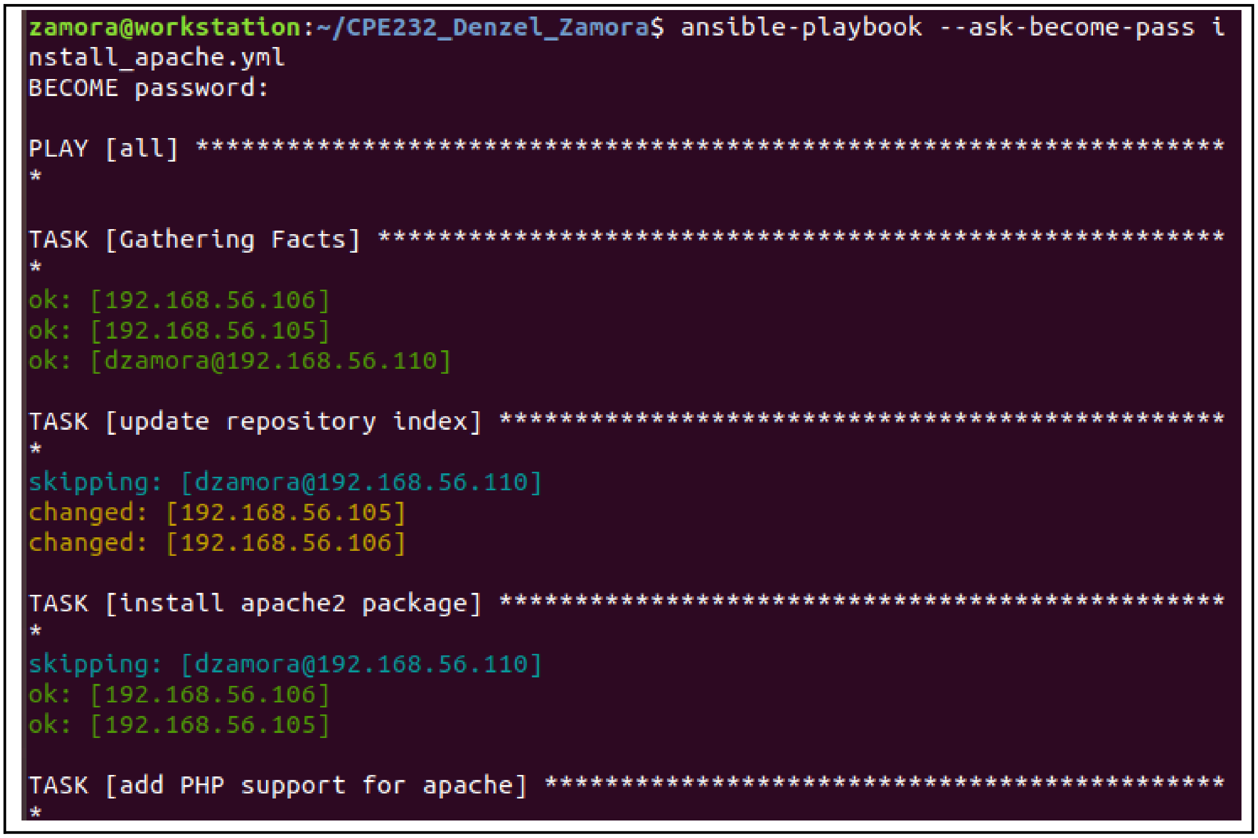
(The result should be a success)



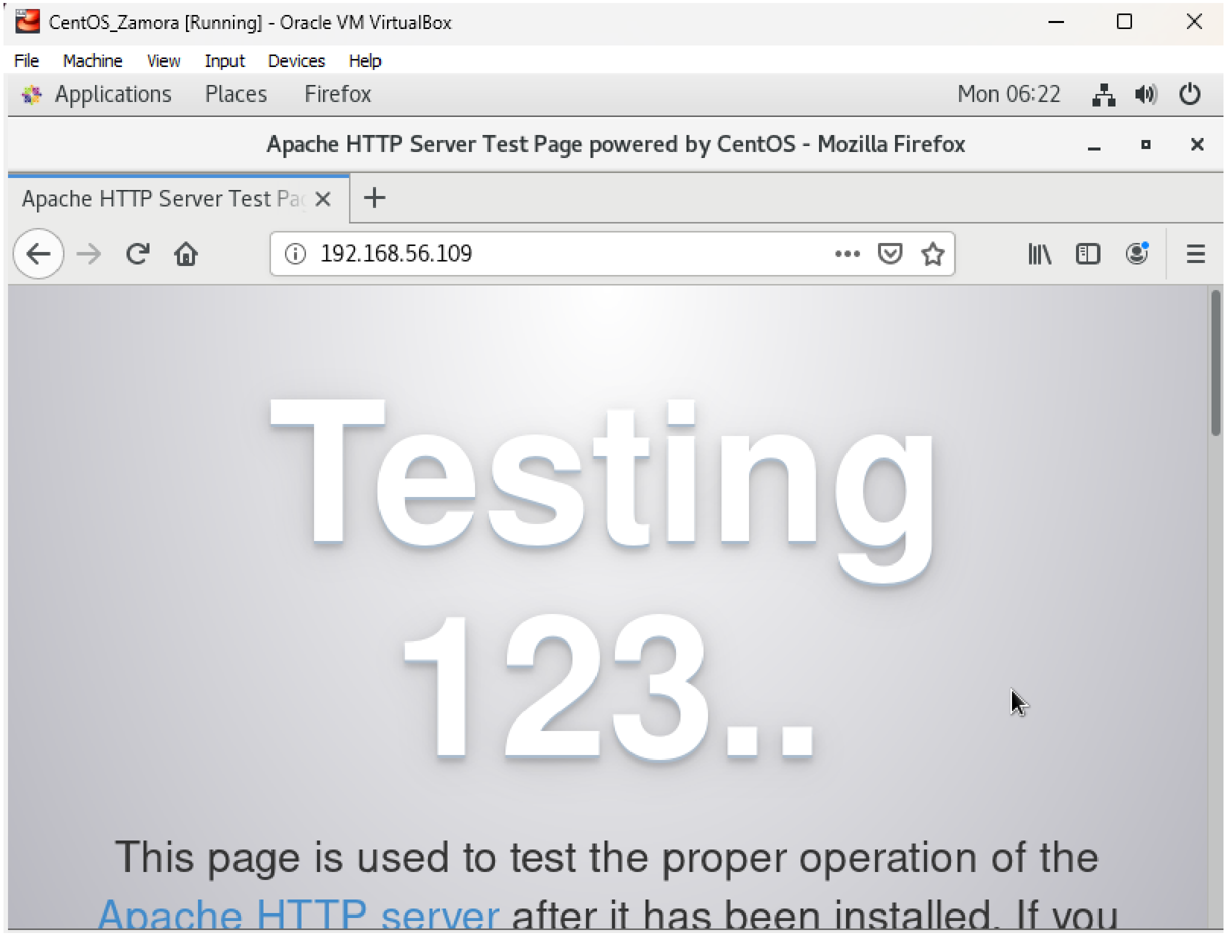
```
[dzamora@localhost ~]$ systemctl start httpd

[dzamora@localhost ~]$
[dzamora@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)

Yes, the connection has been successfully made.



```
zamora@workstation:~/CPE232_Denzel_Zamora$ ansible-playbook --ask-become-pass i
nstall_apache.yml
BECOME password:

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

TASK [Gathering Facts] ********************************************************
*
ok: [192.168.56.106]
ok: [192.168.56.105]
ok: [dzamora@192.168.56.110]

TASK [update repository index] ************************************************
*
skipping: [dzamora@192.168.56.110]
changed: [192.168.56.105]
changed: [192.168.56.106]

TASK [install apache2 package] ************************************************
*
skipping: [dzamora@192.168.56.110]
ok: [192.168.56.106]
ok: [192.168.56.105]

TASK [add PHP support for apache] *********************************************
*
```


CentOS_Zamora [Running] - Oracle VM VirtualBox
File
Machine
View
Input
Devices
Help
Applications
Places
Firefox
Mon 06:22
Apache HTTP Server Test Page powered by CentOS - Mozilla Firefox
Apache HTTP Server Test Pa
192.168.56.109

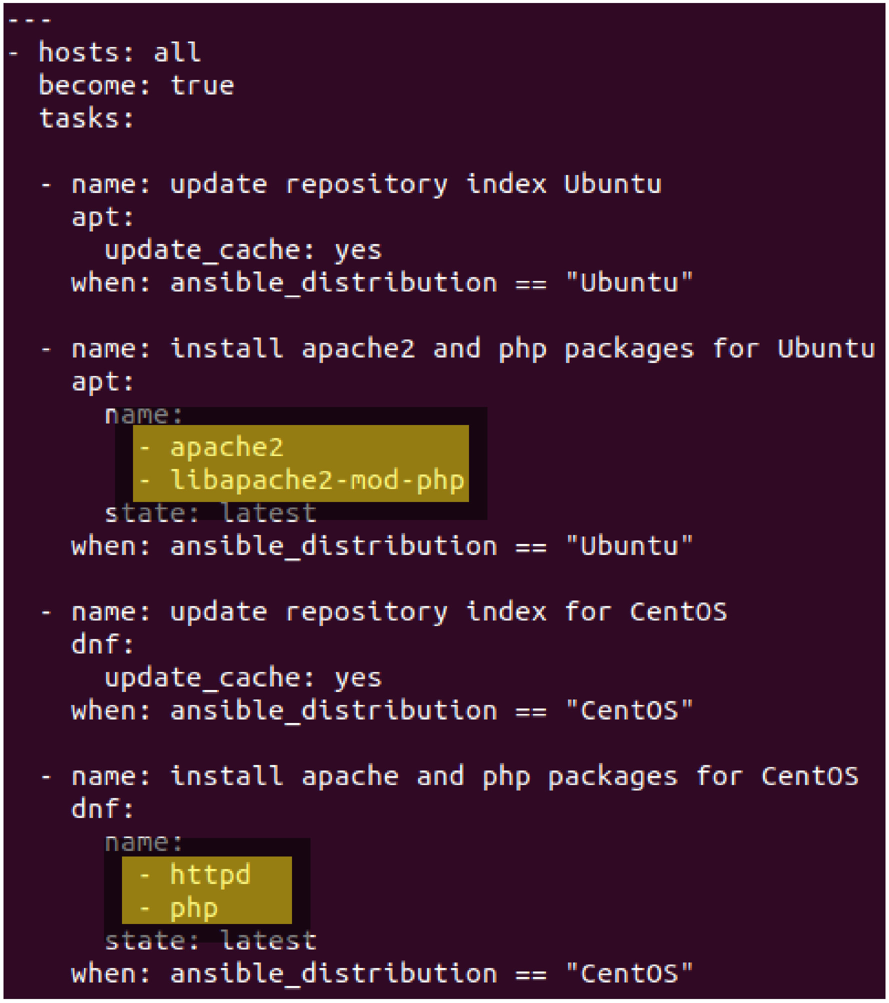
Testing 123..

This page is used to test the proper operation of the Apache HTTP server after it has been installed. If you

Task 2: Refactoring playbook

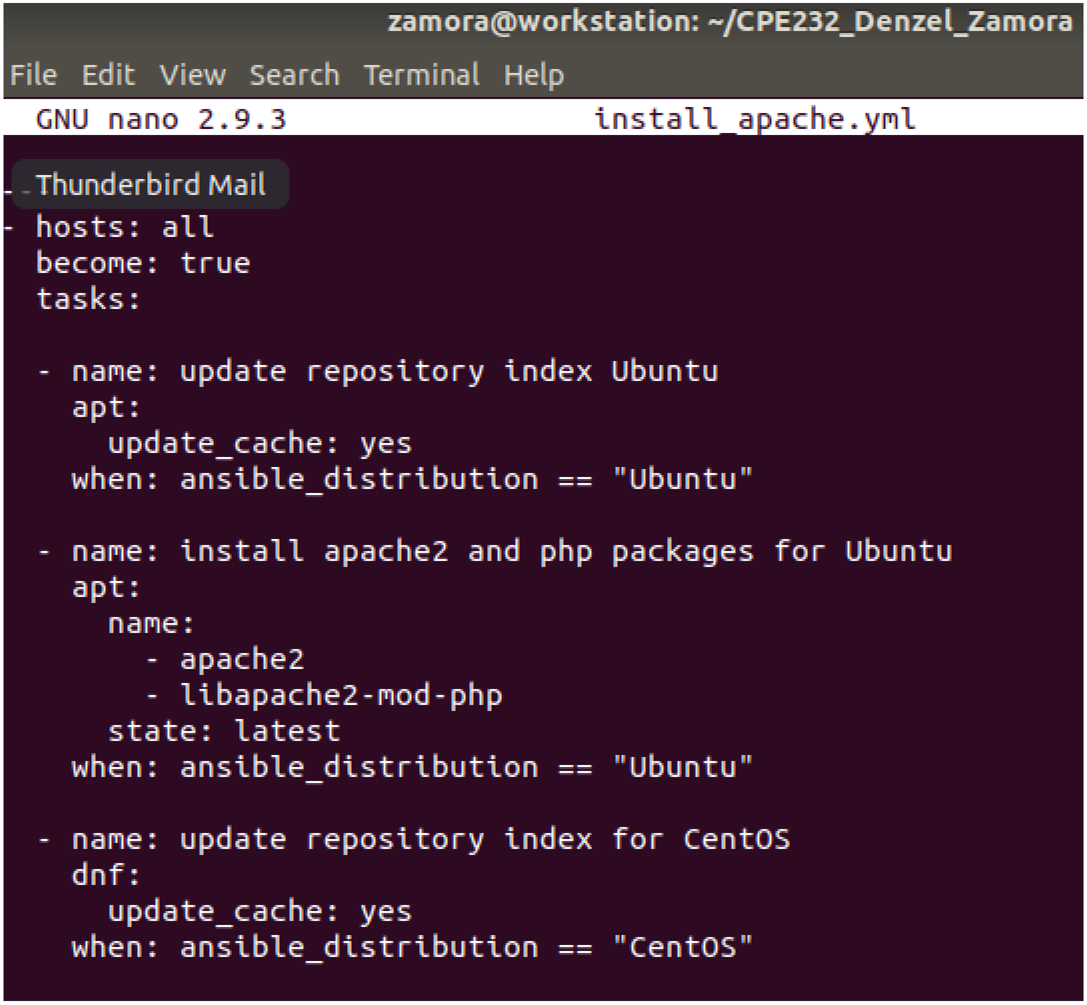
This time, we want to make sure that our playbook is efficient and that the codes are easier to read. This will also mexiakes run ansible more quickly if it has to execute fewer tasks to do the same thing.

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"
```

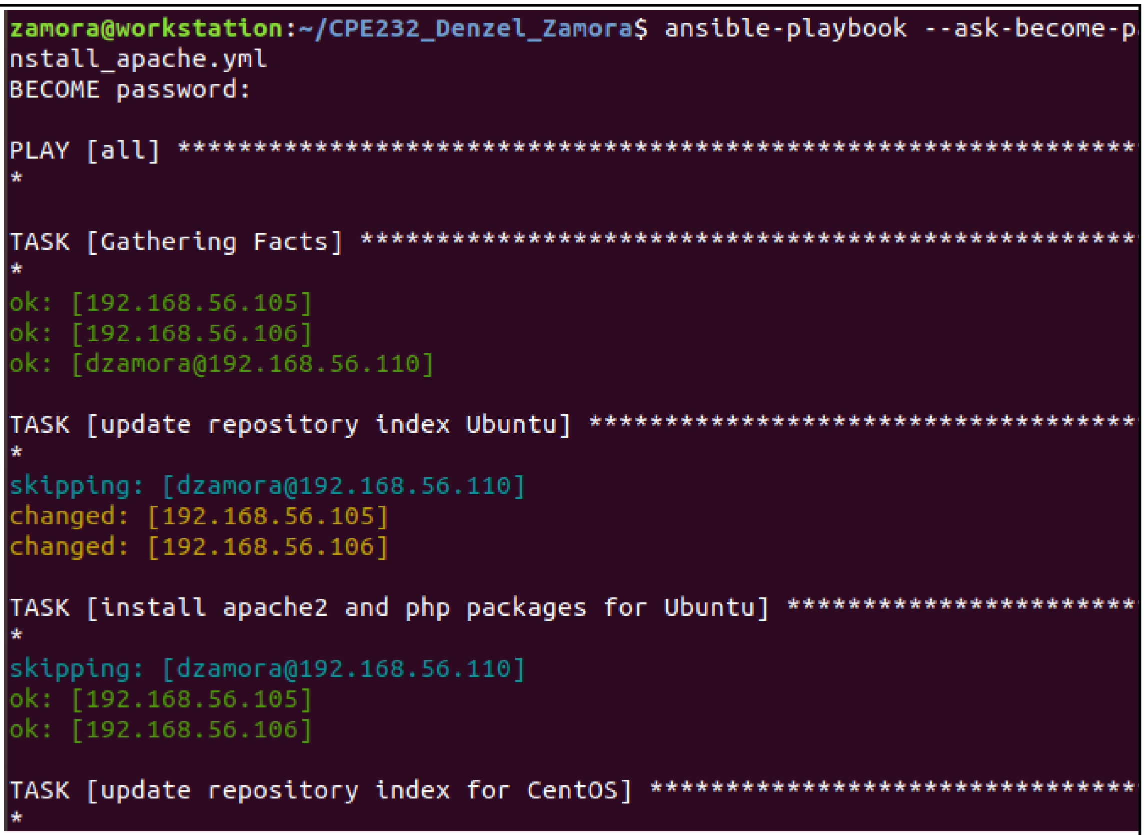
Make sure to save the file and exit.


zamora@workstation: ~/CPE232_Denzel_Zamora
File Edit View Search Terminal Help
Thunderbird Mail

```
  GNU nano 2.9.3                    install_apache.yml

- 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"
```

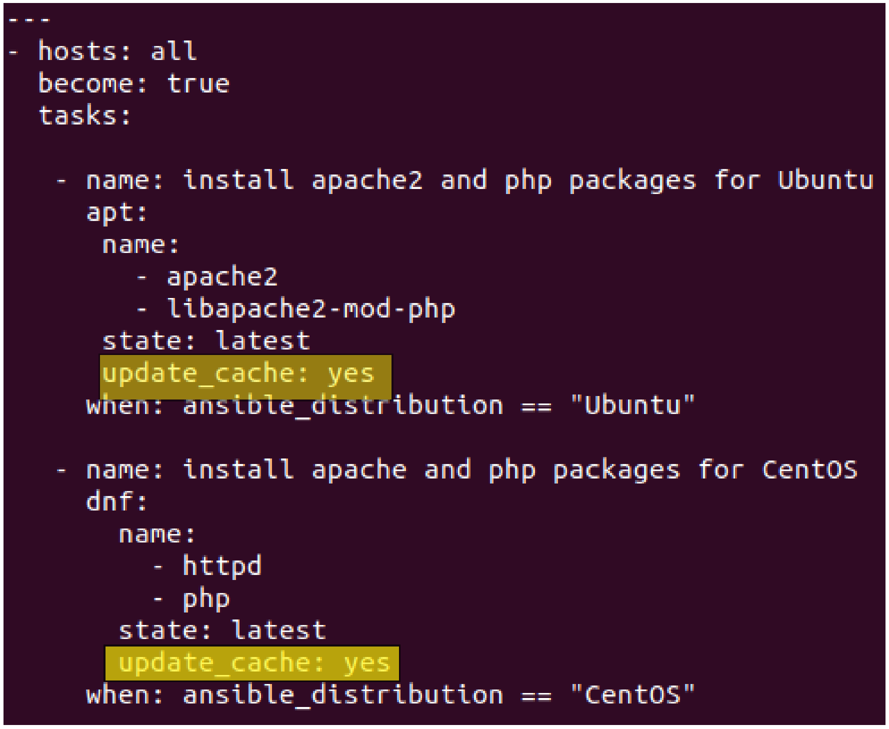
Run *ansible-playbook --ask-become-pass install_apache.yml* and describe the result.

We edited the playbook and combined the install apache2 and php packages into one syntax to lessen inputs.



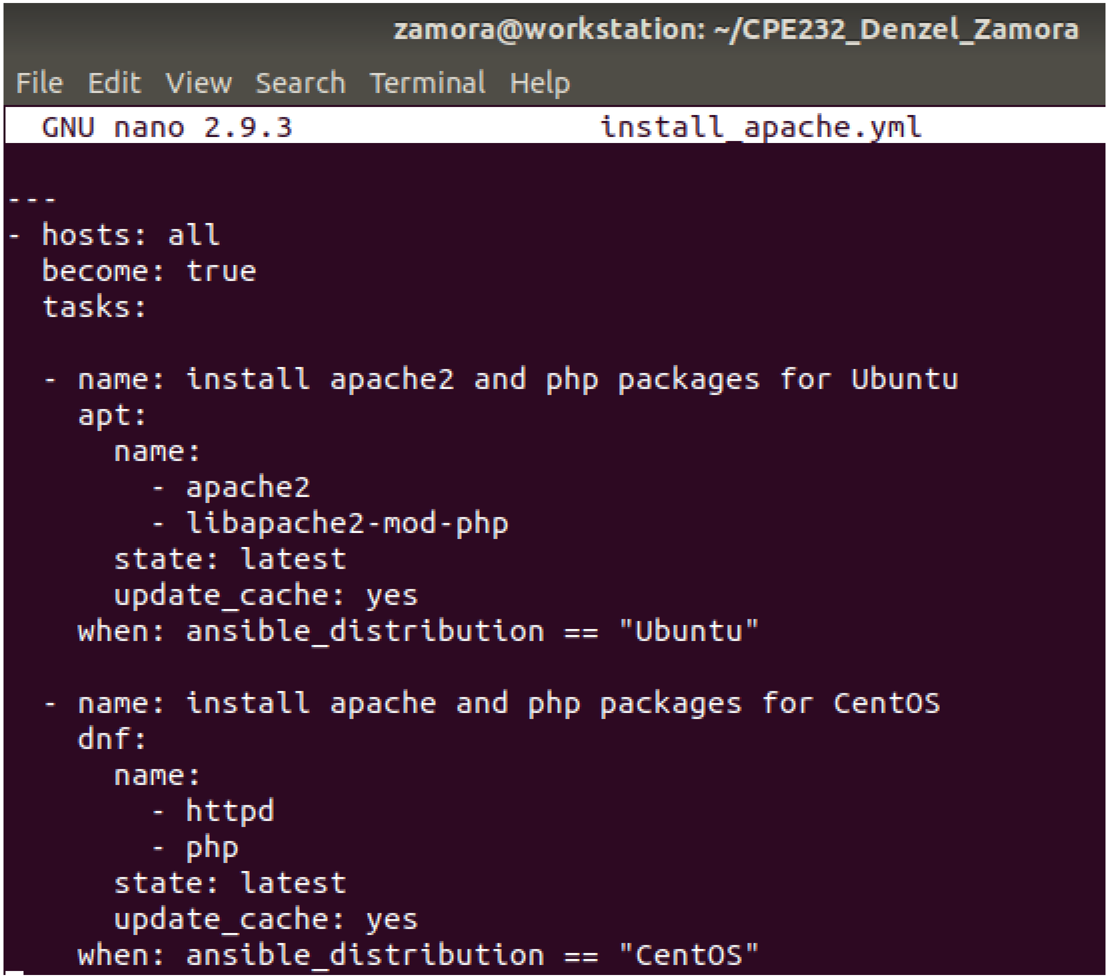
```
zamora@workstation:~/CPE232_Denzel_Zamora$ ansible-playbook --ask-become-p
nstall_apache.yml
BECOME password:

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

TASK [Gathering Facts] *****************************************************
*
ok: [192.168.56.105]
ok: [192.168.56.106]
ok: [dzamora@192.168.56.110]

TASK [update repository index Ubuntu] **************************************
*
skipping: [dzamora@192.168.56.110]
changed: [192.168.56.105]
changed: [192.168.56.106]

TASK [install apache2 and php packages for Ubuntu] *************************
*
skipping: [dzamora@192.168.56.110]
ok: [192.168.56.105]
ok: [192.168.56.106]

TASK [update repository index for 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 consolidate 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"
```

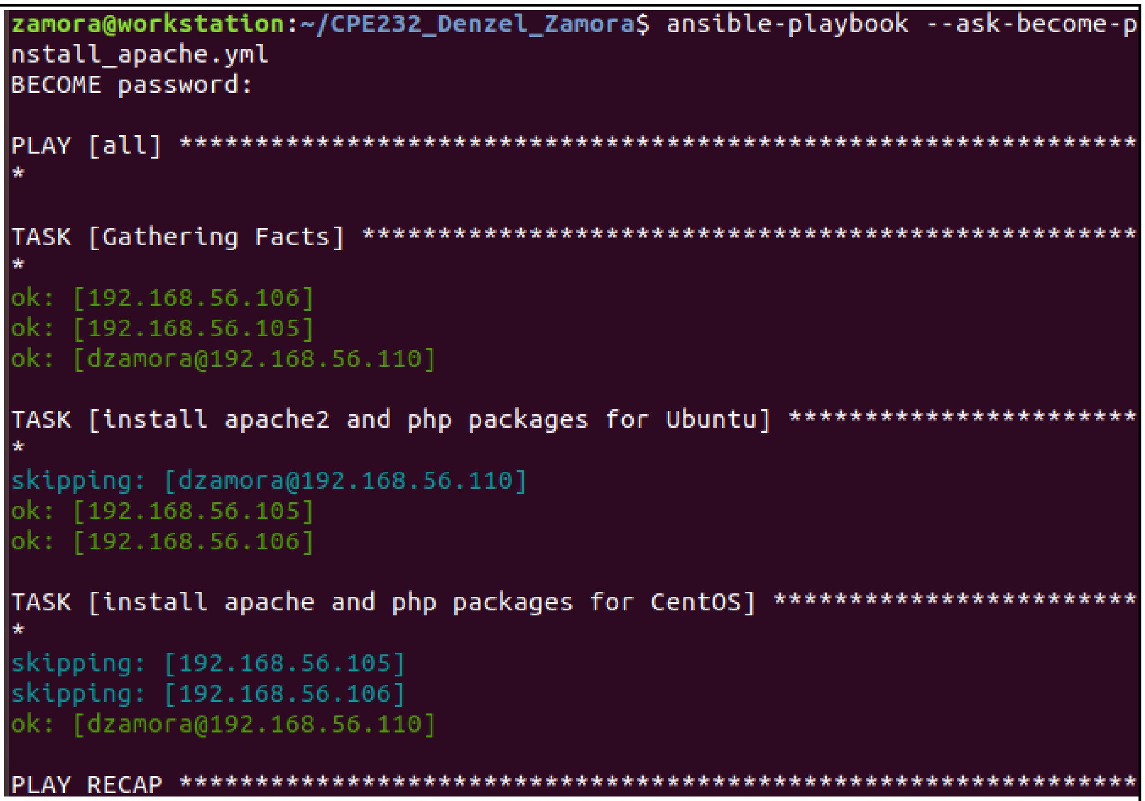
Make sure to save the file and exit.


zamora@workstation: ~/CPE232_Denzel_Zamora
File Edit View Search Terminal Help

```
  GNU nano 2.9.3                    install_apache.yml

---
- 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"
```

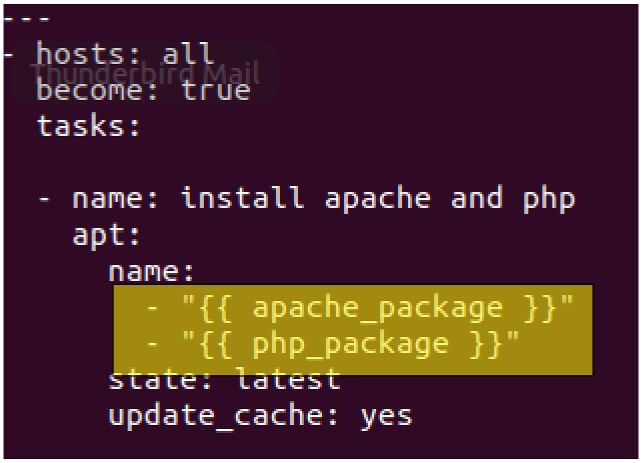
Run *ansible-playbook --ask-become-pass install_apache.yml* and describe the result.

We further lessen the commands typed in the playbook, combined the install and updates in the servers, and also added the 'state: latest' command to ensure that the installed packages are the latest version.

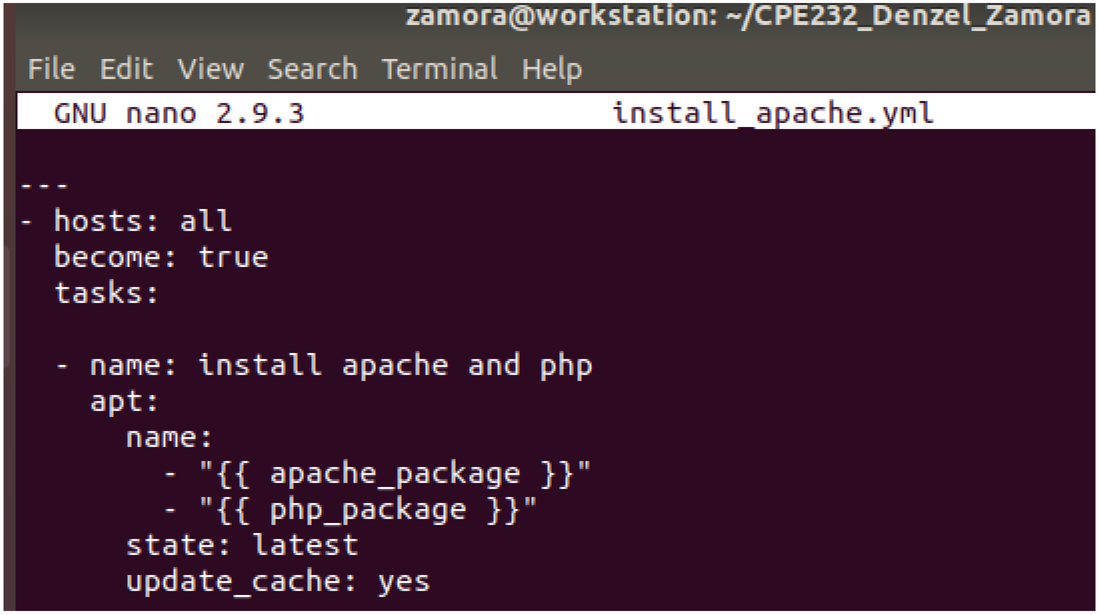


```
zamora@workstation:~/CPE232_Denzel_Zamora$ ansible-playbook --ask-become-p
nstall_apache.yml
BECOME password:

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

TASK [Gathering Facts] ***************************************************
*
ok: [192.168.56.106]
ok: [192.168.56.105]
ok: [dzamora@192.168.56.110]

TASK [install apache2 and php packages for Ubuntu] ***********************
*
skipping: [dzamora@192.168.56.110]
ok: [192.168.56.105]
ok: [192.168.56.106]

TASK [install apache and php packages for CentOS] ************************
*
skipping: [192.168.56.105]
skipping: [192.168.56.106]
ok: [dzamora@192.168.56.110]

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

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.



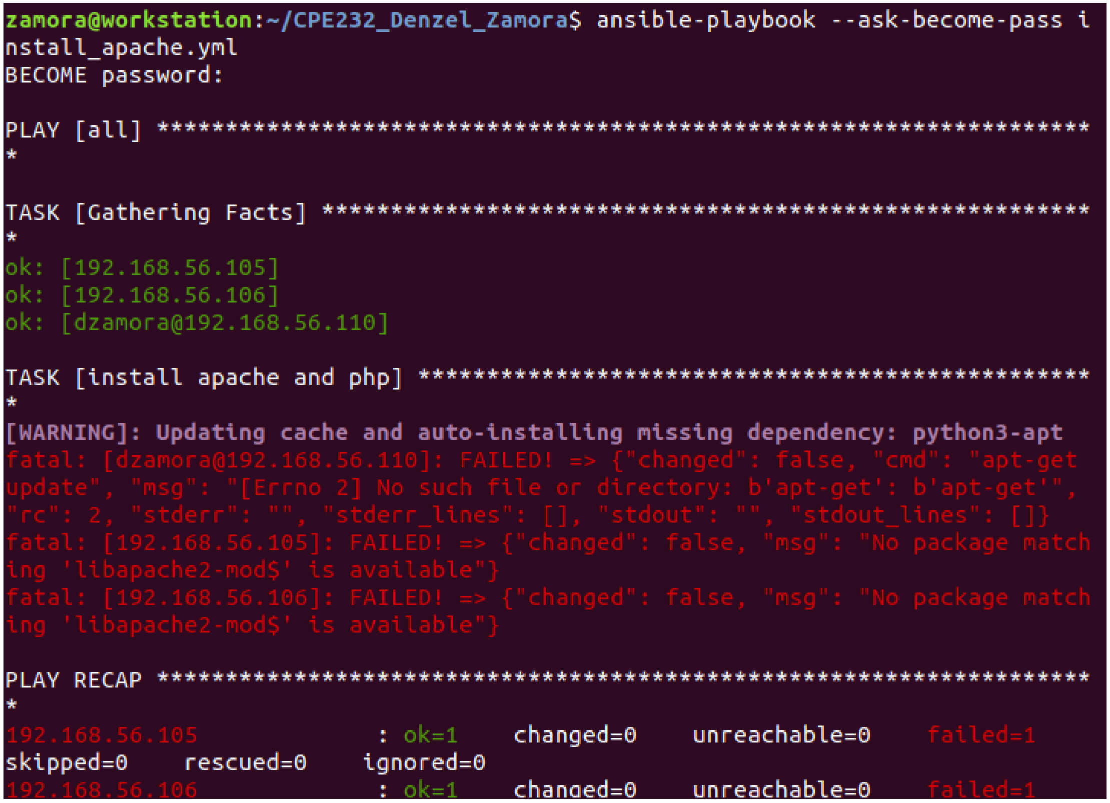
```
---
- hosts: all
  become: true
  tasks:

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


zamora@workstation: ~/CPE232_Denzel_Zamora
File Edit View Search Terminal Help

```
  GNU nano 2.9.3                          install_apache.yml

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

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

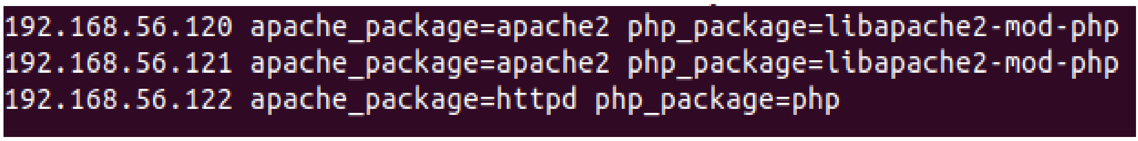
Run *ansible-playbook --ask-become-pass install_apache.yml* and describe the result.

We further shortened the command and combined the commands for Ubuntu and CentOS but the playbook did not run properly as the syntax for apt should be changed since CentOS cannot read the syntax apt.



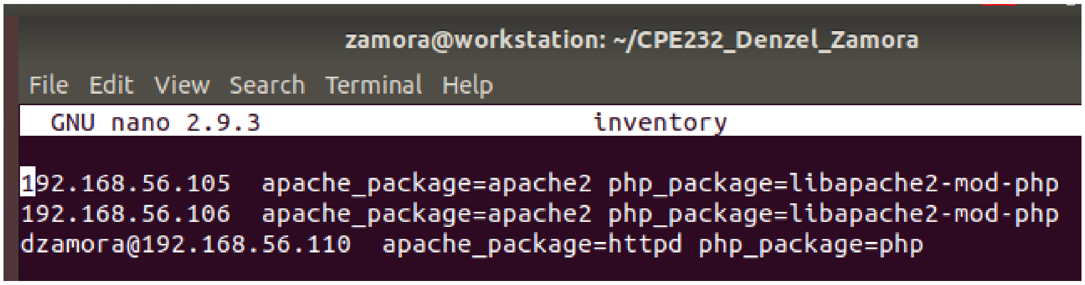
```
zamora@workstation:~/CPE232_Denzel_Zamora$ ansible-playbook --ask-become-pass i
nstall_apache.yml
BECOME password:

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

TASK [Gathering Facts] *********************************************************
*
ok: [192.168.56.105]
ok: [192.168.56.106]
ok: [dzamora@192.168.56.110]

TASK [install apache and php] **************************************************
*
[WARNING]: Updating cache and auto-installing missing dependency: python3-apt
fatal: [dzamora@192.168.56.110]: FAILED! => {"changed": false, "cmd": "apt-get
update", "msg": "[Errno 2] No such file or directory: b'apt-get': b'apt-get'",
"rc": 2, "stderr": "", "stderr_lines": [], "stdout": "", "stdout_lines": []}
fatal: [192.168.56.105]: FAILED! => {"changed": false, "msg": "No package match
ing 'libapache2-mod$' is available"}
fatal: [192.168.56.106]: FAILED! => {"changed": false, "msg": "No package match
ing 'libapache2-mod$' is available"}

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

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
```

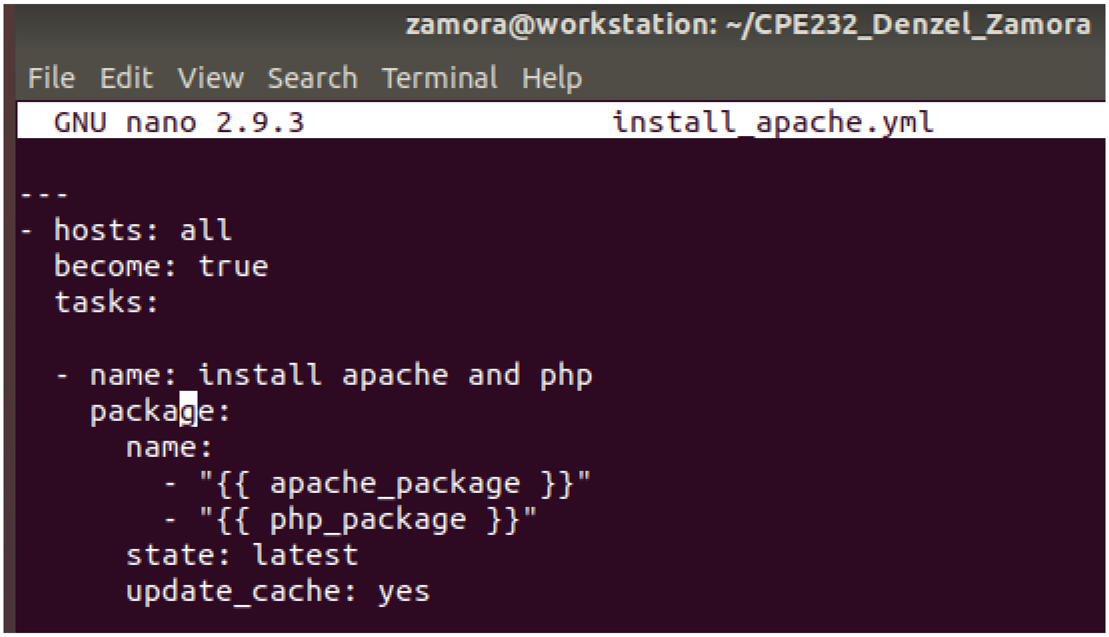
Make sure to save the *inventory* file and exit.



zamora@workstation: ~/CPE232_Denzel_Zamora
File Edit View Search Terminal Help

```
  GNU nano 2.9.3                      inventory

192.168.56.105   apache_package=apache2 php_package=libapache2-mod-php
192.168.56.106   apache_package=apache2 php_package=libapache2-mod-php
dzamora@192.168.56.110   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

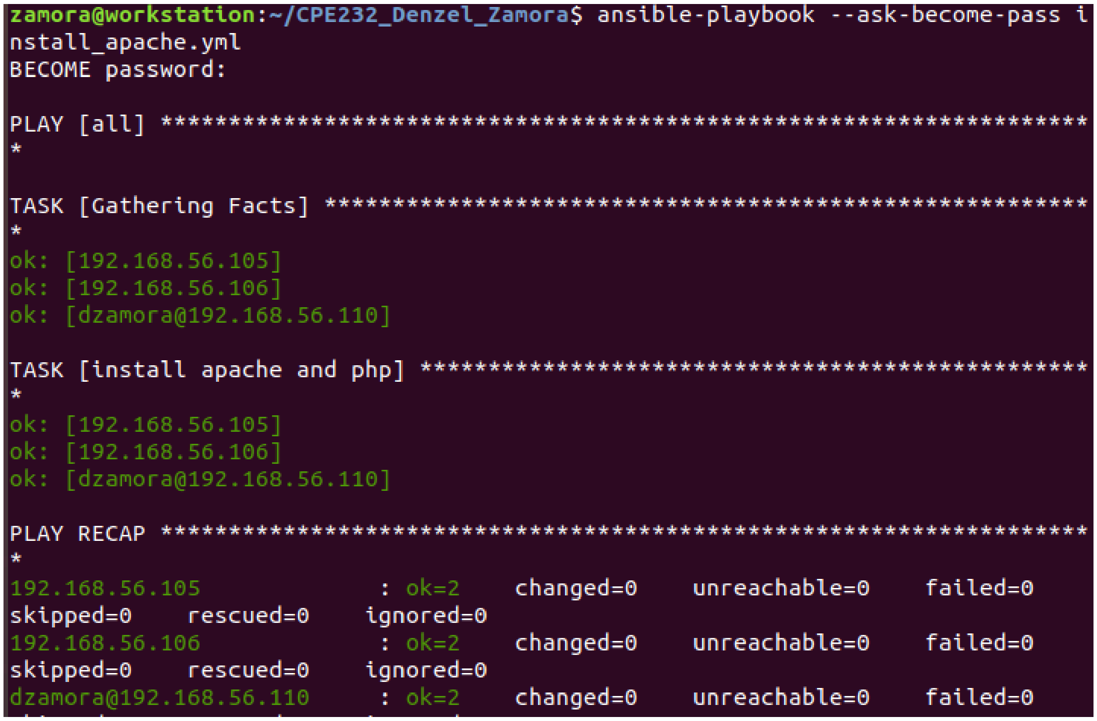

zamora@workstation: ~/CPE232_Denzel_Zamora
File Edit View Search Terminal Help
GNU nano 2.9.3
install_apache.yml

```
---
- 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.

Changing the apt syntax to package allowed CentOS to read it as a package to be installed therefore allowing it to be read by both the

Ubuntu and CentOS server.



```
zamora@workstation:~/CPE232_Denzel_Zamora$ ansible-playbook --ask-become-pass i
nstall_apache.yml
BECOME password:

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

TASK [Gathering Facts] *********************************************************
*
ok: [192.168.56.105]
ok: [192.168.56.106]
ok: [dzamora@192.168.56.110]

TASK [install apache and php] **************************************************
*
ok: [192.168.56.105]
ok: [192.168.56.106]
ok: [dzamora@192.168.56.110]

PLAY RECAP *********************************************************************
*
192.168.56.105             : ok=2    changed=0    unreachable=0    failed=0    
skipped=0    rescued=0    ignored=0   
192.168.56.106             : ok=2    changed=0    unreachable=0    failed=0    
skipped=0    rescued=0    ignored=0   
dzamora@192.168.56.110     : ok=2    changed=0    unreachable=0    failed=0    
```

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 allows for cleaner codes in the playbook. Using this method, we are shortening the code input, making it much cleaner and easier to read. Furthermore, it can also run multiple tasks just by having a shorter code.**

2. When do we use the “when” command in the playbook?
    - **To give it a condition that the command can only run when the condition of the statement has been met.**