Name: Rio Marie G. Suzuki	**Date Performed:** 09/18/2023
Course/Section: CPE232 S6	**Date Submitted:** 09/18/2023
Instructor: Dr. Jonathan Taylar	**Semester and SY:** 1st sem 2023-2024

Activity 5: Consolidating Playbook plays

1. **Objectives:**

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

1.2 Apply refactoring techniques in cleaning up the playbook codes

2. **Discussion**:

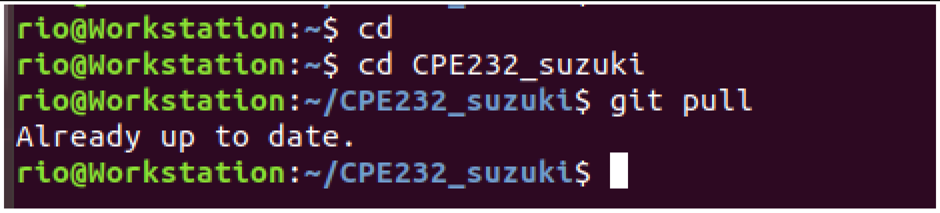
We are going to look at a way that we can differentiate a playbook by a host in terms of which distribution the host is running. It's very common in most Linux shops to run multiple distributions, for example, Ubuntu shop or Debian shop and you need a different distribution for a one off-case or perhaps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

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

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

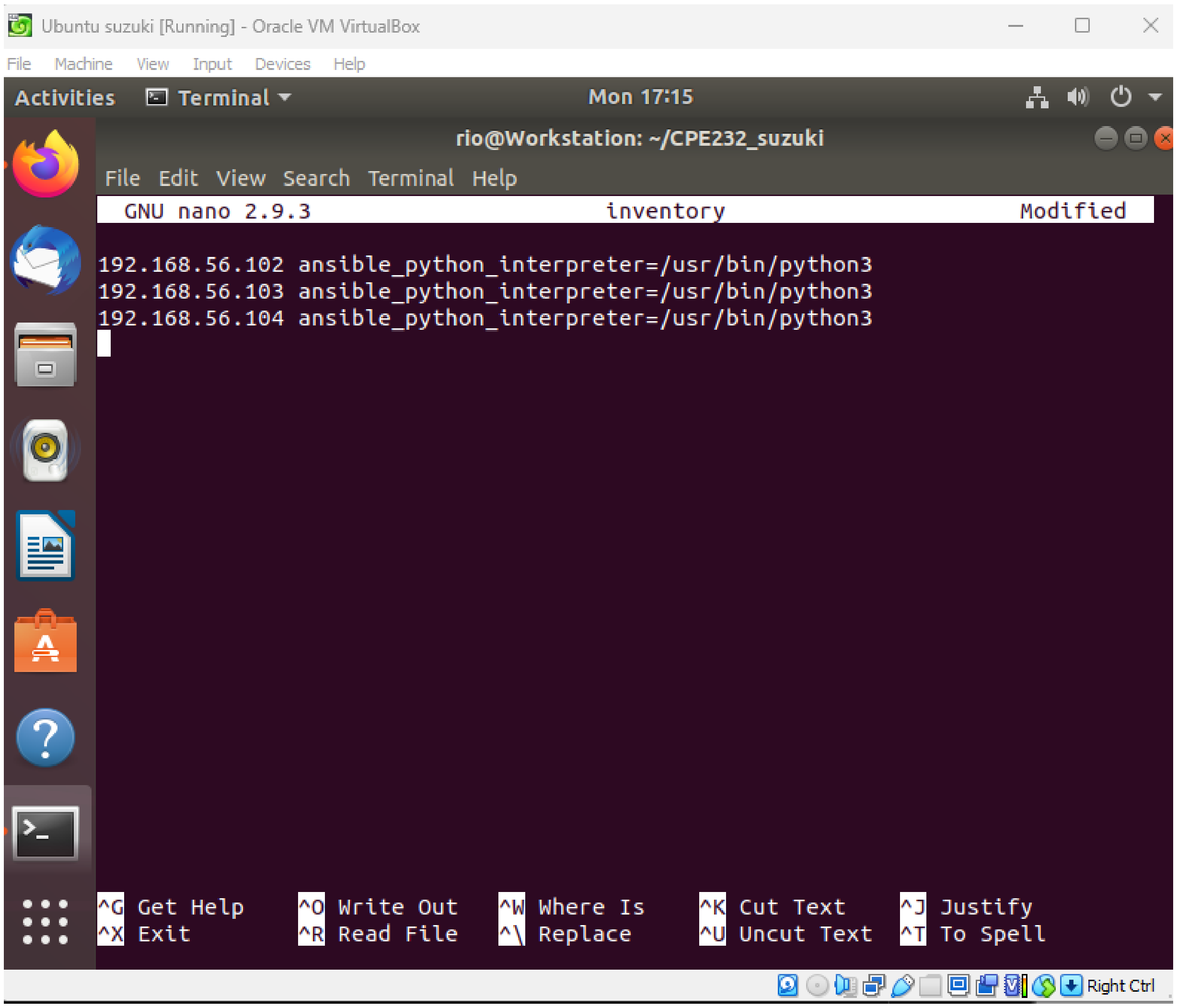
Task 1: Use when command for different distributions

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

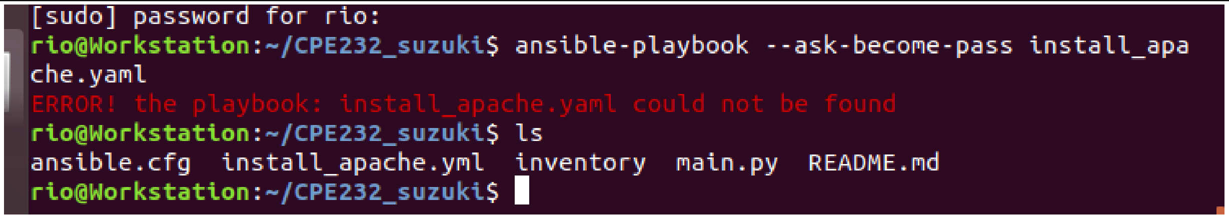


```
rio@Workstation:~$ cd
rio@Workstation:~$ cd CPE232_suzuki
rio@Workstation:~/CPE232_suzuki$ git pull
Already up to date.
rio@Workstation:~/CPE232_suzuki$ 
```

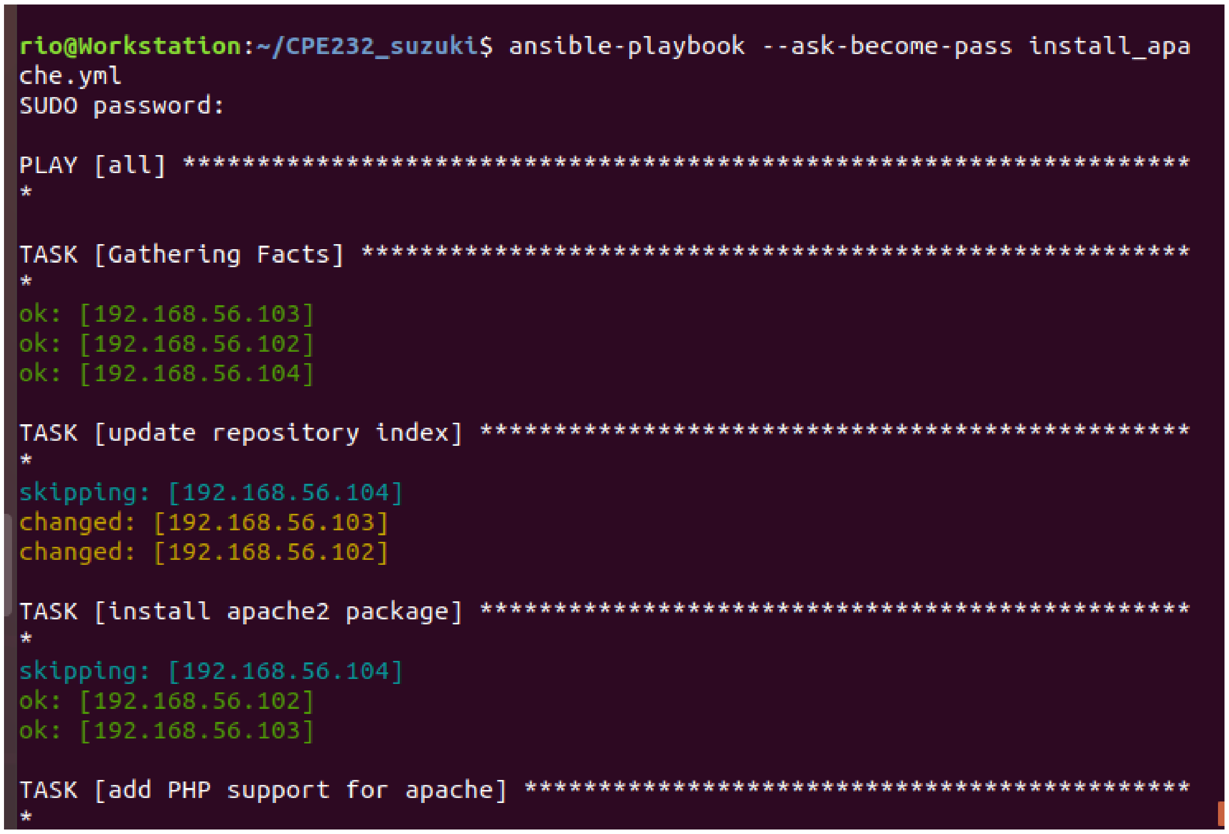
2. Edit the inventory file and add the IP address of the Centos VM. Issue the command we used to execute the playbook (the one we used in the last activity): *ansible-playbook --ask-become-pass install_apache.yml*. After executing this command, you may notice that it did not become successful in the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


Ubuntu suzuki [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities
Terminal
Mon 17:15
rio@Workstation: ~/CPE232_suzuki
File Edit View Search Terminal Help
Right Ctrl

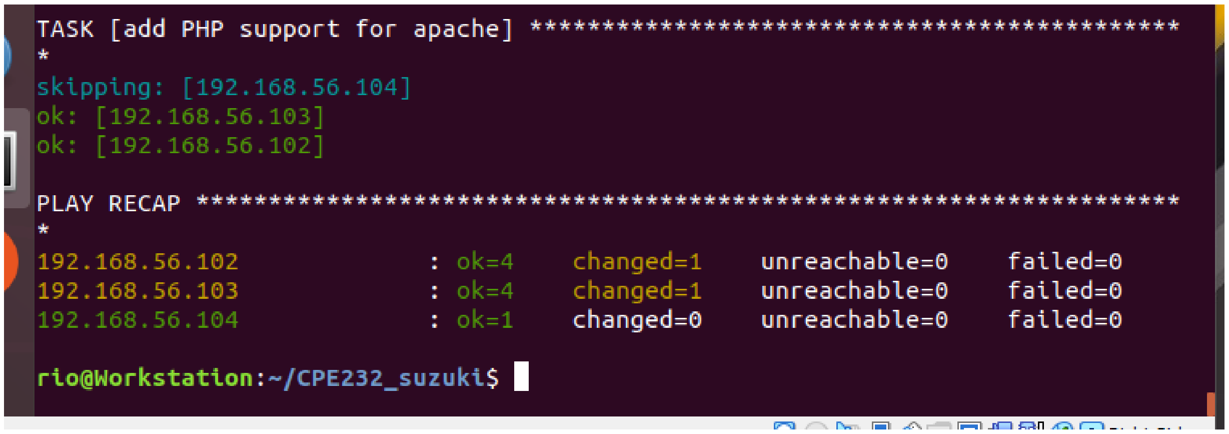
```
  GNU nano 2.9.3                    inventory                        Modified

192.168.56.102 ansible_python_interpreter=/usr/bin/python3
192.168.56.103 ansible_python_interpreter=/usr/bin/python3
192.168.56.104 ansible_python_interpreter=/usr/bin/python3


^G Get Help    ^O Write Out   ^W Where Is    ^K Cut Text    ^J Justify
^X Exit        ^R Read File   ^\ Replace     ^U Uncut Text  ^T To Spell
```



```
[sudo] password for rio:
rio@Workstation:~/CPE232_suzuki$ ansible-playbook --ask-become-pass install_apa
che.yaml
ERROR! the playbook: install_apache.yaml could not be found
rio@Workstation:~/CPE232_suzuki$ ls
ansible.cfg  install_apache.yml  inventory  main.py  README.md
rio@Workstation:~/CPE232_suzuki$ 
```

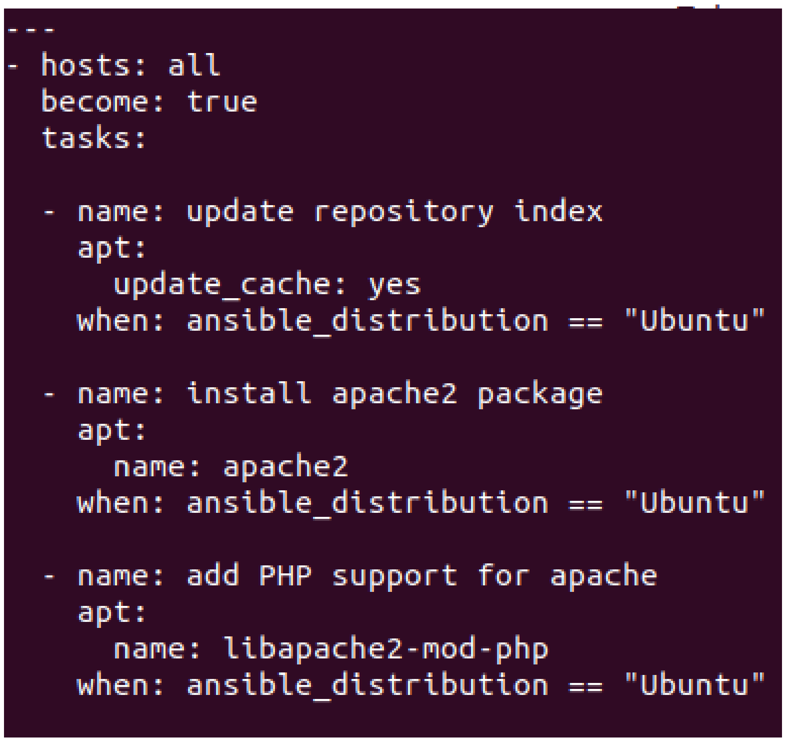


```
rio@Workstation:~/CPE232_suzuki$ ansible-playbook --ask-become-pass install_apa
che.yml
SUDO password:

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

TASK [Gathering Facts] *********************************************************
*
ok: [192.168.56.103]
ok: [192.168.56.102]
ok: [192.168.56.104]

TASK [update repository index] *************************************************
*
skipping: [192.168.56.104]
changed: [192.168.56.103]
changed: [192.168.56.102]

TASK [install apache2 package] *************************************************
*
skipping: [192.168.56.104]
ok: [192.168.56.102]
ok: [192.168.56.103]

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

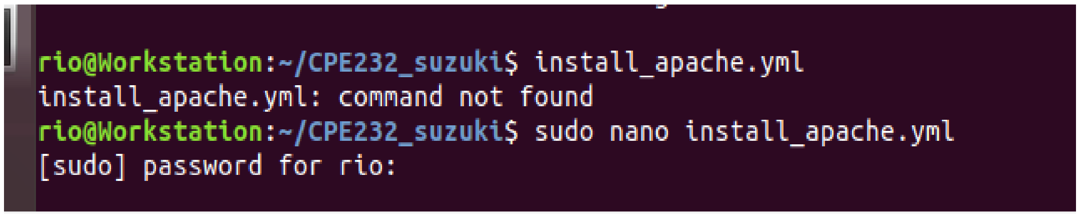


```
TASK [add PHP support for apache] **********************************************
*
skipping: [192.168.56.104]
ok: [192.168.56.103]
ok: [192.168.56.102]

PLAY RECAP *********************************************************************
*
192.168.56.102             : ok=4    changed=1    unreachable=0    failed=0
192.168.56.103             : ok=4    changed=1    unreachable=0    failed=0
192.168.56.104             : ok=1    changed=0    unreachable=0    failed=0

rio@Workstation:~/CPE232_suzuki$ 
```

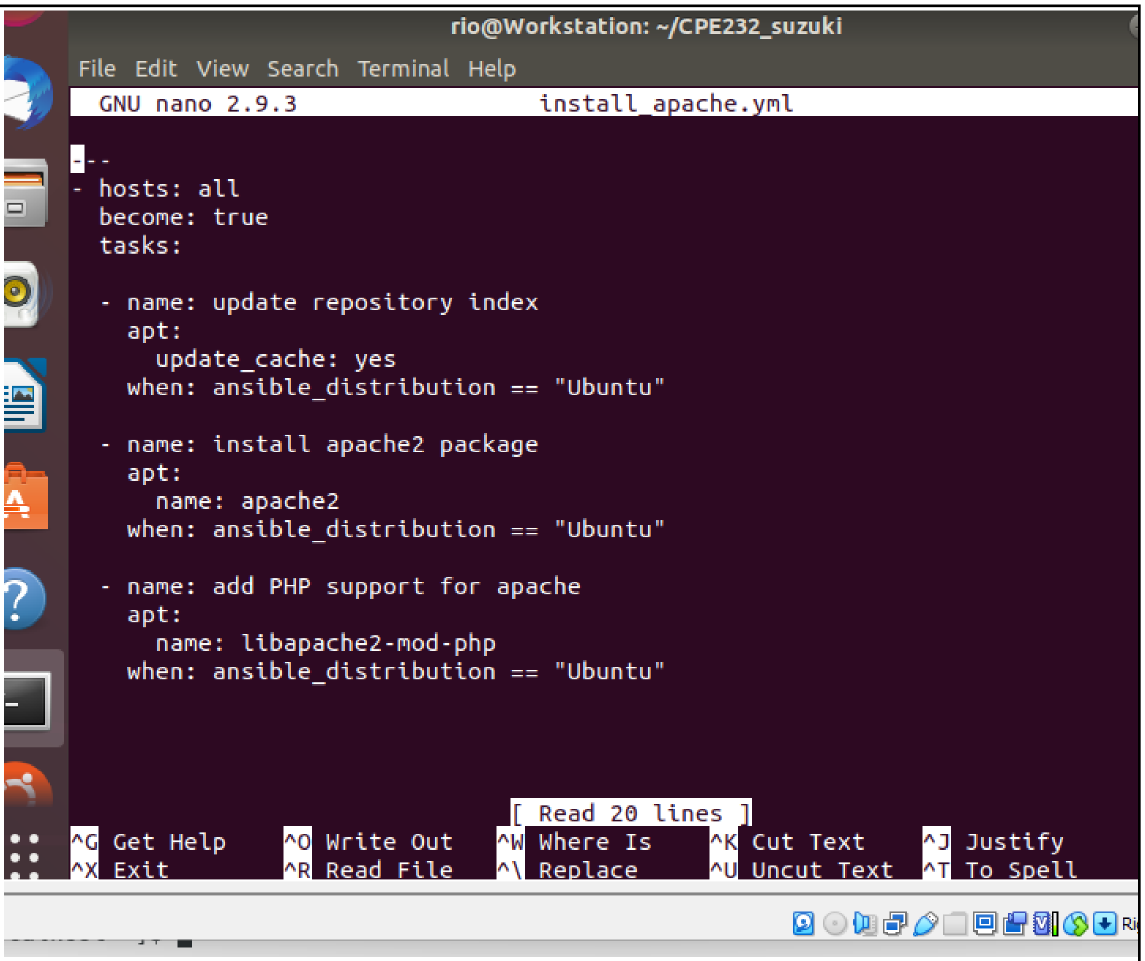
3. Edit the *install_apache.yml* file and insert the lines shown below.



```
---
- 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"
```



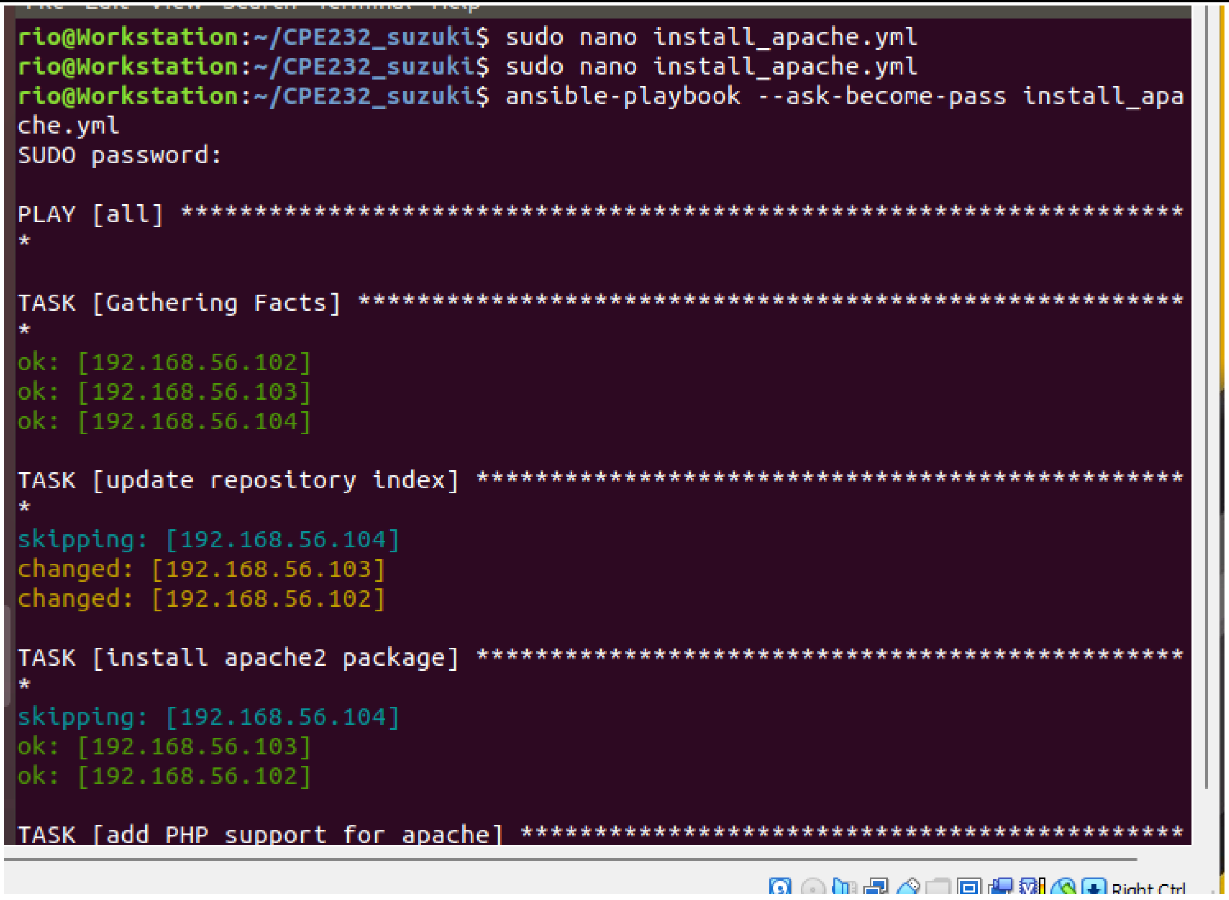
```
rio@Workstation:~/CPE232_suzuki$ install_apache.yml
install_apache.yml: command not found
rio@Workstation:~/CPE232_suzuki$ sudo nano install_apache.yml
[sudo] password for rio:
```



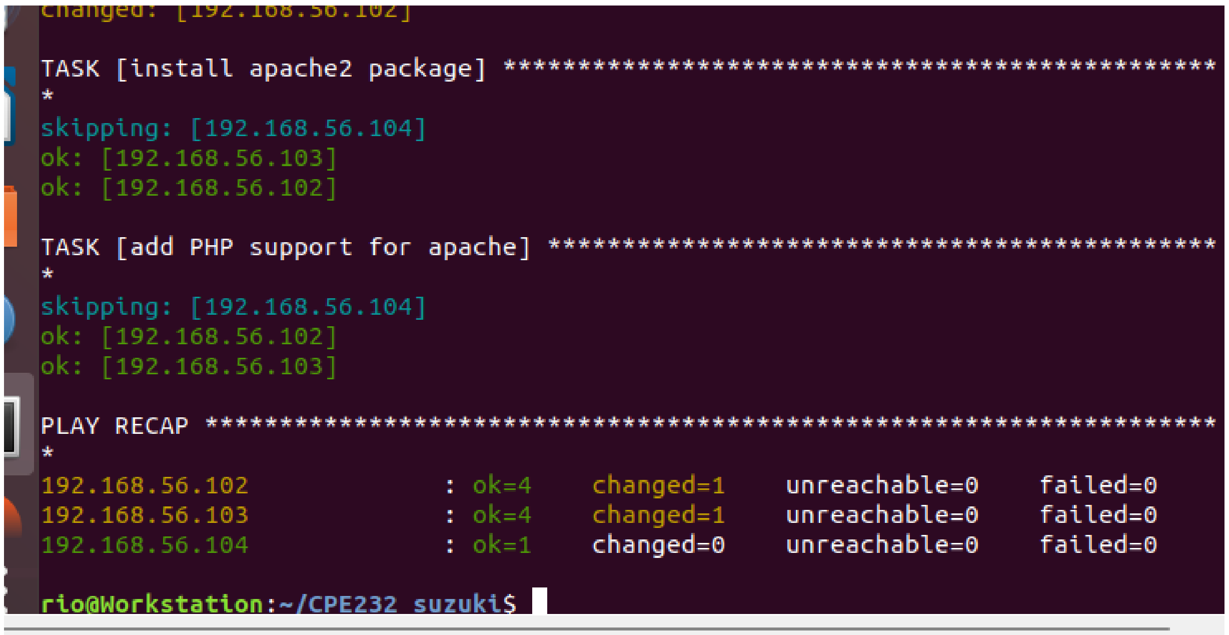
```
rio@Workstation: ~/CPE232_suzuki
File  Edit  View  Search  Terminal  Help
  GNU nano 2.9.3                          install_apache.yml

---
- 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"



                               [ Read 20 lines ]
^G Get Help     ^O Write Out    ^W Where Is     ^K Cut Text     ^J Justify
^X Exit         ^R Read File    ^\ Replace      ^U Uncut Text   ^T To Spell
```

Make sure to save the file and exit.

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



```
rio@Workstation:~/CPE232_suzuki$ sudo nano install_apache.yml
rio@Workstation:~/CPE232_suzuki$ sudo nano install_apache.yml
rio@Workstation:~/CPE232_suzuki$ ansible-playbook --ask-become-pass install_apa
che.yml
SUDO password:

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

TASK [Gathering Facts] *******************************************************
*
ok: [192.168.56.102]
ok: [192.168.56.103]
ok: [192.168.56.104]

TASK [update repository index] ***********************************************
*
skipping: [192.168.56.104]
changed: [192.168.56.103]
changed: [192.168.56.102]

TASK [install apache2 package] ***********************************************
*
skipping: [192.168.56.104]
ok: [192.168.56.103]
ok: [192.168.56.102]

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



```
changed: [192.168.56.102]

TASK [install apache2 package] ***********************************************
*
skipping: [192.168.56.104]
ok: [192.168.56.103]
ok: [192.168.56.102]

TASK [add PHP support for apache] ********************************************
*
skipping: [192.168.56.104]
ok: [192.168.56.102]
ok: [192.168.56.103]

PLAY RECAP *******************************************************************
*
192.168.56.102             : ok=4    changed=1    unreachable=0    failed=0
192.168.56.103             : ok=4    changed=1    unreachable=0    failed=0
192.168.56.104             : ok=1    changed=0    unreachable=0    failed=0

rio@Workstation:~/CPE232_suzuki$
```

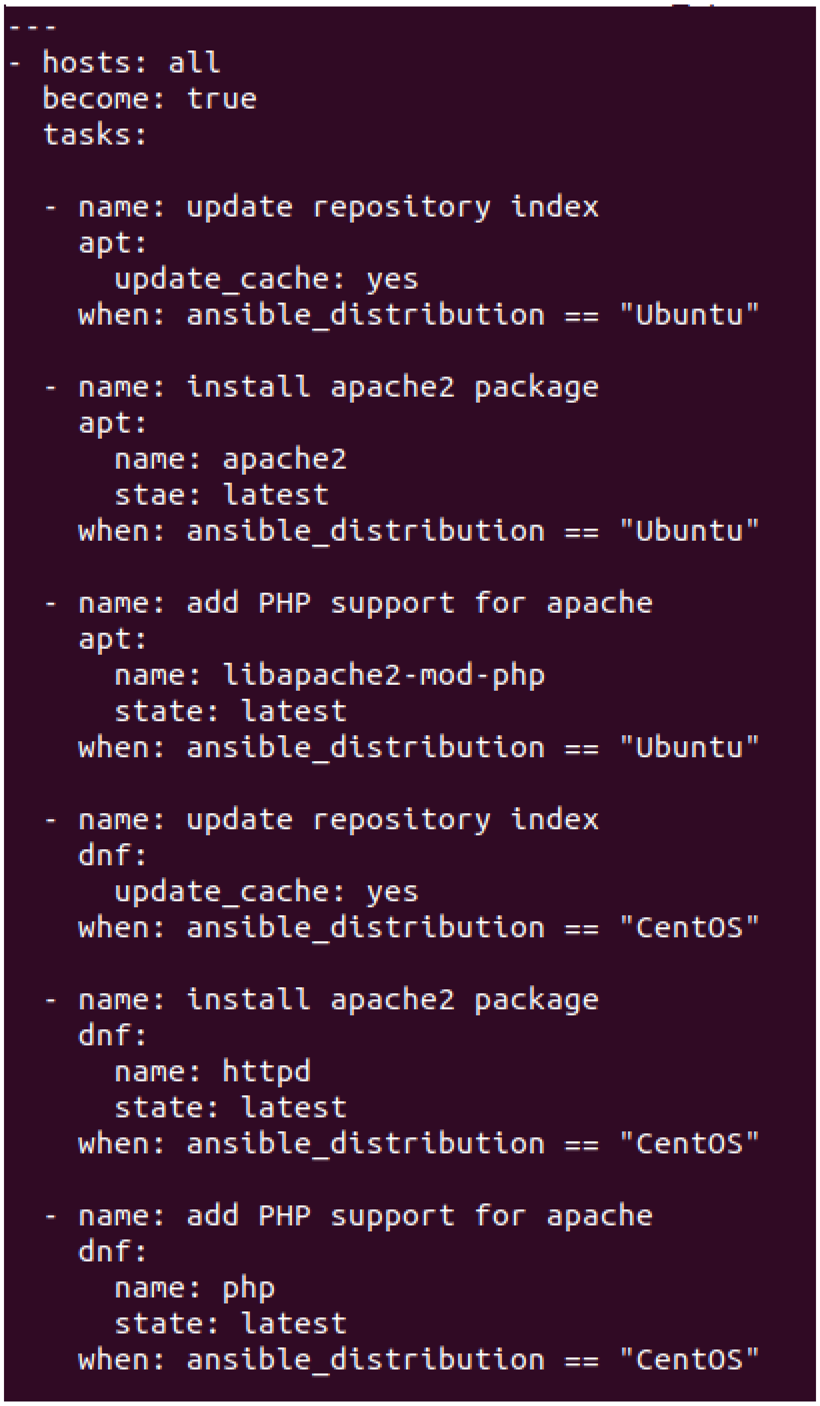
If you have a mix of Debian and Ubuntu servers, you can change the configuration of your playbook like this.

- name: update repository index

```
    apt:
      update_cache: yes
    when: ansible_distribution in [“Debian”, “Ubuntu]
```

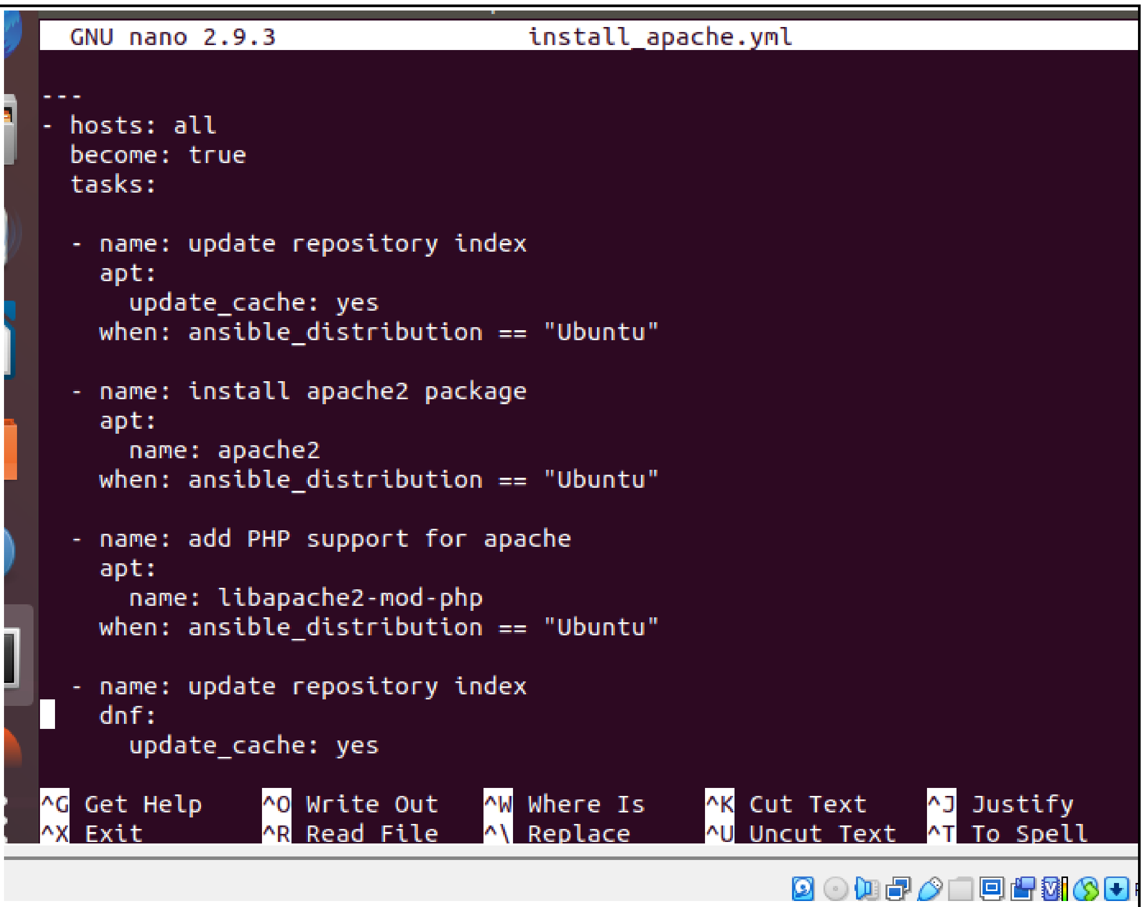
Note: This will work also if you try. Notice the changes are highlighted.

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



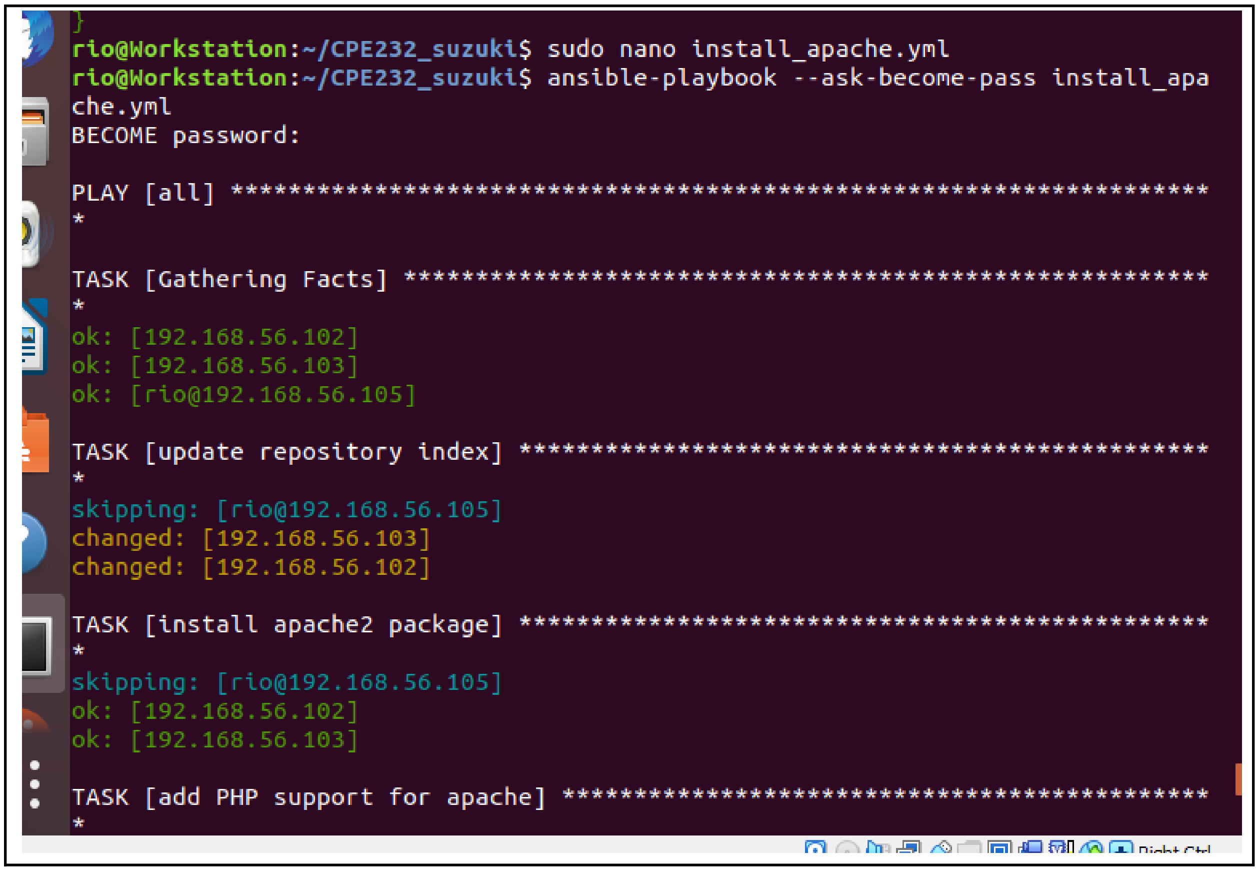
```
---
- hosts: all
  become: true
  tasks:

  - 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.

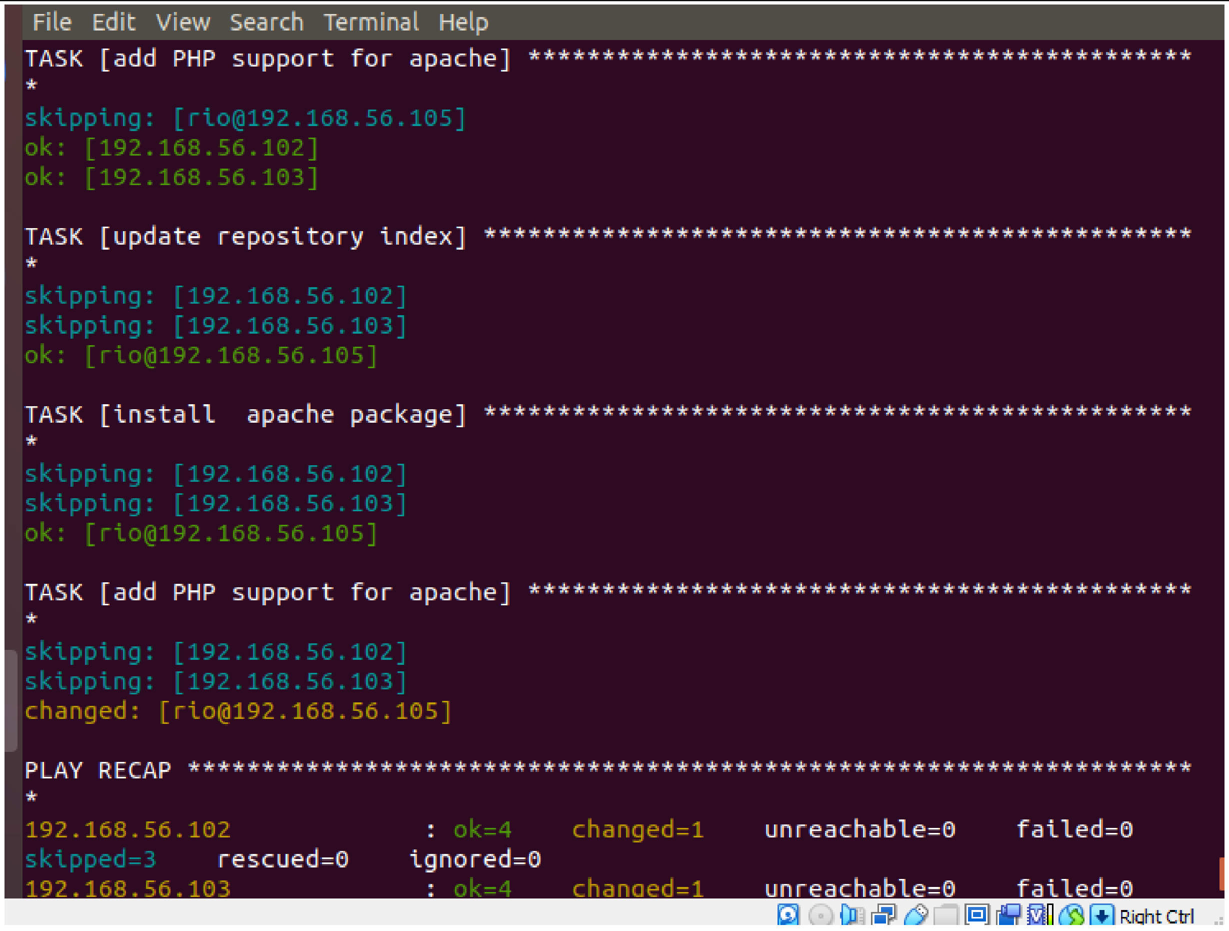


```
GNU nano 2.9.3                    install_apache.yml

---
- 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
    dnf:
      update_cache: yes

^G Get Help    ^O Write Out   ^W Where Is    ^K Cut Text    ^J Justify
^X Exit        ^R Read File   ^\ Replace     ^U Uncut Text  ^T To Spell
```

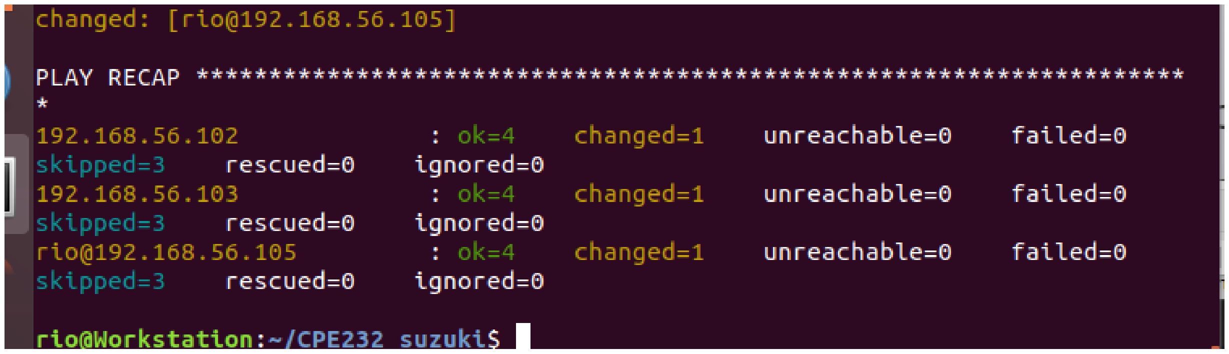
Run *ansible-playbook --ask-become-pass install_apache.yml* and describe the result.



```
}
rio@Workstation:~/CPE232_suzuki$ sudo nano install_apache.yml
rio@Workstation:~/CPE232_suzuki$ ansible-playbook --ask-become-pass install_apa
che.yml
BECOME password:

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

TASK [Gathering Facts] **********************************************************
*
ok: [192.168.56.102]
ok: [192.168.56.103]
ok: [rio@192.168.56.105]

TASK [update repository index] **************************************************
*
skipping: [rio@192.168.56.105]
changed: [192.168.56.103]
changed: [192.168.56.102]

TASK [install apache2 package] **************************************************
*
skipping: [rio@192.168.56.105]
ok: [192.168.56.102]
ok: [192.168.56.103]

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



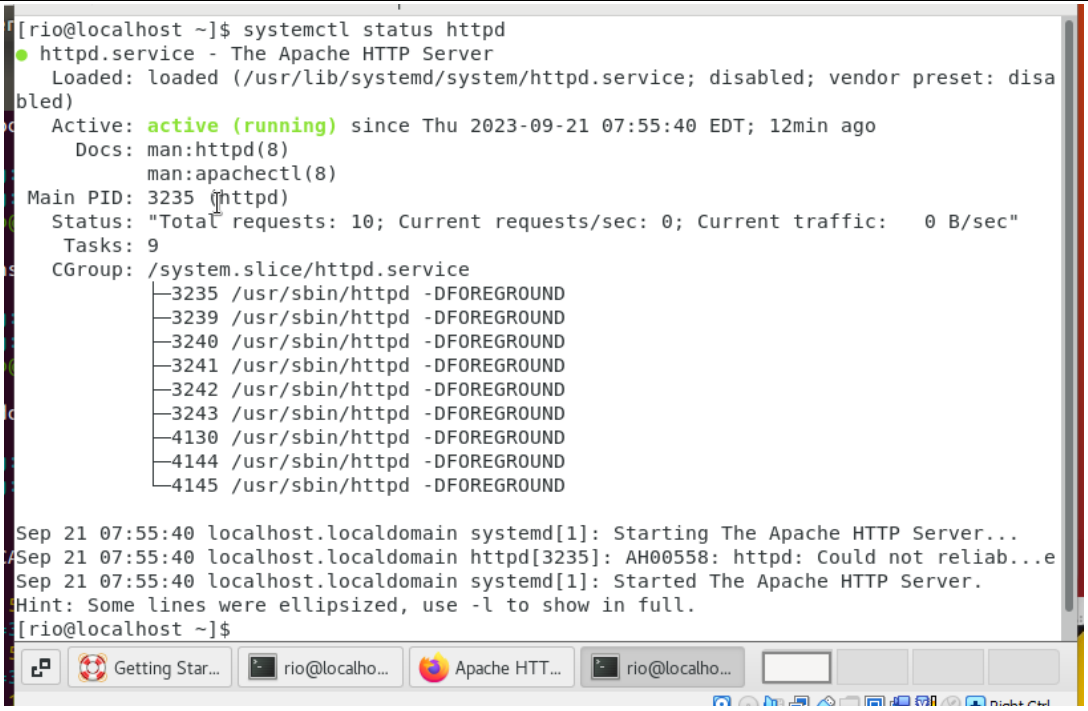
```
TASK [add PHP support for apache] *********************************************
*
skipping: [rio@192.168.56.105]
ok: [192.168.56.102]
ok: [192.168.56.103]

TASK [update repository index] ************************************************
*
skipping: [192.168.56.102]
skipping: [192.168.56.103]
ok: [rio@192.168.56.105]

TASK [install  apache package] ************************************************
*
skipping: [192.168.56.102]
skipping: [192.168.56.103]
ok: [rio@192.168.56.105]

TASK [add PHP support for apache] *********************************************
*
skipping: [192.168.56.102]
skipping: [192.168.56.103]
changed: [rio@192.168.56.105]

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



```
changed: [rio@192.168.56.105]

PLAY RECAP ********************************************************************
*
192.168.56.102             : ok=4    changed=1    unreachable=0    failed=0
skipped=3    rescued=0    ignored=0
192.168.56.103             : ok=4    changed=1    unreachable=0    failed=0
skipped=3    rescued=0    ignored=0
rio@192.168.56.105         : ok=4    changed=1    unreachable=0    failed=0
skipped=3    rescued=0    ignored=0

rio@Workstation:~/CPE232_suzuki$ 
```

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

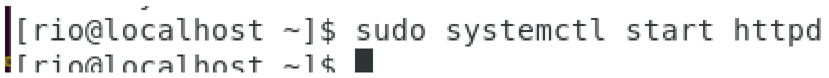


```
[rio@localhost ~]$ systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; vendor preset: disa
bled)
   Active: active (running) since Thu 2023-09-21 07:55:40 EDT; 12min ago
     Docs: man:httpd(8)
           man:apachectl(8)
 Main PID: 3235 (httpd)
   Status: "Total requests: 10; Current requests/sec: 0; Current traffic:   0 B/sec"
    Tasks: 9
   CGroup: /system.slice/httpd.service
           ├─3235 /usr/sbin/httpd -DFOREGROUND
           ├─3239 /usr/sbin/httpd -DFOREGROUND
           ├─3240 /usr/sbin/httpd -DFOREGROUND
           ├─3241 /usr/sbin/httpd -DFOREGROUND
           ├─3242 /usr/sbin/httpd -DFOREGROUND
           ├─3243 /usr/sbin/httpd -DFOREGROUND
           ├─4130 /usr/sbin/httpd -DFOREGROUND
           ├─4144 /usr/sbin/httpd -DFOREGROUND
           └─4145 /usr/sbin/httpd -DFOREGROUND

Sep 21 07:55:40 localhost.localdomain systemd[1]: Starting The Apache HTTP Server...
Sep 21 07:55:40 localhost.localdomain httpd[3235]: AH00558: httpd: Could not reliab...e
Sep 21 07:55:40 localhost.localdomain systemd[1]: Started The Apache HTTP Server.
Hint: Some lines were ellipsized, use -l to show in full.
[rio@localhost ~]$
```

The result of this command tells you that the service is inactive.

5.2 Issue the following command to start the service:

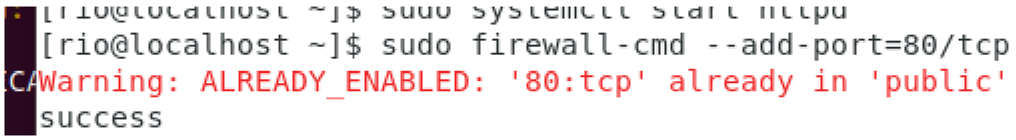
sudo systemctl start httpd



```
[rio@localhost ~]$ sudo systemctl start httpd
[rio@localhost ~]$
```

(When prompted, enter the sudo password)

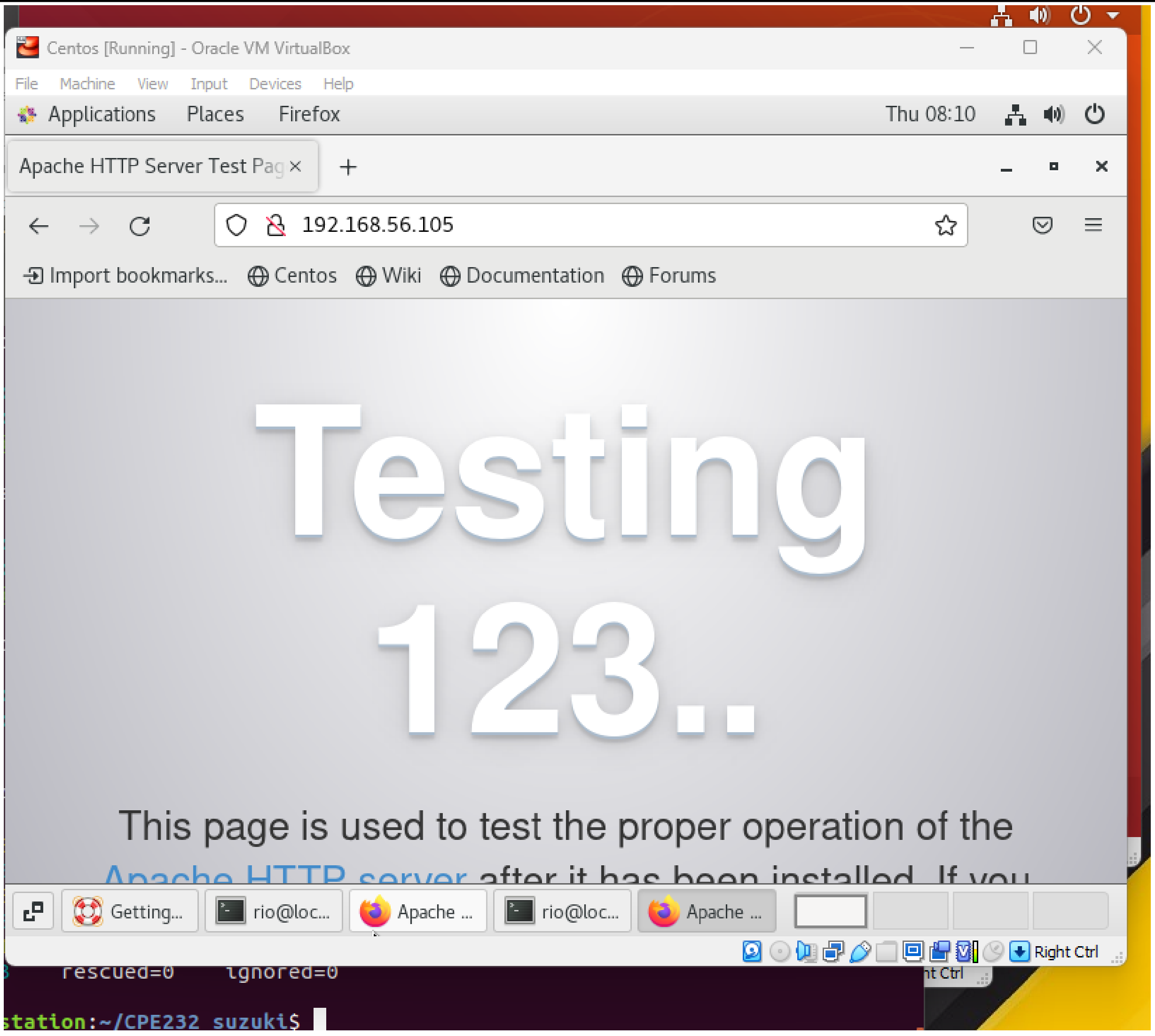
sudo firewall-cmd --add-port=80/tcp



```
[rio@localhost ~]$ sudo firewall-cmd --add-port=80/tcp
Warning: ALREADY_ENABLED: '80:tcp' already in 'public'
success
```

(The result should be a 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)


Centos [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places Firefox
Thu 08:10
Apache HTTP Server Test Pag
192.168.56.105
Import bookmarks... Centos Wiki Documentation Forums
Getting... rio@loc... Apache ... rio@loc... Apache ...
Right Ctrl

Testing 123..

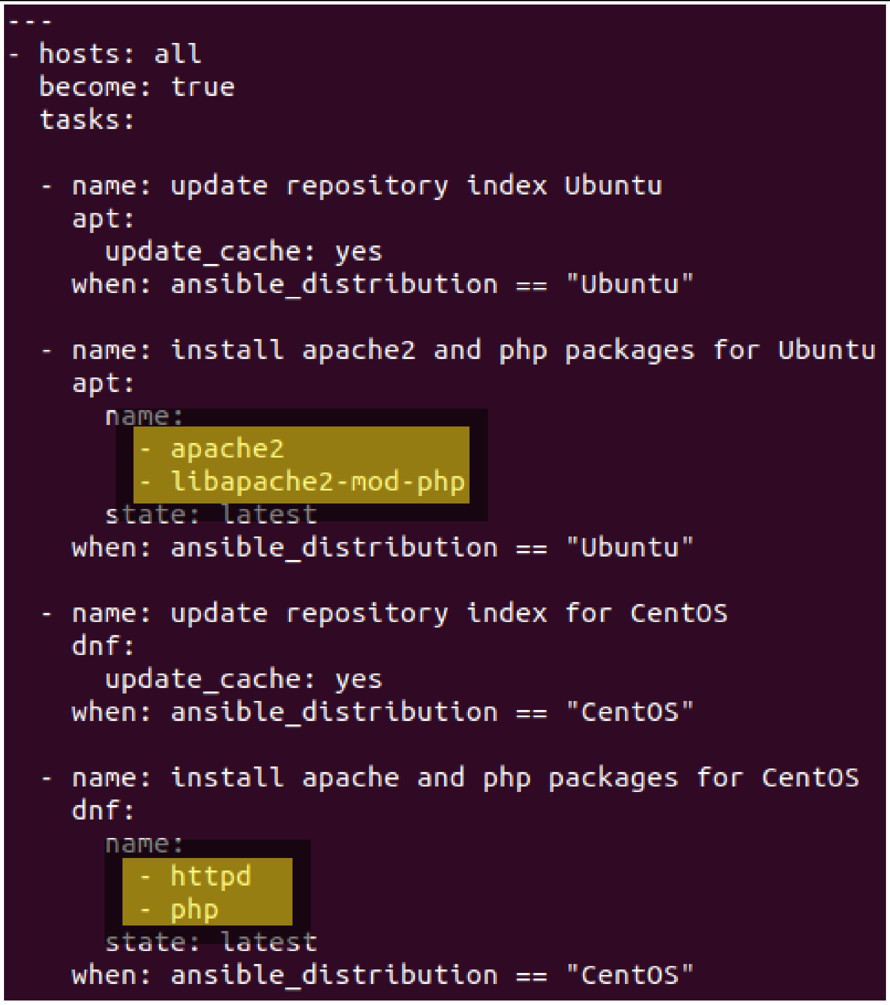
This page is used to test the proper operation of the Apache HTTP server after it has been installed. If you

```
rescued=0    ignored=0
station:~/CPE232 suzuki$
```

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.

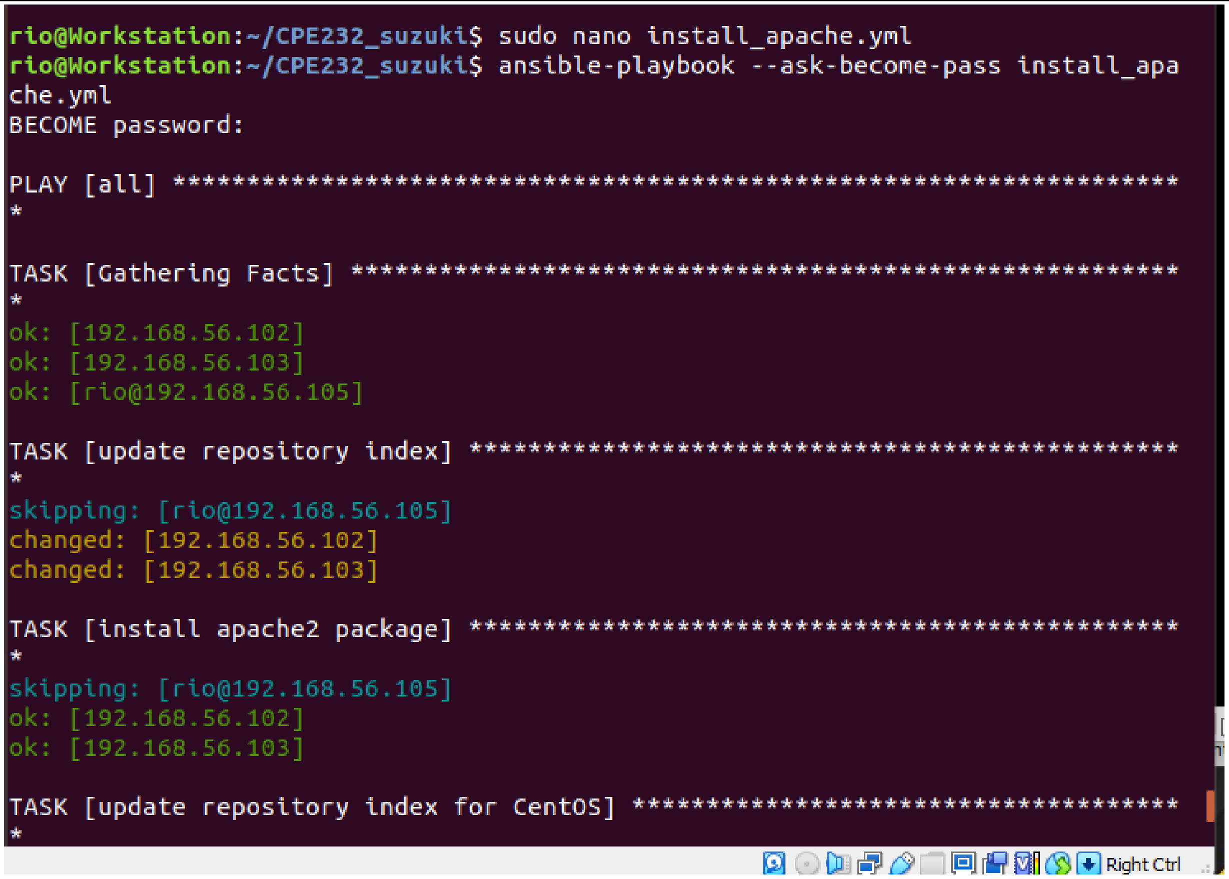
1. Edit the playbook *install_apache.yml*. Currently, we have three tasks targeting our Ubuntu machines and 3 tasks targeting our CentOS machine. Right now, we try to consolidate some tasks that are typically the same. For example, we can consolidate two plays that install packages. We can do that by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



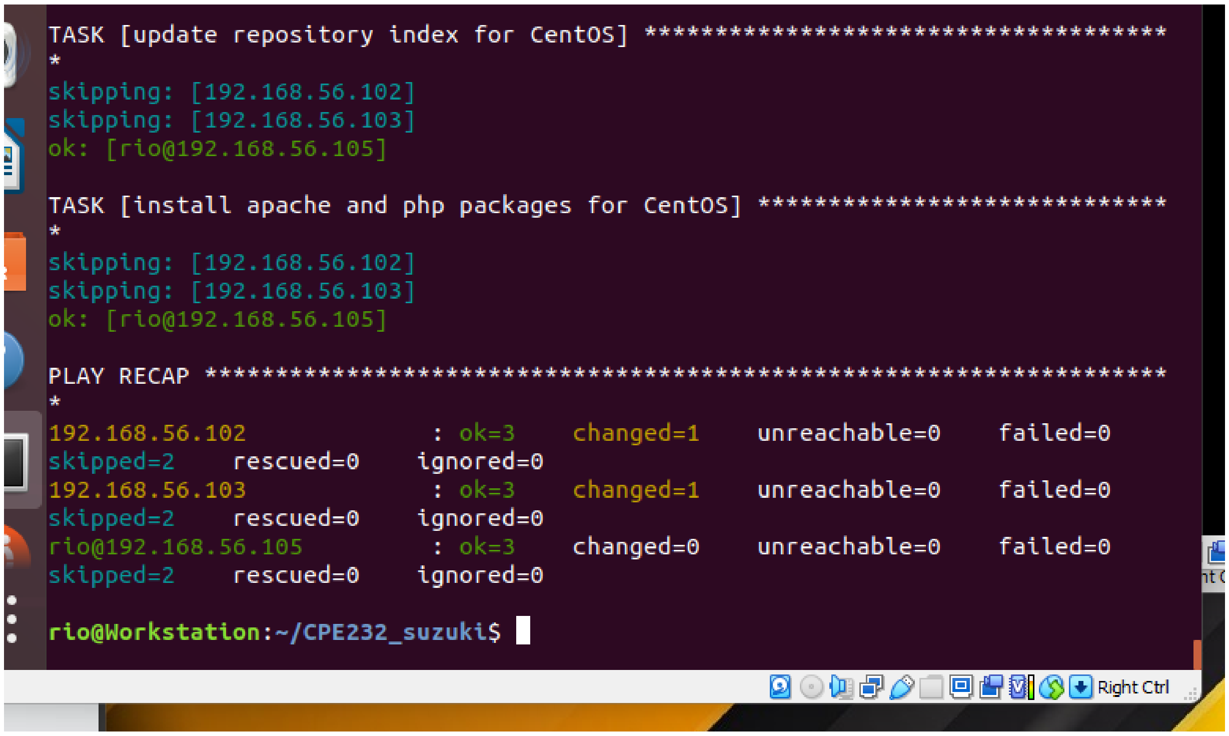
```
---
- hosts: all
  become: true
  tasks:

  - 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.

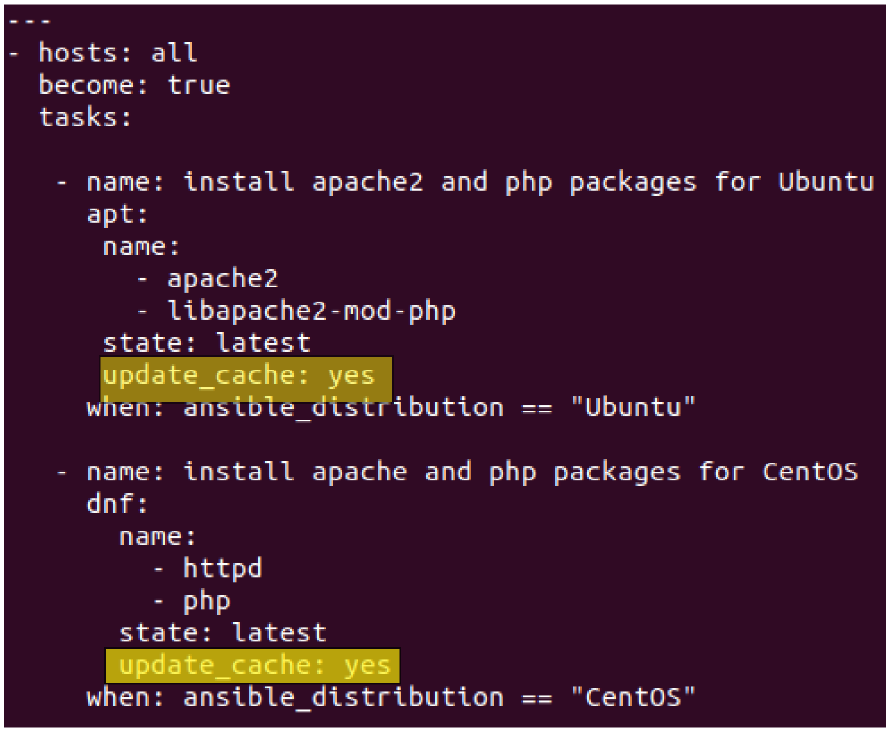


```
rio@Workstation:~/CPE232_suzuki$ sudo nano install_apache.yml
rio@Workstation:~/CPE232_suzuki$ ansible-playbook --ask-become-pass install_apa
che.yml
BECOME password:

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

TASK [Gathering Facts] *********************************************************
*
ok: [192.168.56.102]
ok: [192.168.56.103]
ok: [rio@192.168.56.105]

TASK [update repository index] *************************************************
*
skipping: [rio@192.168.56.105]
changed: [192.168.56.102]
changed: [192.168.56.103]

TASK [install apache2 package] *************************************************
*
skipping: [rio@192.168.56.105]
ok: [192.168.56.102]
ok: [192.168.56.103]

TASK [update repository index for CentOS] **************************************
*
```



```
TASK [update repository index for CentOS] **************************************
*
skipping: [192.168.56.102]
skipping: [192.168.56.103]
ok: [rio@192.168.56.105]

TASK [install apache and php packages for CentOS] ******************************
*
skipping: [192.168.56.102]
skipping: [192.168.56.103]
ok: [rio@192.168.56.105]

PLAY RECAP *********************************************************************
*
192.168.56.102             : ok=3    changed=1    unreachable=0    failed=0
skipped=2    rescued=0    ignored=0
192.168.56.103             : ok=3    changed=1    unreachable=0    failed=0
skipped=2    rescued=0    ignored=0
rio@192.168.56.105         : ok=3    changed=0    unreachable=0    failed=0
skipped=2    rescued=0    ignored=0

rio@Workstation:~/CPE232_suzuki$
```

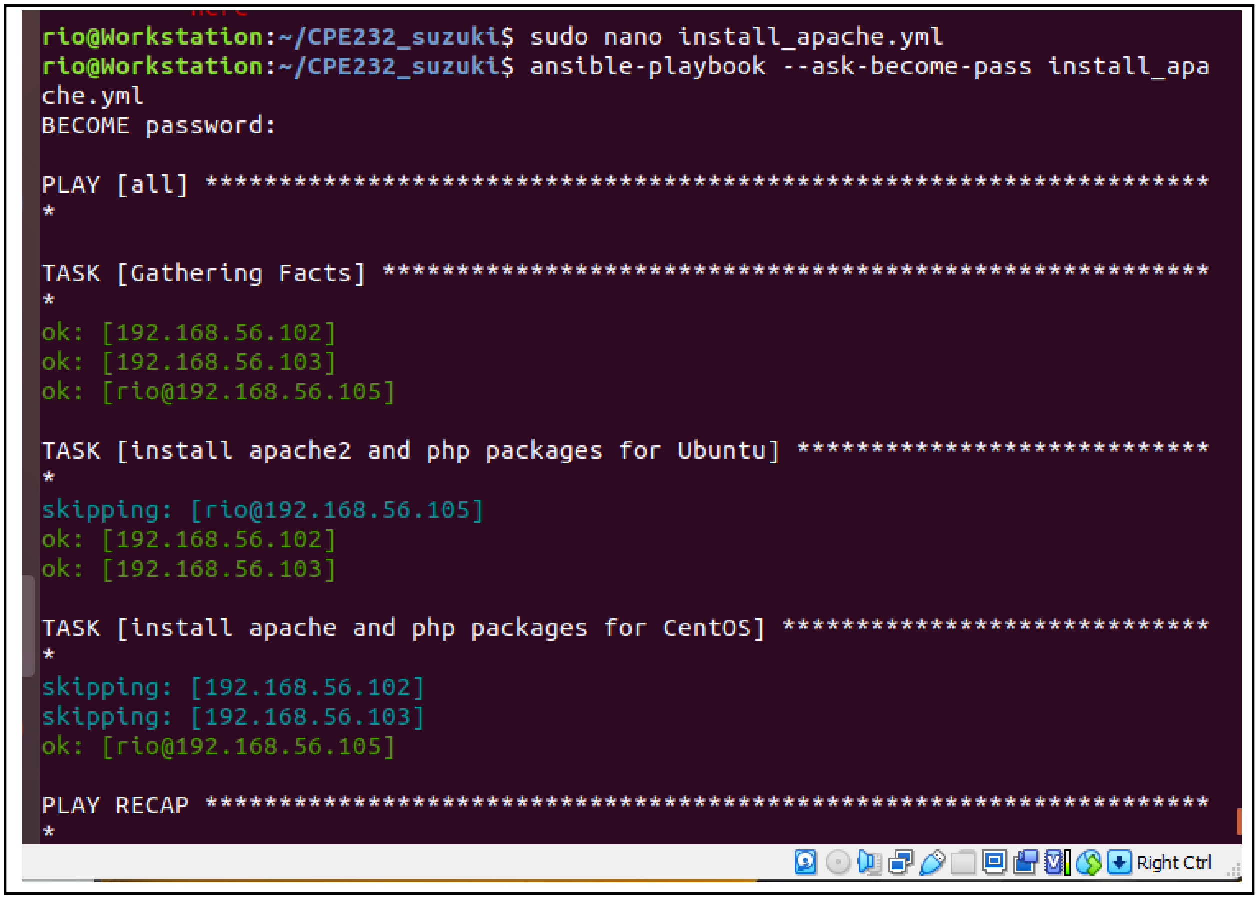
2. Edit the playbook *install_apache.yml* again. In task 2.1, we consolidated the plays into one play. This time we can actually consolidated everything in just 2 plays. This can be done by removing the update repository play and putting the command *update_cache: yes* below the command *state: latest*.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



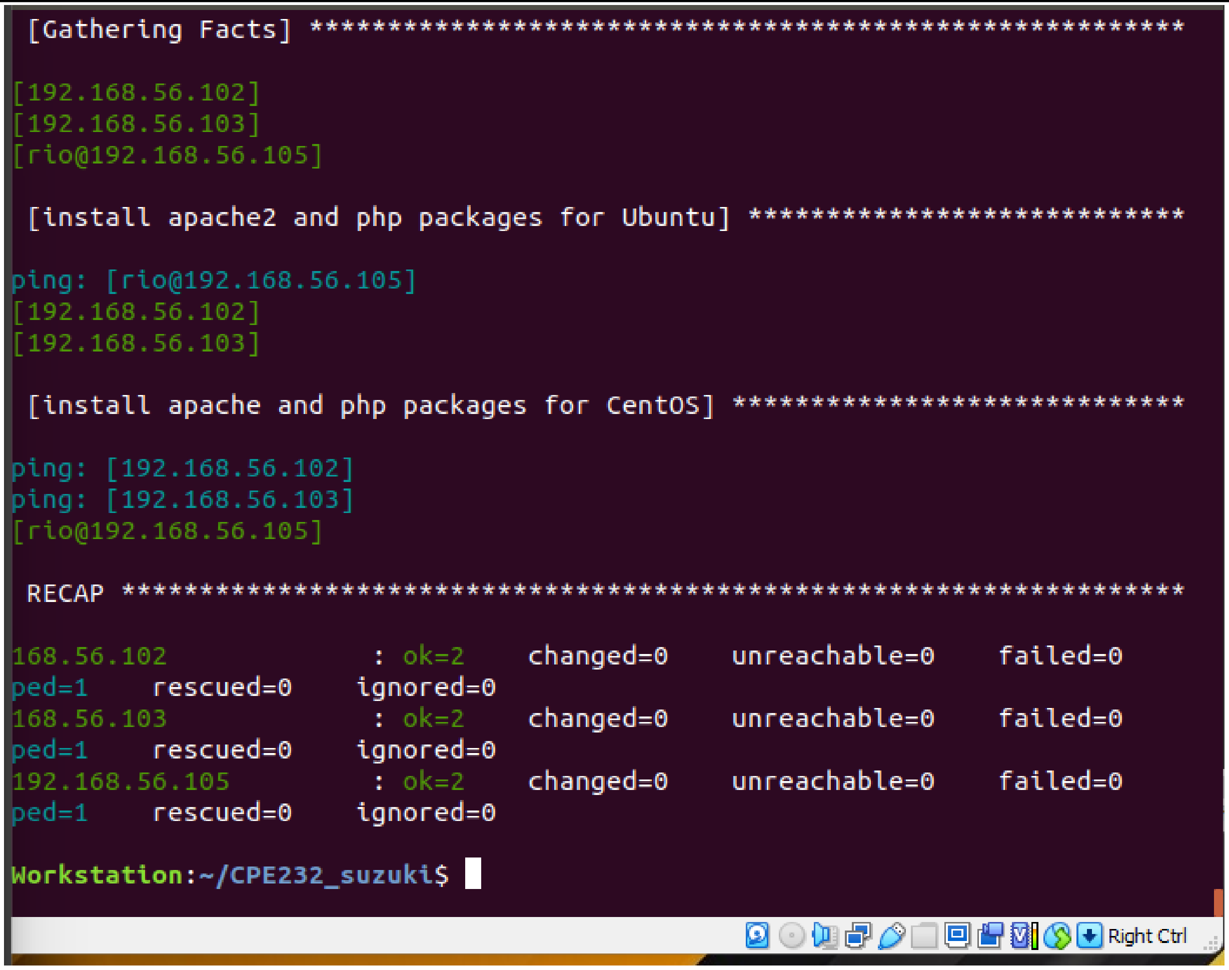
```
---
- hosts: all
  become: true
  tasks:

   - 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.

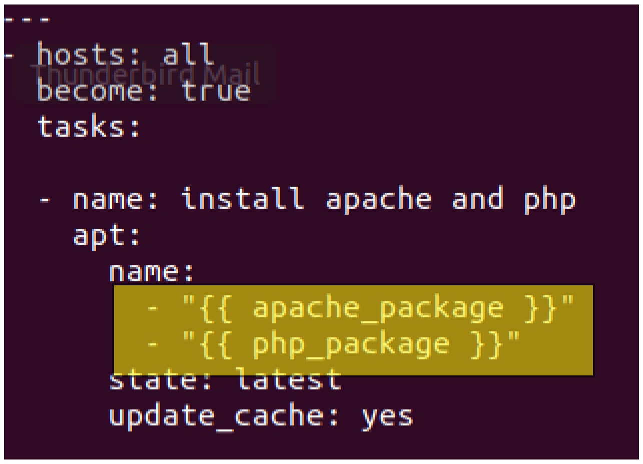


```
rio@Workstation:~/CPE232_suzuki$ sudo nano install_apache.yml
rio@Workstation:~/CPE232_suzuki$ ansible-playbook --ask-become-pass install_apa
che.yml
BECOME password:

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

TASK [Gathering Facts] *********************************************************
*
ok: [192.168.56.102]
ok: [192.168.56.103]
ok: [rio@192.168.56.105]

TASK [install apache2 and php packages for Ubuntu] *****************************
*
skipping: [rio@192.168.56.105]
ok: [192.168.56.102]
ok: [192.168.56.103]

TASK [install apache and php packages for CentOS] ******************************
*
skipping: [192.168.56.102]
skipping: [192.168.56.103]
ok: [rio@192.168.56.105]

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



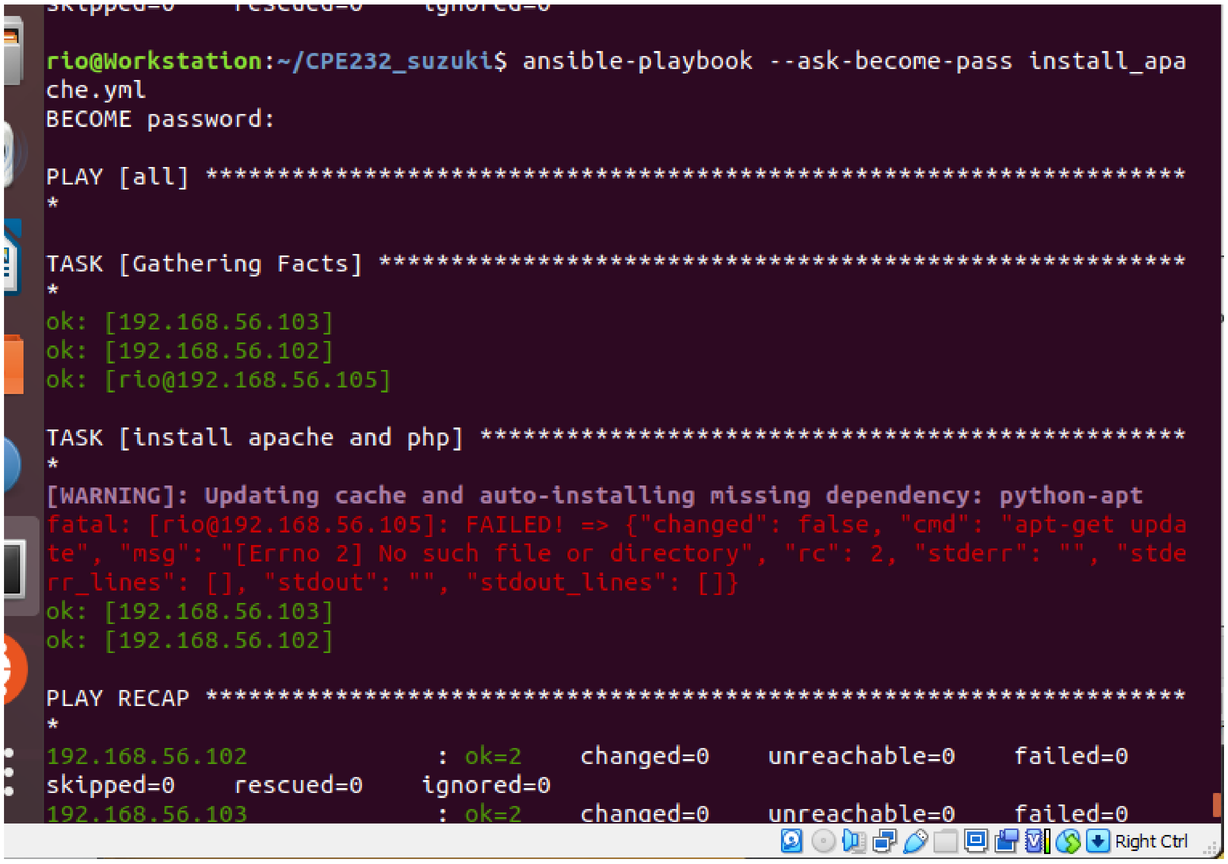
```
 [Gathering Facts] *********************************************************

[192.168.56.102]
[192.168.56.103]
[rio@192.168.56.105]

 [install apache2 and php packages for Ubuntu] *****************************

ping: [rio@192.168.56.105]
[192.168.56.102]
[192.168.56.103]

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

ping: [192.168.56.102]
ping: [192.168.56.103]
[rio@192.168.56.105]

 RECAP *********************************************************************

168.56.102             : ok=2    changed=0    unreachable=0    failed=0
ped=1    rescued=0    ignored=0
168.56.103             : ok=2    changed=0    unreachable=0    failed=0
ped=1    rescued=0    ignored=0
192.168.56.105         : ok=2    changed=0    unreachable=0    failed=0
ped=1    rescued=0    ignored=0

Workstation:~/CPE232_suzuki$ 
```

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.

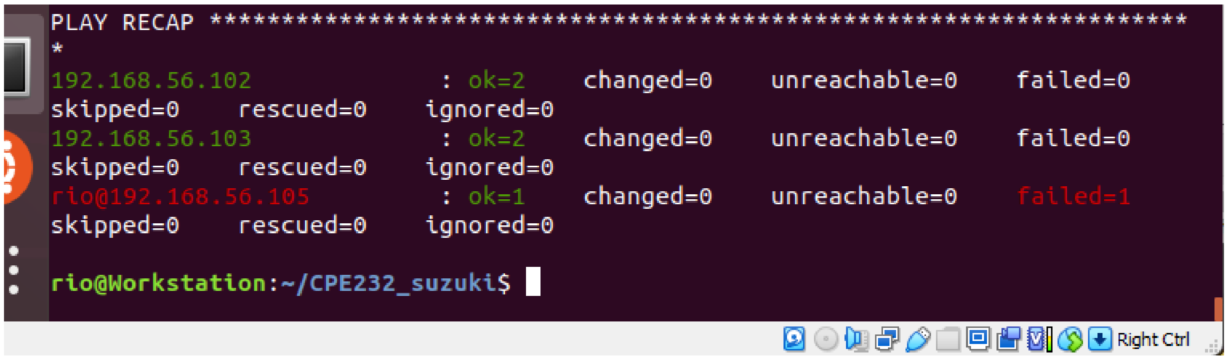


```
---
- hosts: all
  become: true
  tasks:

  - 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.



```
skipped=0     rescued=0     ignored=0

rio@Workstation:~/CPE232_suzuki$ ansible-playbook --ask-become-pass install_apa
che.yml
BECOME password:

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

TASK [Gathering Facts] *********************************************************
*
ok: [192.168.56.103]
ok: [192.168.56.102]
ok: [rio@192.168.56.105]

TASK [install apache and php] **************************************************
*
[WARNING]: Updating cache and auto-installing missing dependency: python-apt
fatal: [rio@192.168.56.105]: FAILED! => {"changed": false, "cmd": "apt-get upda
te", "msg": "[Errno 2] No such file or directory", "rc": 2, "stderr": "", "stde
rr_lines": [], "stdout": "", "stdout_lines": []}
ok: [192.168.56.103]
ok: [192.168.56.102]

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

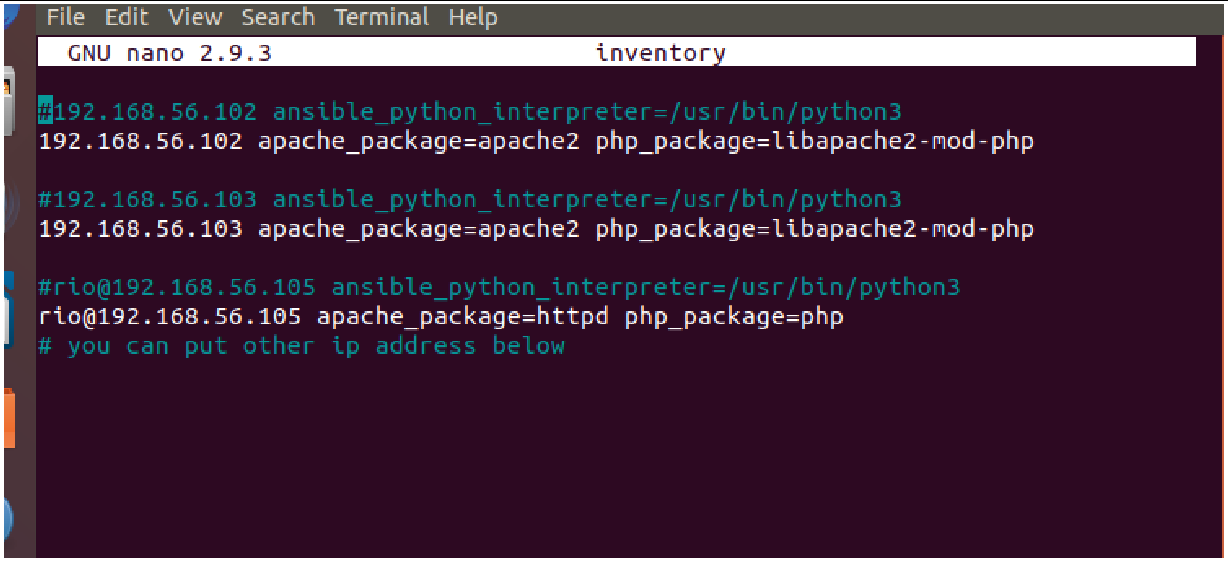


```
PLAY RECAP *********************************************************************
*
192.168.56.102             : ok=2    changed=0    unreachable=0    failed=0
skipped=0     rescued=0     ignored=0
192.168.56.103             : ok=2    changed=0    unreachable=0    failed=0
skipped=0     rescued=0     ignored=0
rio@192.168.56.105         : ok=1    changed=0    unreachable=0    failed=1
skipped=0     rescued=0     ignored=0

rio@Workstation:~/CPE232_suzuki$ 
```

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:

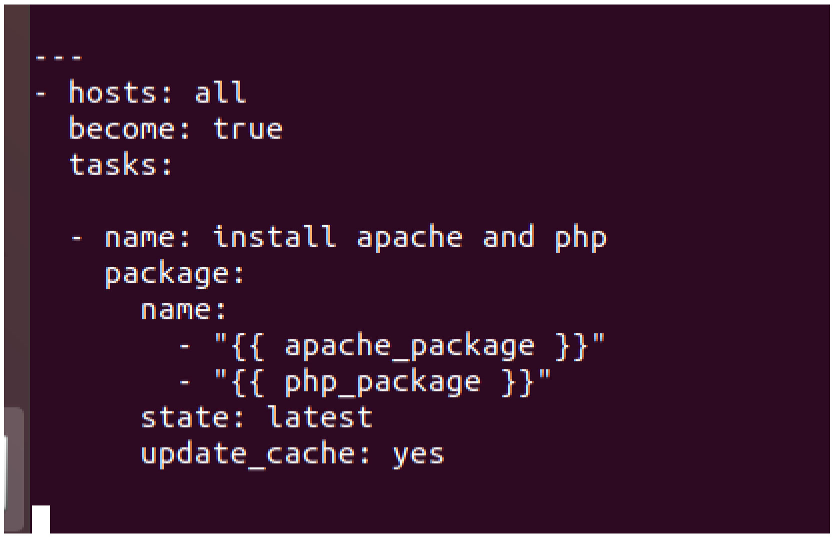
```
192.168.56.120 apache_package=apache2 php_package=libapache2-mod-php
192.168.56.121 apache_package=apache2 php_package=libapache2-mod-php
192.168.56.122 apache_package=httpd php_package=php
```

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



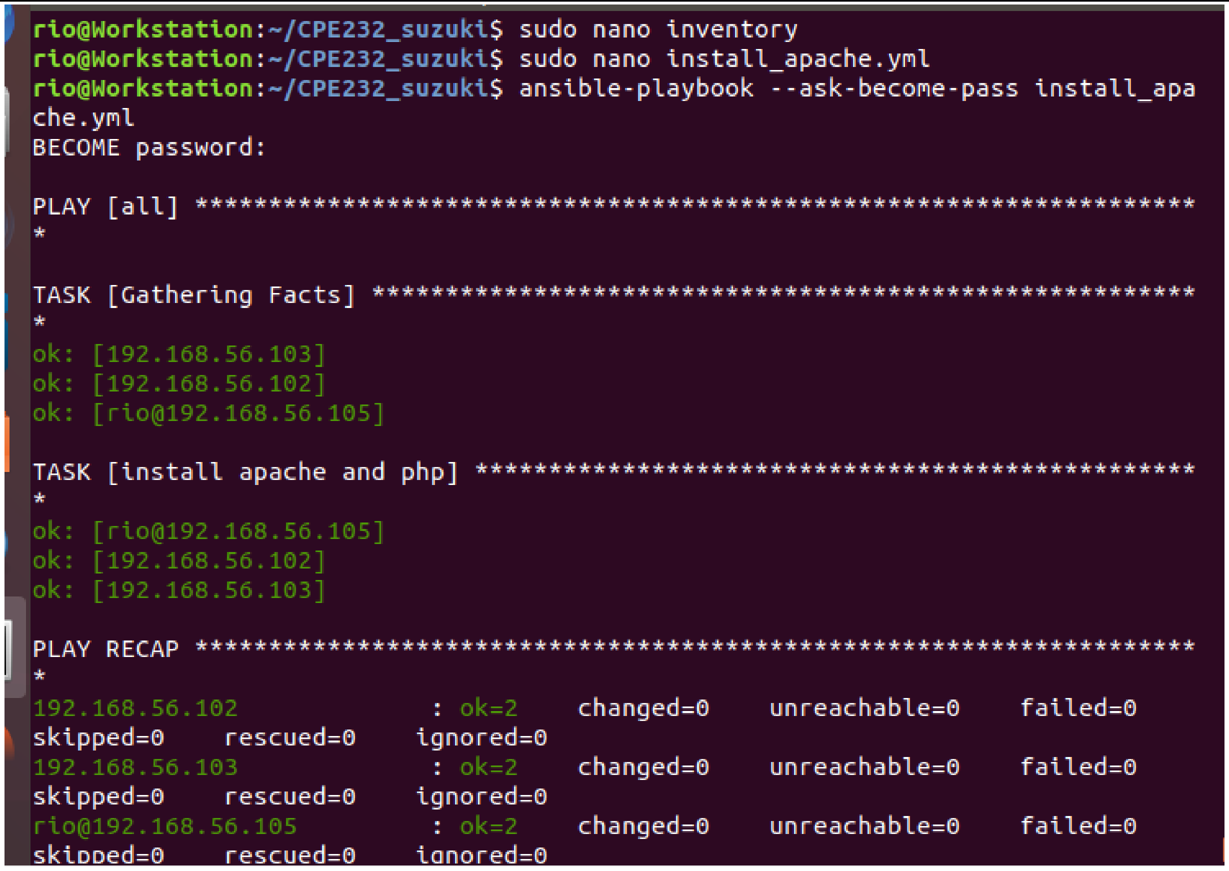
```
  GNU nano 2.9.3                              inventory

#192.168.56.102 ansible_python_interpreter=/usr/bin/python3
192.168.56.102 apache_package=apache2 php_package=libapache2-mod-php

#192.168.56.103 ansible_python_interpreter=/usr/bin/python3
192.168.56.103 apache_package=apache2 php_package=libapache2-mod-php

#rio@192.168.56.105 ansible_python_interpreter=/usr/bin/python3
rio@192.168.56.105 apache_package=httpd php_package=php
# you can put other ip address below
```

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



```
---
- 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.



```
rio@Workstation:~/CPE232_suzuki$ sudo nano inventory
rio@Workstation:~/CPE232_suzuki$ sudo nano install_apache.yml
rio@Workstation:~/CPE232_suzuki$ ansible-playbook --ask-become-pass install_apa
che.yml
BECOME password:

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

TASK [Gathering Facts] *********************************************************
*
ok: [192.168.56.103]
ok: [192.168.56.102]
ok: [rio@192.168.56.105]

TASK [install apache and php] **************************************************
*
ok: [rio@192.168.56.105]
ok: [192.168.56.102]
ok: [192.168.56.103]

PLAY RECAP *********************************************************************
*
192.168.56.102             : ok=2    changed=0    unreachable=0    failed=0
skipped=0    rescued=0    ignored=0
192.168.56.103             : ok=2    changed=0    unreachable=0    failed=0
skipped=0    rescued=0    ignored=0
rio@192.168.56.105         : 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.

Reflections:

Answer the following:

1. Why do you think refactoring of playbook codes is important?
   - refactoring playbook code is important because it makes it more easier to maintain for automating tasks and it will make sure that the codes are up to date.
2. When do we use the "when" command in playbook?
   - We use the when command in playbooks when we have a conditional based like to execute a task only on certain operating systems, like running a task that's specific to CentOS only.

Conclusion:

To conclude, this activity aims for the student to have better knowledge and skills in using the when command. I encountered a lot of problems and one of this is the missing dnf file. I was able to solve it by re-installing CentOS. The when command is a very important command in ansible since it enables you to apply conditional logic to your tasks and roles, allowing you to tailor your automation to specific scenarios and conditions.