Name: Solis, Paul Vincent M.	**Date Performed:9/3/25**
Course/Section: CPE212 - CPE31S2	**Date Submitted:9/3/25**
Instructor: Engr. Robin Valenzuela	**Semester and SY: 1st Sem yr 25-26**

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:

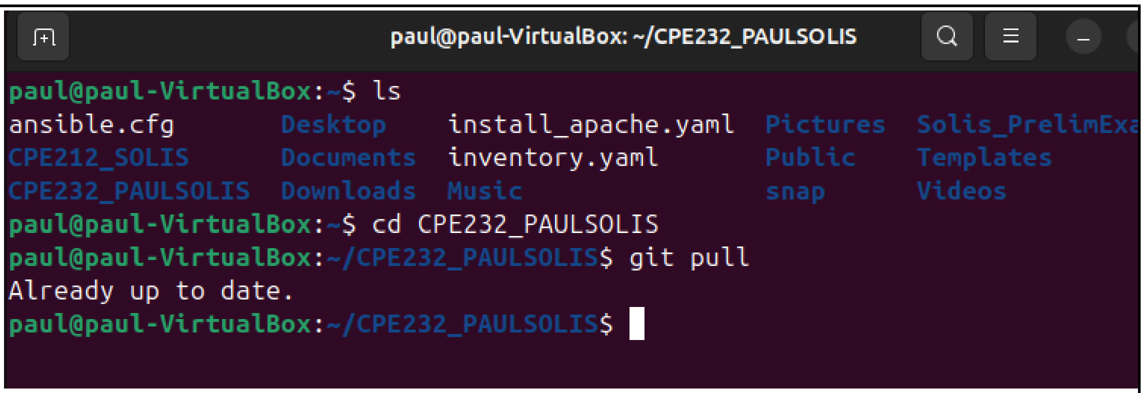
We are going to look at a way that we can differentiate a playbook by a host in terms of which distribution the host is running. It's very common in most Linux shops to run multiple distributions, for example, Ubuntu shop or Debian shop and you need a different distribution for a one off-case or perhaps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

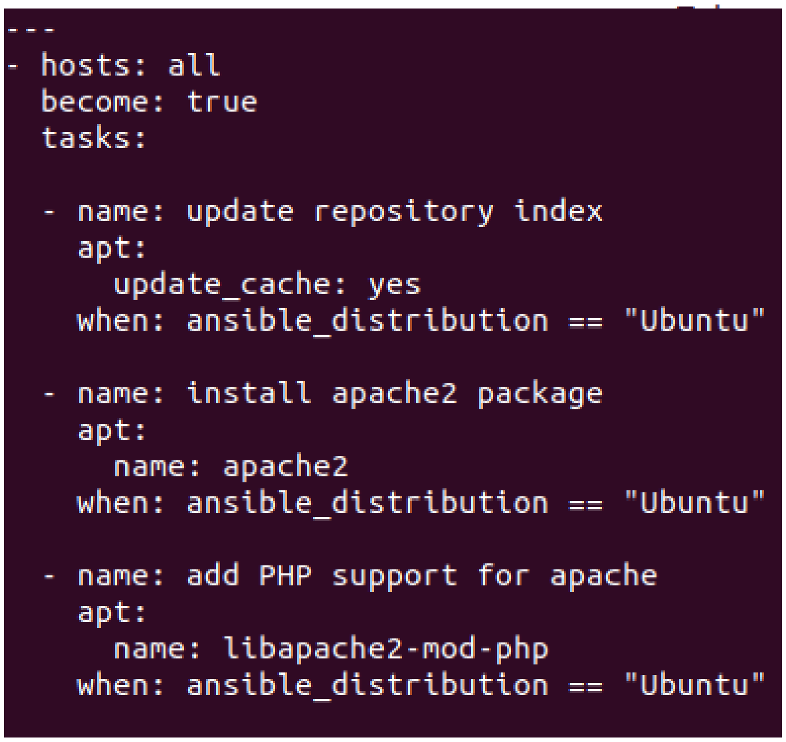
1. In the local machine, make sure you are in the local repository directory (*CPE232_yourname*). Issue the command git pull. When prompted, enter the correct passphrase or password. Describe what happened when you issue this command. Did something happen? Why?



```
paul@paul-VirtualBox:~$ ls
ansible.cfg        Desktop     install_apache.yaml  Pictures   Solis_PrelimExa
CPE212_SOLIS       Documents   inventory.yaml       Public     Templates
CPE232_PAULSOLIS   Downloads   Music                snap       Videos
paul@paul-VirtualBox:~$ cd CPE232_PAULSOLIS
paul@paul-VirtualBox:~/CPE232_PAULSOLIS$ git pull
Already up to date.
paul@paul-VirtualBox:~/CPE232_PAULSOLIS$ 
```

when i issue the command git pull it says already up to date because git pull used to fetch some and download and immediately update its github content.

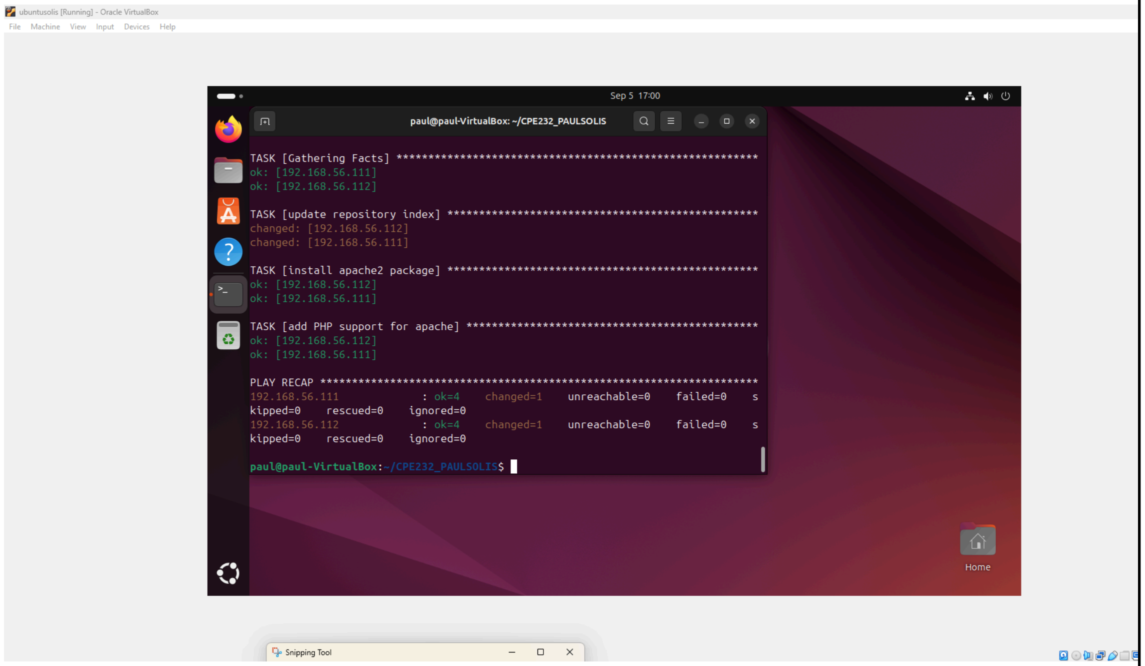
2. Edit the inventory file and add the IP address of the Centos VM. Issue the command we used to execute the playbook (the one we used in the last activity): *ansible-playbook --ask-become-pass install_apache.yml*. After executing this command, you may notice that it did not become successful in the 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."
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.


ubuntusolis [Running] - Oracle VirtualBox
File Machine View Input Devices Help
Sep 5 17:00
paul@paul-VirtualBox: ~/CPE232_PAULSOLIS
Home
Snipping Tool

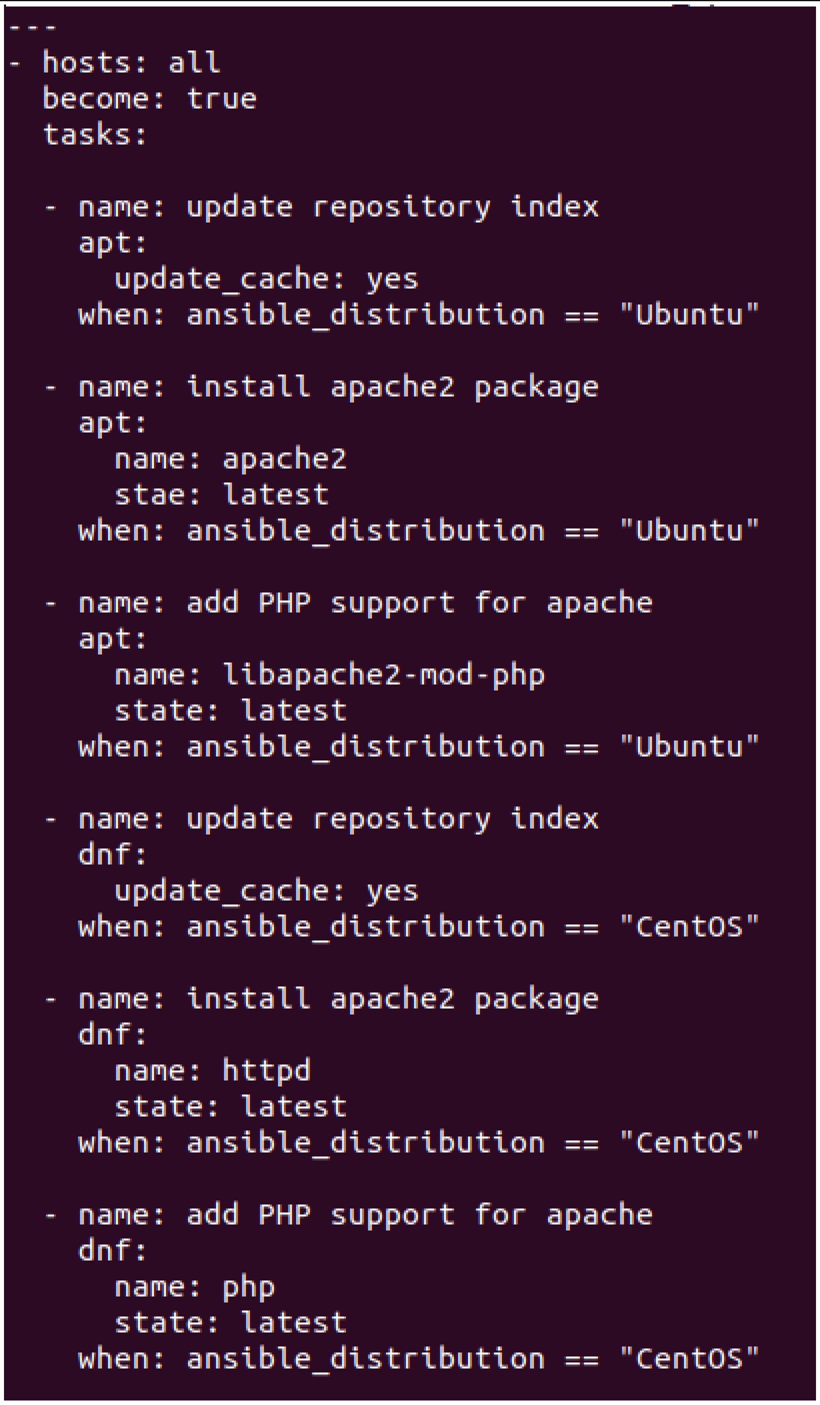
```
TASK [Gathering Facts] *********************************************************
ok: [192.168.56.111]
ok: [192.168.56.112]

TASK [update repository index] *************************************************
changed: [192.168.56.112]
changed: [192.168.56.111]

TASK [install apache2 package] *************************************************
ok: [192.168.56.112]
ok: [192.168.56.111]

TASK [add PHP support for apache] **********************************************
ok: [192.168.56.112]
ok: [192.168.56.111]

PLAY RECAP *********************************************************************
192.168.56.111             : ok=4    changed=1    unreachable=0    failed=0    s
kipped=0    rescued=0    ignored=0
192.168.56.112             : ok=4    changed=1    unreachable=0    failed=0    s
kipped=0    rescued=0    ignored=0

paul@paul-VirtualBox:~/CPE232_PAULSOLIS$ 
```

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.

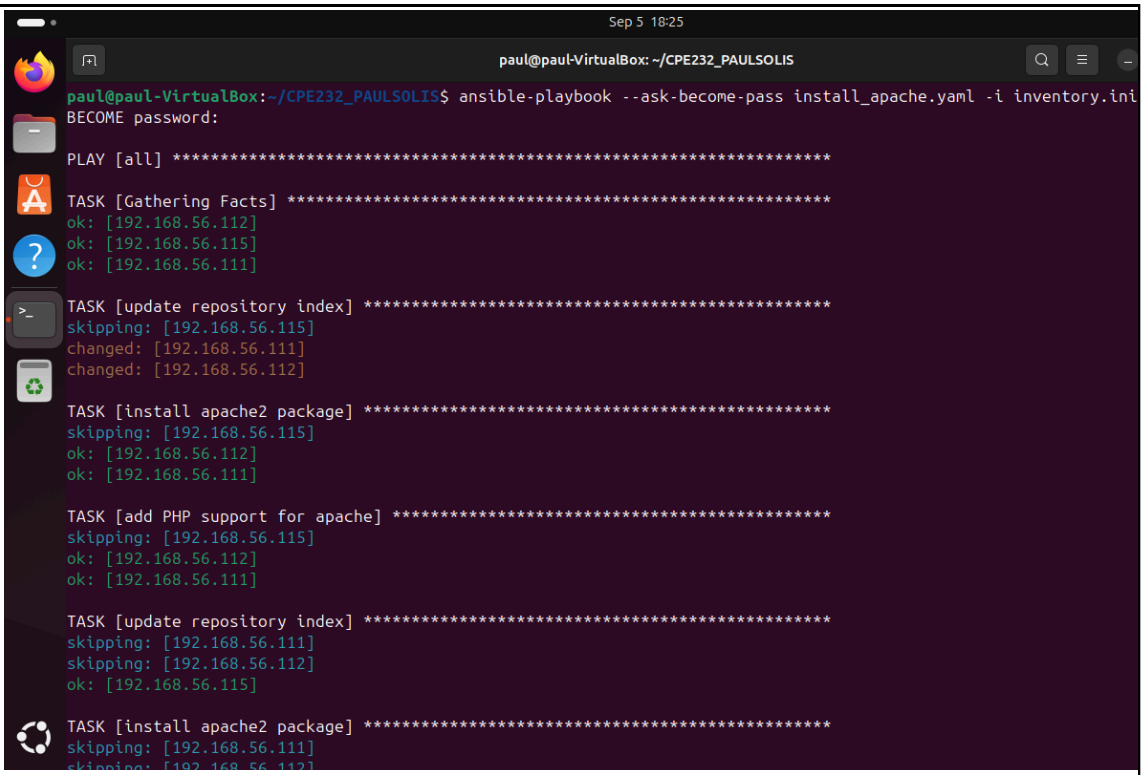
4. Edit the *install_apache.yml* file and insert the lines shown below.



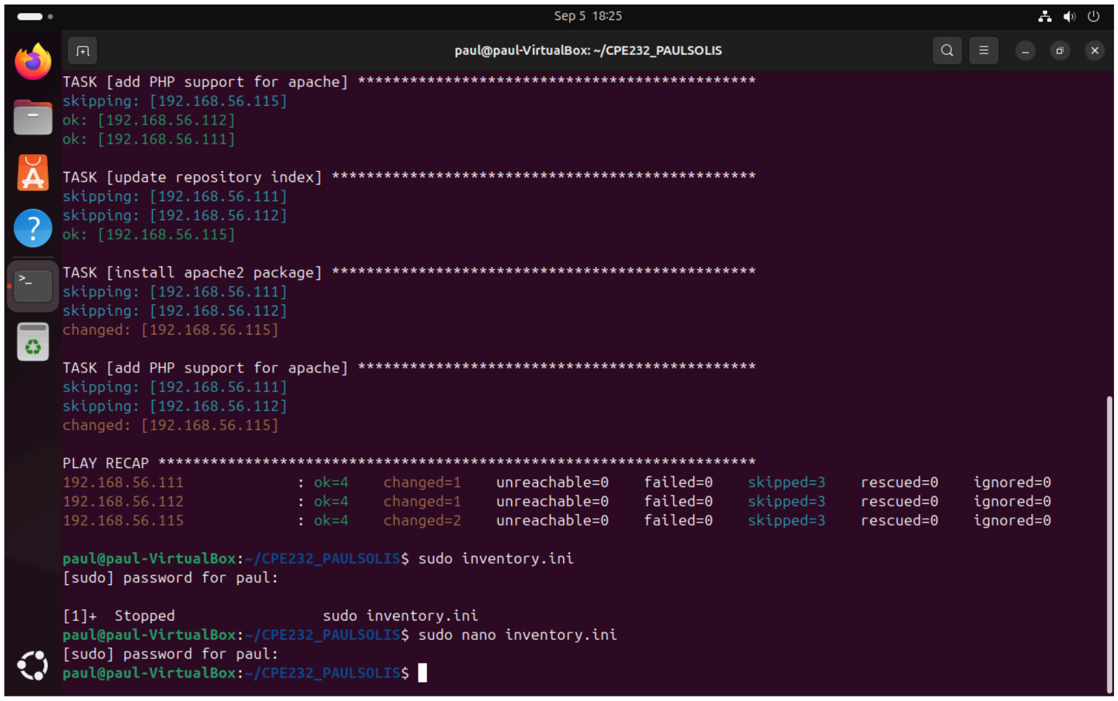
```
---
- hosts: all
  become: true
  tasks:

  - 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.

Run *ansible-playbook --ask-become-pass install_apache.yml* and describe the result.
it was a successful execution. It successfully ran on my 3 nodes
it skipped some yum on some other node but ok for my centos and for my centos it skipped some for some apt update.



```
paul@paul-VirtualBox:~/CPE232_PAULSOLIS$ ansible-playbook --ask-become-pass install_apache.yaml -i inventory.ini
BECOME password:

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

TASK [Gathering Facts] *********************************************************
ok: [192.168.56.112]
ok: [192.168.56.115]
ok: [192.168.56.111]

TASK [update repository index] *************************************************
skipping: [192.168.56.115]
changed: [192.168.56.111]
changed: [192.168.56.112]

TASK [install apache2 package] *************************************************
skipping: [192.168.56.115]
ok: [192.168.56.112]
ok: [192.168.56.111]

TASK [add PHP support for apache] **********************************************
skipping: [192.168.56.115]
ok: [192.168.56.112]
ok: [192.168.56.111]

TASK [update repository index] *************************************************
skipping: [192.168.56.111]
skipping: [192.168.56.112]
ok: [192.168.56.115]

TASK [install apache2 package] *************************************************
skipping: [192.168.56.111]
```

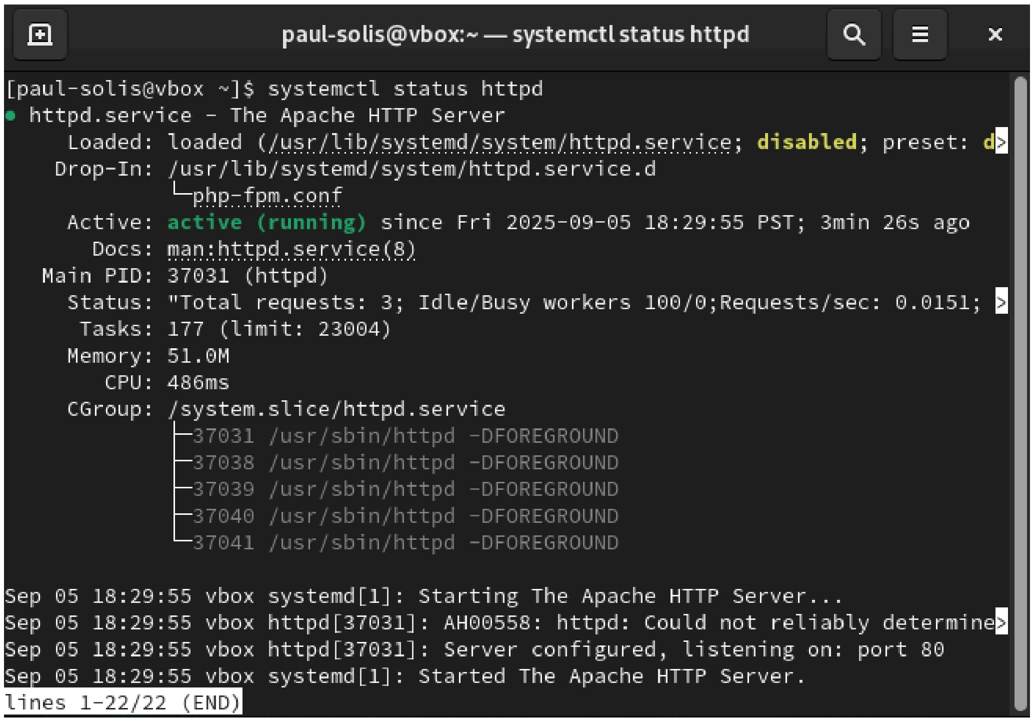


```
TASK [add PHP support for apache] **********************************************
skipping: [192.168.56.115]
ok: [192.168.56.112]
ok: [192.168.56.111]

TASK [update repository index] *************************************************
skipping: [192.168.56.111]
skipping: [192.168.56.112]
ok: [192.168.56.115]

TASK [install apache2 package] *************************************************
skipping: [192.168.56.111]
skipping: [192.168.56.112]
changed: [192.168.56.115]

TASK [add PHP support for apache] **********************************************
skipping: [192.168.56.111]
skipping: [192.168.56.112]
changed: [192.168.56.115]

PLAY RECAP *********************************************************************
192.168.56.111             : ok=4    changed=1    unreachable=0    failed=0    skipped=3    rescued=0    ignored=0
192.168.56.112             : ok=4    changed=1    unreachable=0    failed=0    skipped=3    rescued=0    ignored=0
192.168.56.115             : ok=4    changed=2    unreachable=0    failed=0    skipped=3    rescued=0    ignored=0

paul@paul-VirtualBox:~/CPE232_PAULSOLIS$ sudo inventory.ini
[sudo] password for paul:

[1]+  Stopped                 sudo inventory.ini
paul@paul-VirtualBox:~/CPE232_PAULSOLIS$ sudo nano inventory.ini
[sudo] password for paul:
paul@paul-VirtualBox:~/CPE232_PAULSOLIS$ 
```

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

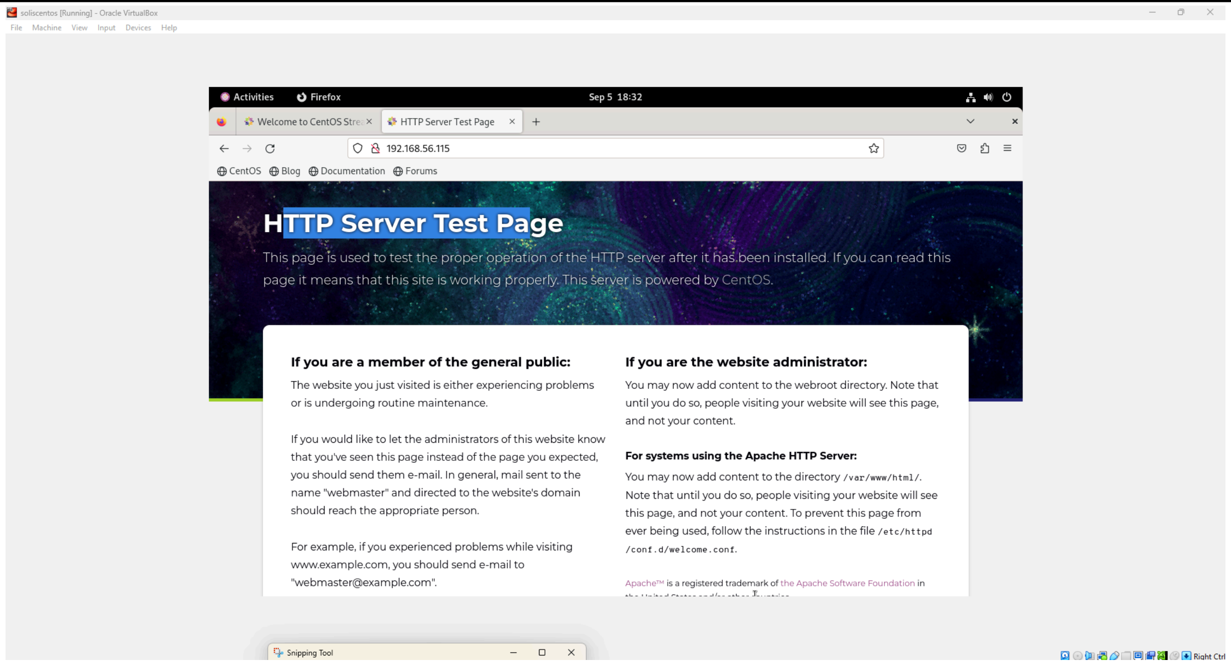


paul-solis@vbox:~ — systemctl status httpd

```
[paul-solis@vbox ~]$ systemctl status httpd
● httpd.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; preset: d>
    Drop-In: /usr/lib/systemd/system/httpd.service.d
             └─php-fpm.conf
     Active: active (running) since Fri 2025-09-05 18:29:55 PST; 3min 26s ago
       Docs: man:httpd.service(8)
   Main PID: 37031 (httpd)
     Status: "Total requests: 3; Idle/Busy workers 100/0;Requests/sec: 0.0151; >
      Tasks: 177 (limit: 23004)
     Memory: 51.0M
        CPU: 486ms
     CGroup: /system.slice/httpd.service
             ├─37031 /usr/sbin/httpd -DFOREGROUND
             ├─37038 /usr/sbin/httpd -DFOREGROUND
             ├─37039 /usr/sbin/httpd -DFOREGROUND
             ├─37040 /usr/sbin/httpd -DFOREGROUND
             └─37041 /usr/sbin/httpd -DFOREGROUND

Sep 05 18:29:55 vbox systemd[1]: Starting The Apache HTTP Server...
Sep 05 18:29:55 vbox httpd[37031]: AH00558: httpd: Could not reliably determine>
Sep 05 18:29:55 vbox httpd[37031]: Server configured, listening on: port 80
Sep 05 18:29:55 vbox systemd[1]: Started The Apache HTTP Server.
lines 1-22/22 (END)
```

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 th

5.4 e 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.

For example, if you experienced problems while visiting www.example.com, you should send e-mail to "webmaster@example.com".

If you are the website administrator:

You may now add content to the webroot directory. Note that until you do so, people visiting your website will see this page, and not your content.

For systems using the Apache HTTP Server:

You may now add content to the directory `/var/www/html/`. Note that until you do so, people visiting your website will see this page, and not your content. To prevent this page from ever being used, follow the instructions in the file `/etc/httpd/conf.d/welcome.conf`.

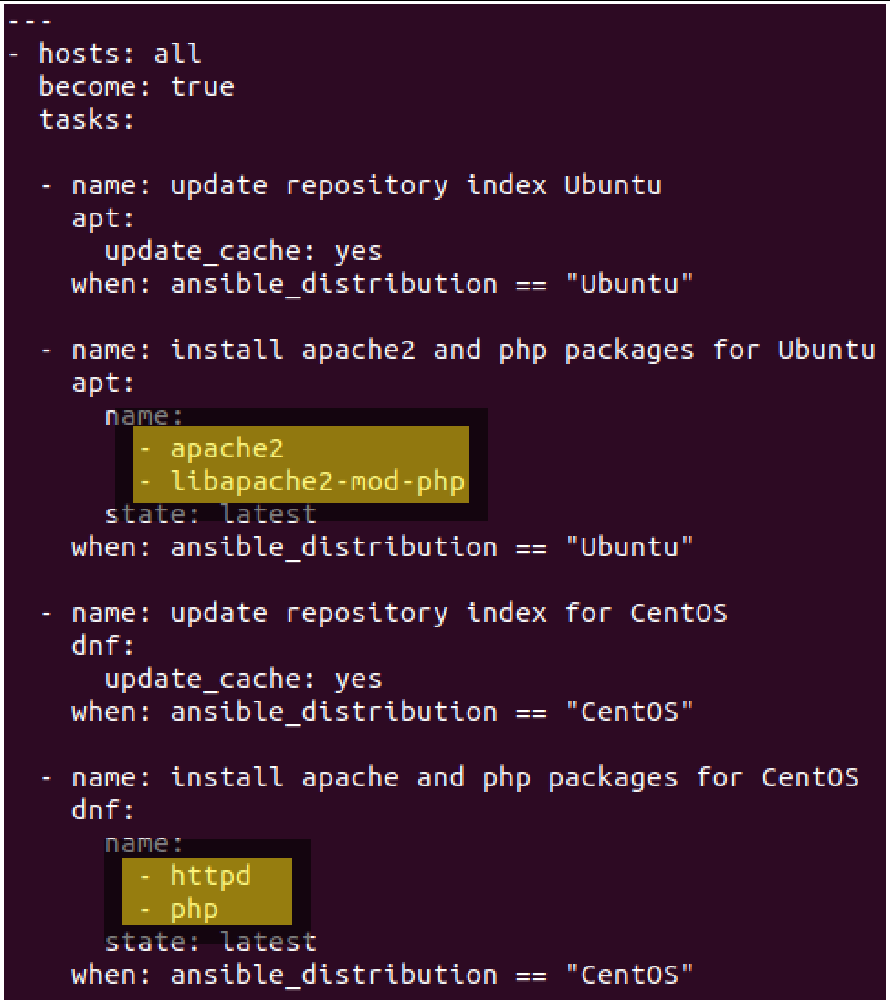
Apache™ is a registered trademark of the Apache Software Foundation in

yes it was successful

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.

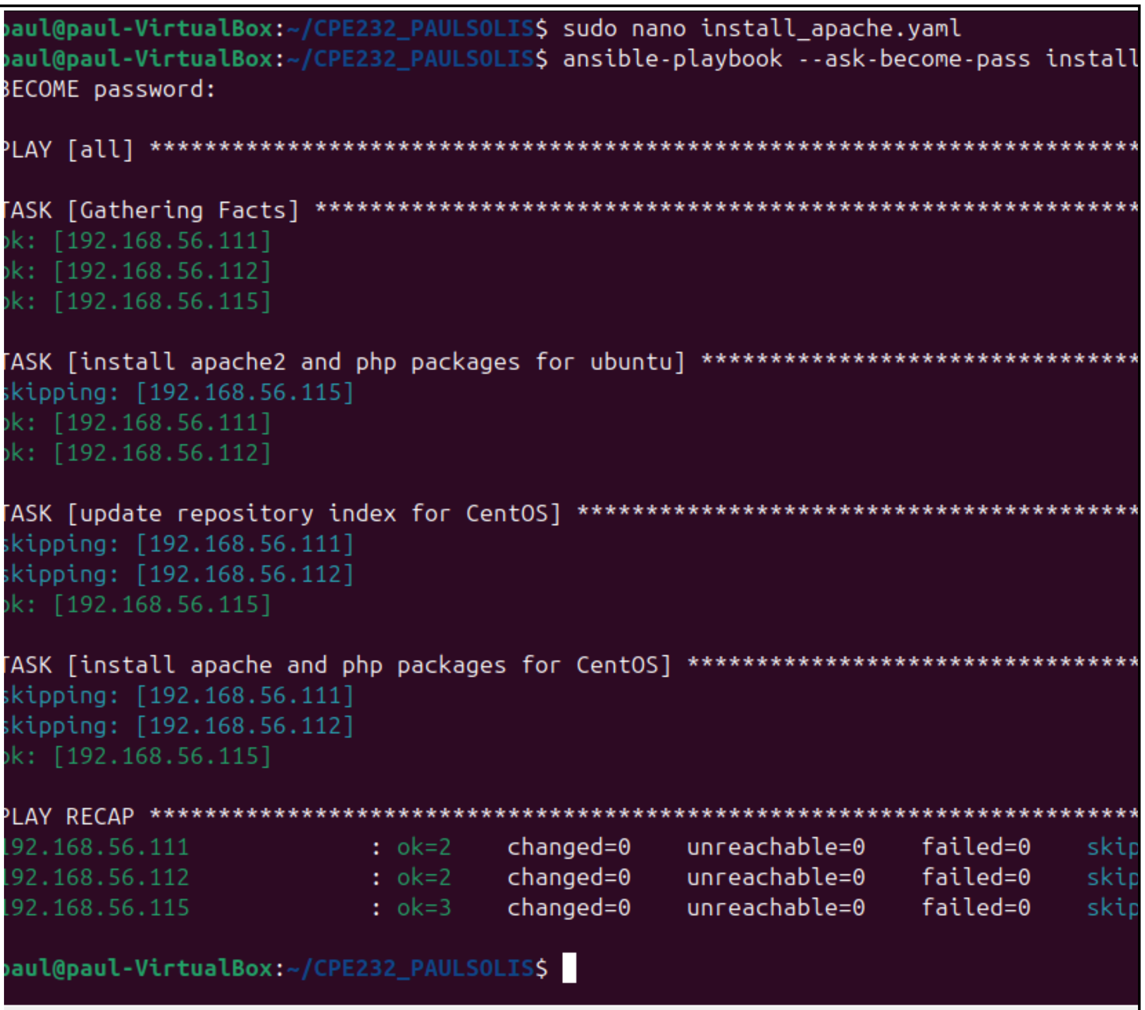
1. Edit the playbook *install_apache.yml*. Currently, we have three tasks targeting our Ubuntu machines and 3 tasks targeting our CentOS machine. Right now, we try to consolidate some tasks that are typically the same. For example, we can consolidate two plays that install packages. We can do that by 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"
```

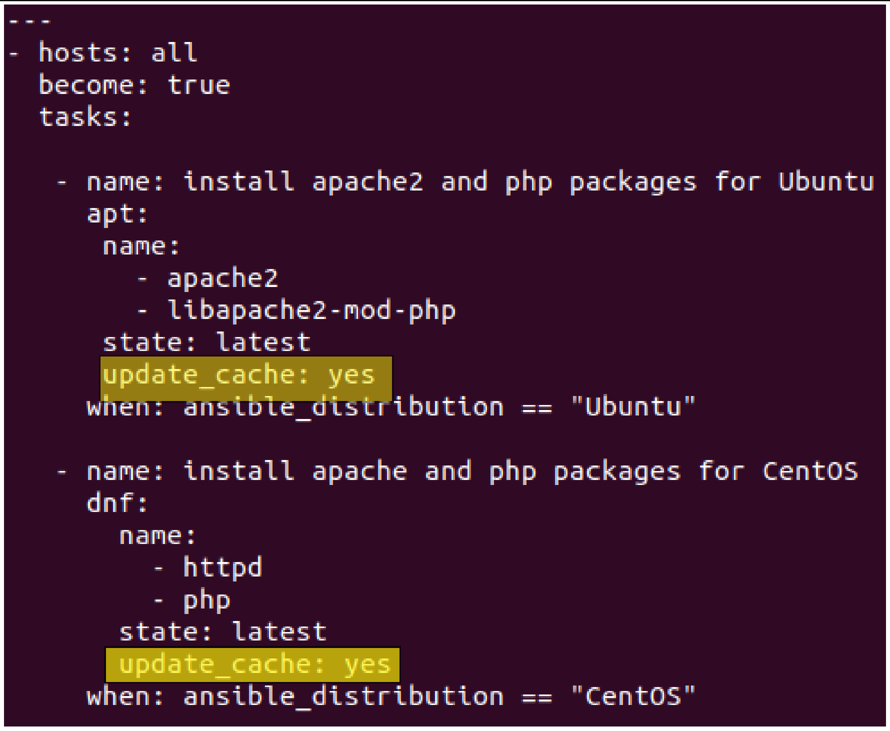
Make sure to save the file and exit.

Run *ansible-playbook --ask-become-pass install_apache.yml* and describe the result.
it executes of all of my 3 nodes



```
paul@paul-VirtualBox:~/CPE232_PAULSOLIS$ sudo nano install_apache.yaml
paul@paul-VirtualBox:~/CPE232_PAULSOLIS$ ansible-playbook --ask-become-pass install
BECOME password:

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

TASK [Gathering Facts] *************************************************************
ok: [192.168.56.111]
ok: [192.168.56.112]
ok: [192.168.56.115]

TASK [install apache2 and php packages for ubuntu] *********************************
skipping: [192.168.56.115]
ok: [192.168.56.111]
ok: [192.168.56.112]

TASK [update repository index for CentOS] ******************************************
skipping: [192.168.56.111]
skipping: [192.168.56.112]
ok: [192.168.56.115]

TASK [install apache and php packages for CentOS] **********************************
skipping: [192.168.56.111]
skipping: [192.168.56.112]
ok: [192.168.56.115]

PLAY RECAP *************************************************************************
192.168.56.111             : ok=2    changed=0    unreachable=0    failed=0    skip
192.168.56.112             : ok=2    changed=0    unreachable=0    failed=0    skip
192.168.56.115             : ok=3    changed=0    unreachable=0    failed=0    skip

paul@paul-VirtualBox:~/CPE232_PAULSOLIS$
```

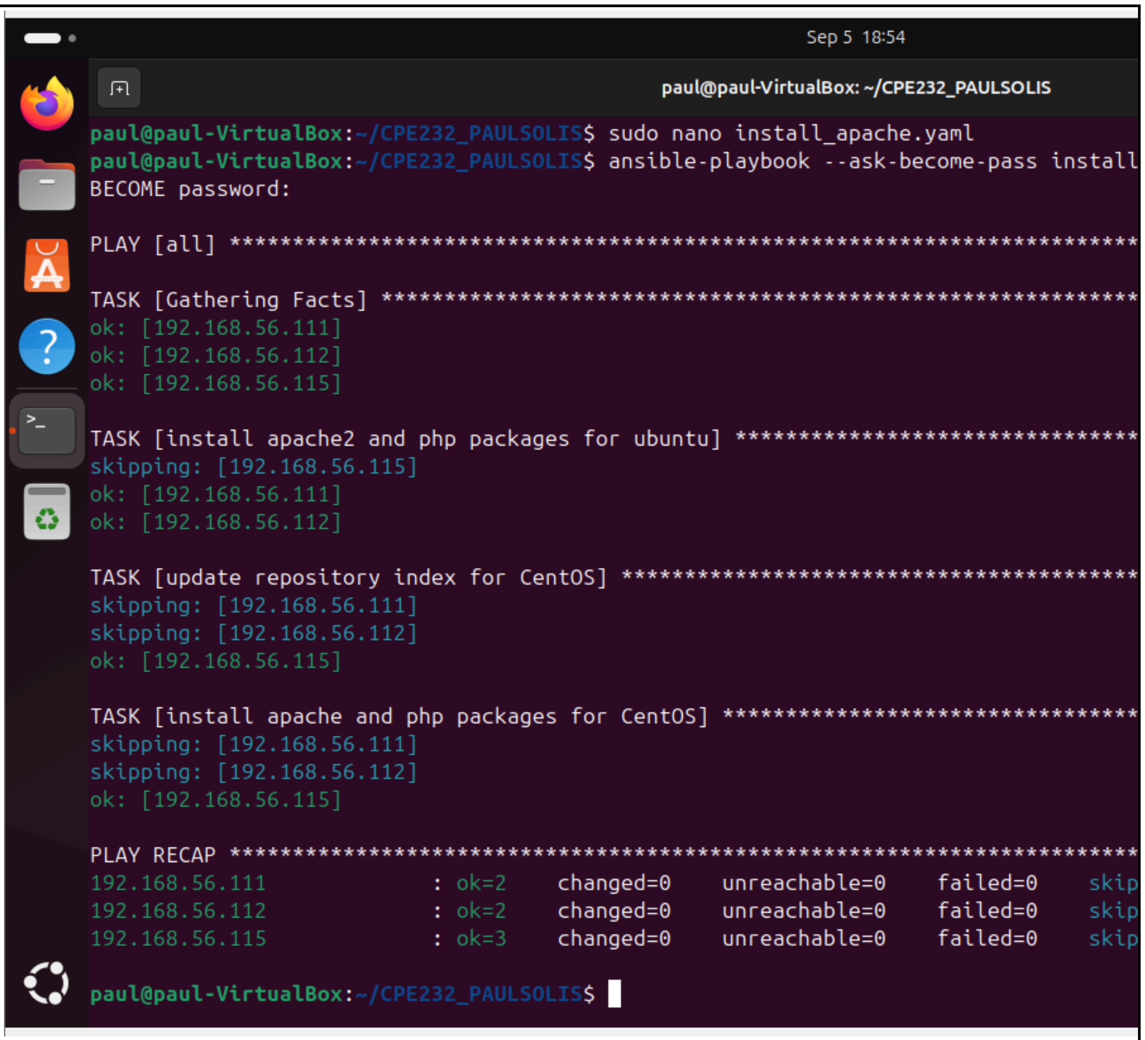
2. Edit the playbook *install_apache.yml* again. In task 2.1, we consolidated the plays into one play. This time we can actually consolidated everything in just 2 plays. This can be done by removing the update repository play and putting the command *update_cache: yes* below the command *state: latest*.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

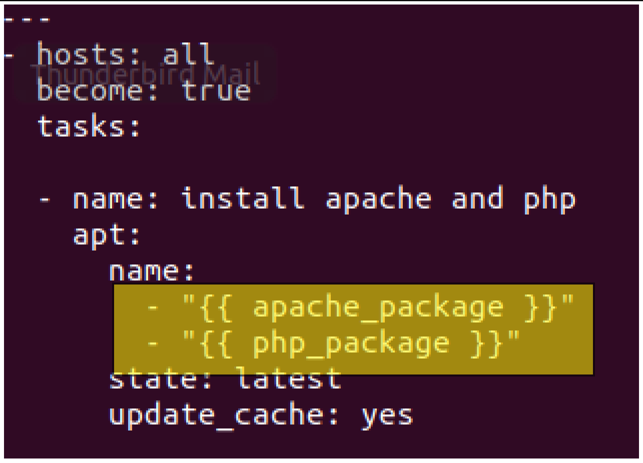
Run *ansible-playbook --ask-become-pass install_apache.yml* and describe the result.



```
paul@paul-VirtualBox:~/CPE232_PAULSOLIS$ sudo nano install_apache.yaml
paul@paul-VirtualBox:~/CPE232_PAULSOLIS$ ansible-playbook --ask-become-pass install
BECOME password:

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

TASK [Gathering Facts] ********************************************************
ok: [192.168.56.111]
ok: [192.168.56.112]
ok: [192.168.56.115]

TASK [install apache2 and php packages for ubuntu] ****************************
skipping: [192.168.56.115]
ok: [192.168.56.111]
ok: [192.168.56.112]

TASK [update repository index for CentOS] *************************************
skipping: [192.168.56.111]
skipping: [192.168.56.112]
ok: [192.168.56.115]

TASK [install apache and php packages for CentOS] *****************************
skipping: [192.168.56.111]
skipping: [192.168.56.112]
ok: [192.168.56.115]

PLAY RECAP ********************************************************************
192.168.56.111             : ok=2    changed=0    unreachable=0    failed=0    skip
192.168.56.112             : ok=2    changed=0    unreachable=0    failed=0    skip
192.168.56.115             : ok=3    changed=0    unreachable=0    failed=0    skip

paul@paul-VirtualBox:~/CPE232_PAULSOLIS$
```

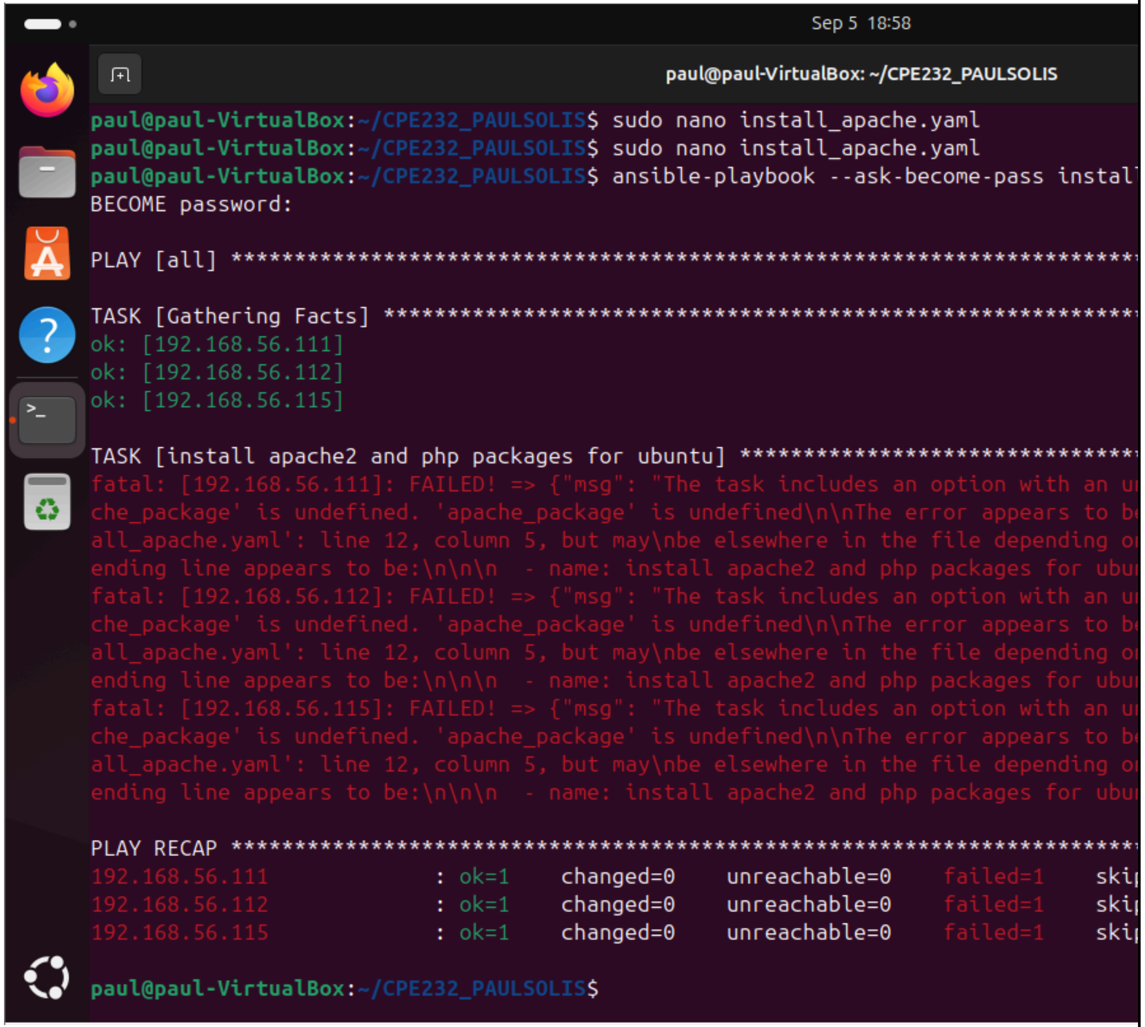
it updates the cache

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
```

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



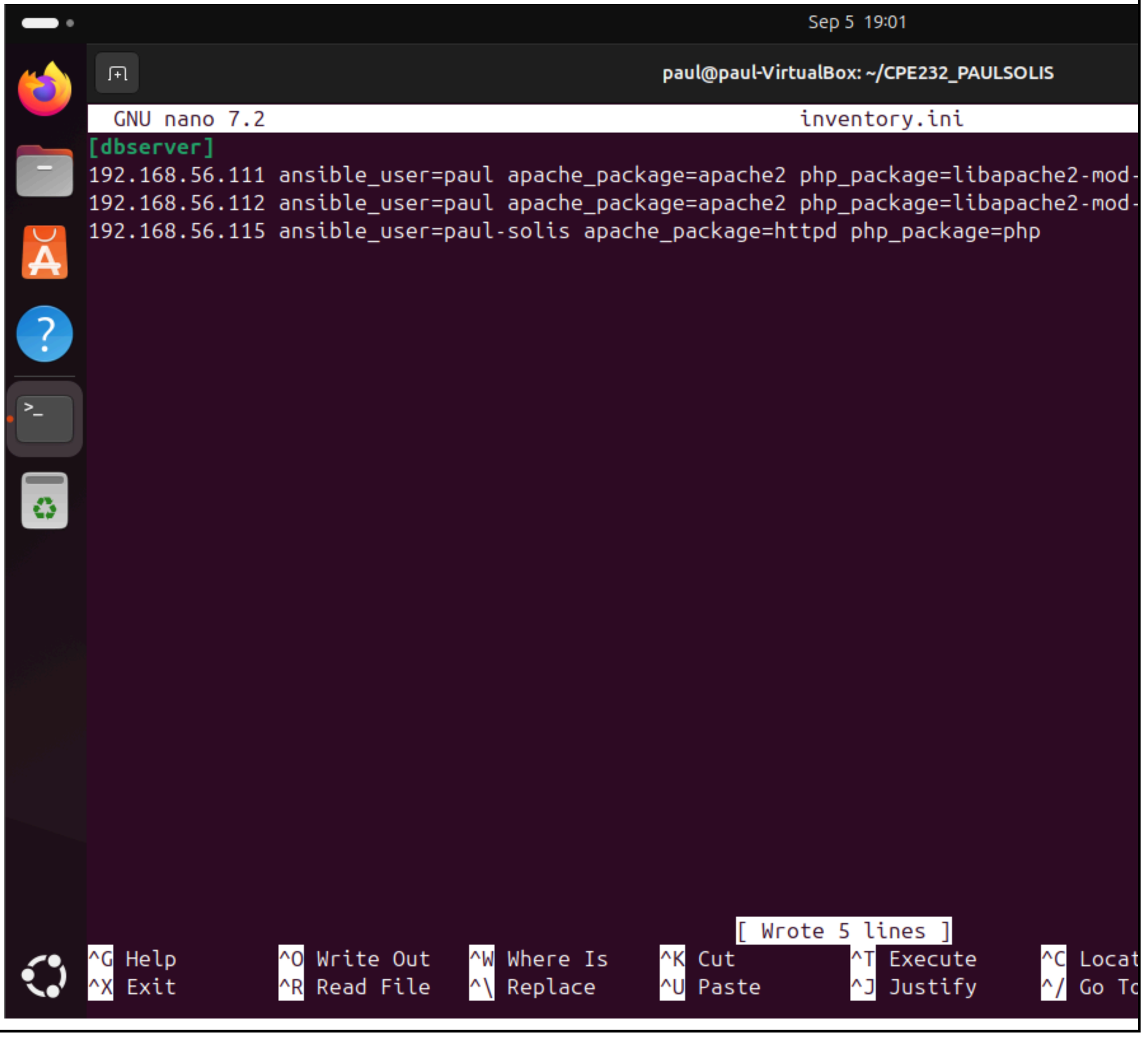
```
paul@paul-VirtualBox:~/CPE232_PAULSOLIS$ sudo nano install_apache.yaml
paul@paul-VirtualBox:~/CPE232_PAULSOLIS$ sudo nano install_apache.yaml
paul@paul-VirtualBox:~/CPE232_PAULSOLIS$ ansible-playbook --ask-become-pass instal
BECOME password:

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

TASK [Gathering Facts] *********************************************************
ok: [192.168.56.111]
ok: [192.168.56.112]
ok: [192.168.56.115]

TASK [install apache2 and php packages for ubuntu] *****************************
fatal: [192.168.56.111]: FAILED! => {"msg": "The task includes an option with an u
che_package' is undefined. 'apache_package' is undefined\n\nThe error appears to b
all_apache.yaml': line 12, column 5, but may\nbe elsewhere in the file depending o
ending line appears to be:\n\n\n  - name: install apache2 and php packages for ubu
fatal: [192.168.56.112]: FAILED! => {"msg": "The task includes an option with an u
che_package' is undefined. 'apache_package' is undefined\n\nThe error appears to b
all_apache.yaml': line 12, column 5, but may\nbe elsewhere in the file depending o
ending line appears to be:\n\n\n  - name: install apache2 and php packages for ubu
fatal: [192.168.56.115]: FAILED! => {"msg": "The task includes an option with an u
che_package' is undefined. 'apache_package' is undefined\n\nThe error appears to b
all_apache.yaml': line 12, column 5, but may\nbe elsewhere in the file depending o
ending line appears to be:\n\n\n  - name: install apache2 and php packages for ubu

PLAY RECAP *********************************************************************
192.168.56.111             : ok=1    changed=0    unreachable=0    failed=1    ski
192.168.56.112             : ok=1    changed=0    unreachable=0    failed=1    ski
192.168.56.115             : ok=1    changed=0    unreachable=0    failed=1    ski

paul@paul-VirtualBox:~/CPE232_PAULSOLIS$
```

it doesnt work because the variable werent defined yet in the inventory.

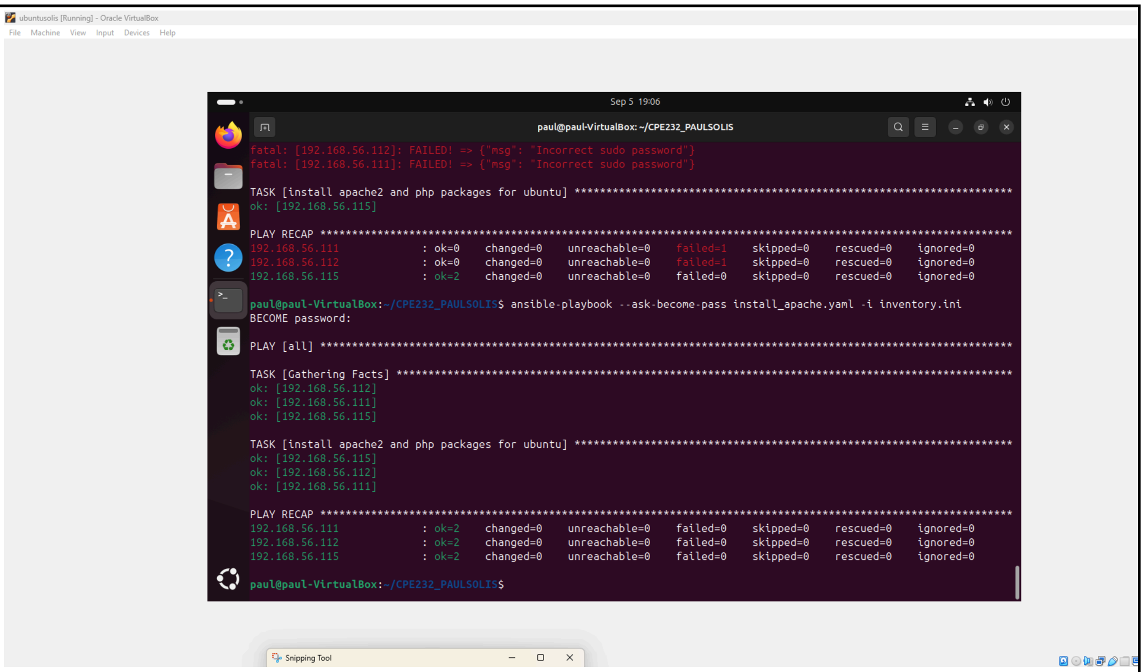
4. Unfortunately, task 2.3 was not successful. It's because we need to change something in the inventory file so that the variables we declared will be in place. Edit the *inventory* file and follow the below configuration:

```
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.



```
Sep 5 19:01
paul@paul-VirtualBox: ~/CPE232_PAULSOLIS
  GNU nano 7.2                                        inventory.ini
[dbserver]
192.168.56.111 ansible_user=paul apache_package=apache2 php_package=libapache2-mod-
192.168.56.112 ansible_user=paul apache_package=apache2 php_package=libapache2-mod-
192.168.56.115 ansible_user=paul-solis apache_package=httpd php_package=php

                                                [ Wrote 5 lines ]
^G Help        ^O Write Out   ^W Where Is    ^K Cut         ^T Execute     ^C Locat
^X Exit        ^R Read File   ^\ Replace     ^U Paste       ^J Justify     ^/ Go To
```



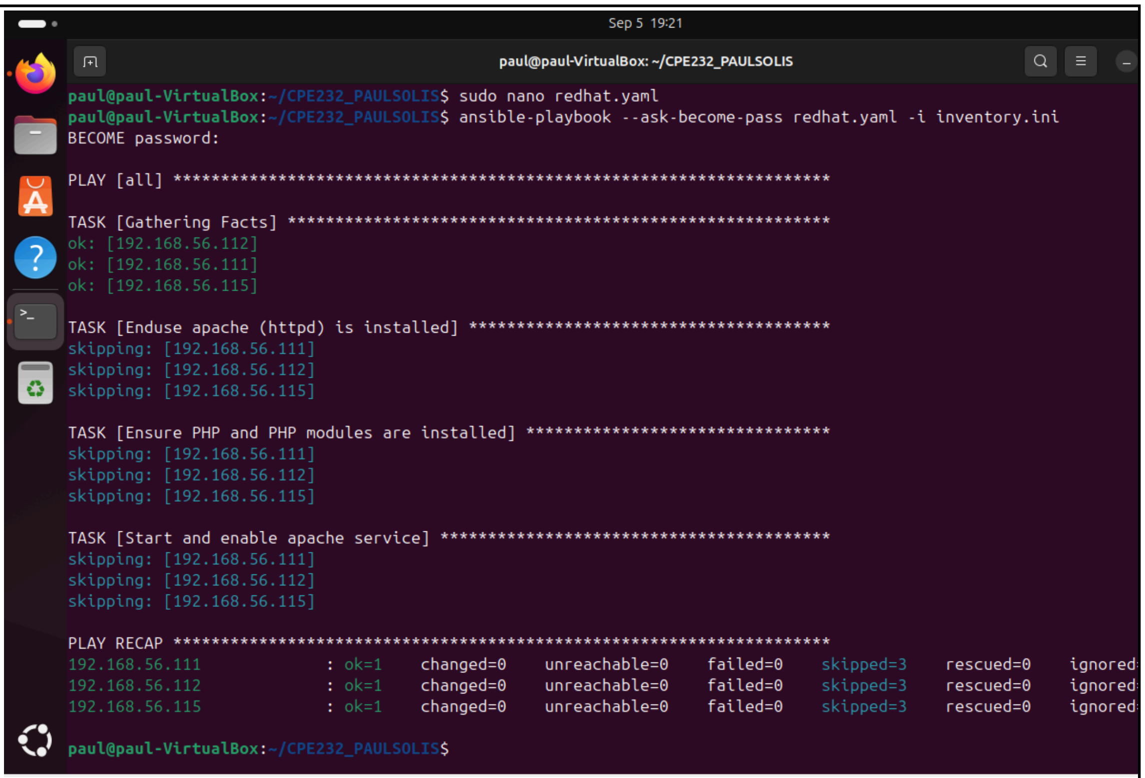
```
fatal: [192.168.56.112]: FAILED! => {"msg": "Incorrect sudo password"}
fatal: [192.168.56.111]: FAILED! => {"msg": "Incorrect sudo password"}

TASK [install apache2 and php packages for ubuntu] *****************************************************************
ok: [192.168.56.115]

PLAY RECAP ***********************************************************************************************************
192.168.56.111             : ok=0    changed=0    unreachable=0    failed=1    skipped=0    rescued=0    ignored=0
192.168.56.112             : ok=0    changed=0    unreachable=0    failed=1    skipped=0    rescued=0    ignored=0
192.168.56.115             : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

paul@paul-VirtualBox:~/CPE232_PAULSOLIS$ ansible-playbook --ask-become-pass install_apache.yaml -i inventory.ini
BECOME password:

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

TASK [Gathering Facts] ***********************************************************************************************
ok: [192.168.56.112]
ok: [192.168.56.111]
ok: [192.168.56.115]

TASK [install apache2 and php packages for ubuntu] *****************************************************************
ok: [192.168.56.115]
ok: [192.168.56.112]
ok: [192.168.56.111]

PLAY RECAP ***********************************************************************************************************
192.168.56.111             : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
192.168.56.112             : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
192.168.56.115             : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

paul@paul-VirtualBox:~/CPE232_PAULSOLIS$
```

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

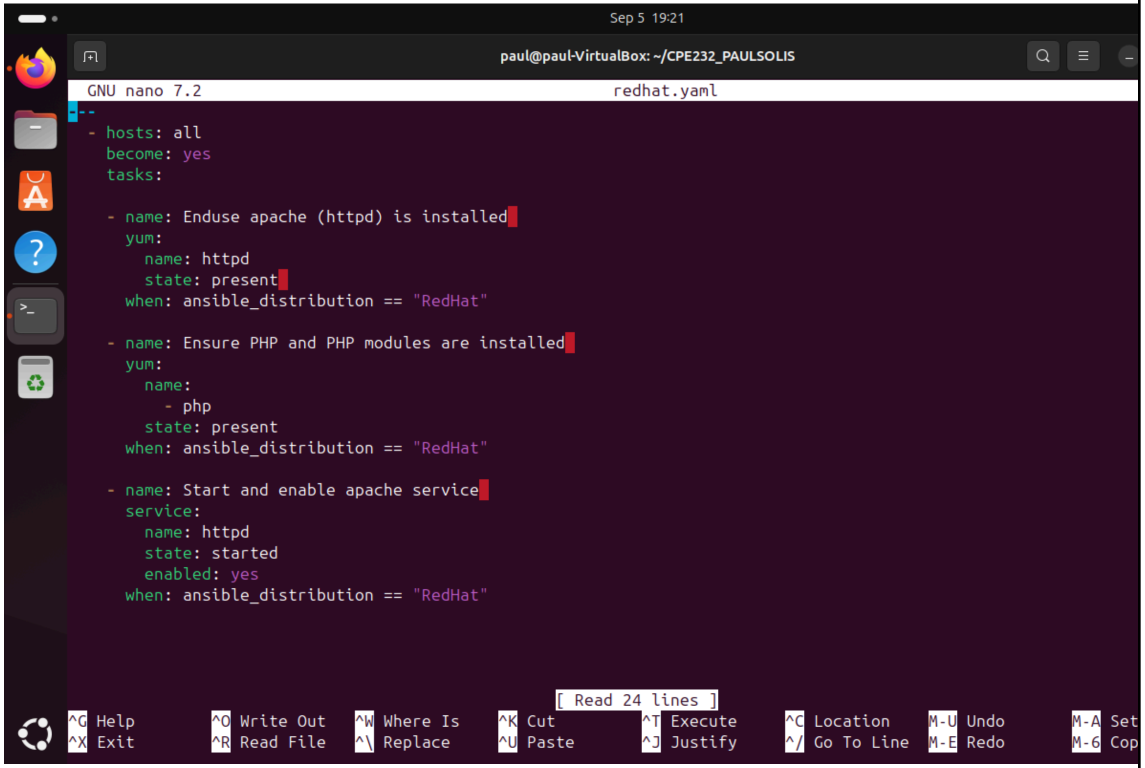
Run *ansible-playbook --ask-become-pass install_apache.yml* and describe the result.

Supplementary Activity:

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



```
paul@paul-VirtualBox:~/CPE232_PAULSOLIS$ sudo nano redhat.yaml
paul@paul-VirtualBox:~/CPE232_PAULSOLIS$ ansible-playbook --ask-become-pass redhat.yaml -i inventory.ini
BECOME password:

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

TASK [Gathering Facts] *********************************************************
ok: [192.168.56.112]
ok: [192.168.56.111]
ok: [192.168.56.115]

TASK [Enduse apache (httpd) is installed] **************************************
skipping: [192.168.56.111]
skipping: [192.168.56.112]
skipping: [192.168.56.115]

TASK [Ensure PHP and PHP modules are installed] ********************************
skipping: [192.168.56.111]
skipping: [192.168.56.112]
skipping: [192.168.56.115]

TASK [Start and enable apache service] *****************************************
skipping: [192.168.56.111]
skipping: [192.168.56.112]
skipping: [192.168.56.115]

PLAY RECAP *********************************************************************
192.168.56.111             : ok=1    changed=0    unreachable=0    failed=0    skipped=3    rescued=0    ignored
192.168.56.112             : ok=1    changed=0    unreachable=0    failed=0    skipped=3    rescued=0    ignored
192.168.56.115             : ok=1    changed=0    unreachable=0    failed=0    skipped=3    rescued=0    ignored

paul@paul-VirtualBox:~/CPE232_PAULSOLIS$
```



```
  GNU nano 7.2                          redhat.yaml
---
 - hosts: all
   become: yes
   tasks:

   - name: Enduse apache (httpd) is installed
     yum:
       name: httpd
       state: present
     when: ansible_distribution == "RedHat"

   - name: Ensure PHP and PHP modules are installed
     yum:
       name:
         - php
       state: present
     when: ansible_distribution == "RedHat"

   - name: Start and enable apache service
     service:
       name: httpd
       state: started
       enabled: yes
     when: ansible_distribution == "RedHat"

                              [ Read 24 lines ]
^G Help       ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location   M-U Undo      M-A Set
^X Exit       ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^/ Go To Line M-E Redo      M-6 Cop
```

Reflections:

Answer the following:

1. Why do you think refactoring of playbook codes is important?
   Refactoring makes the code more readable and understandable. This is especially important when multiple people are working on the same project or when you need to maintain the code over time.
2. When do we use the "when" command in playbook?

   The when command is used to apply conditional logic in an Ansible playbook. It allows you to run a task only when certain conditions are met, making your playbook more flexible and adaptable.