Name: Darroca, Josiah Miguel B.	**Date Performed: 18/09/2023**
Course/Section: CPE31S4	**Date Submitted: 18/09/2023**
Instructor: Engr. Jonathan Taylar	**Semester and SY: 1st / 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**:

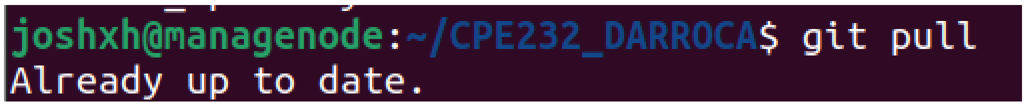
We are going to look at a way that we can differentiate a playbook by a host in terms of which distribution the host is running. It's very common in most Linux shops to run multiple distributions, for example, Ubuntu shop or Debian shop and you need a different distribution for a one off-case or perhaps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

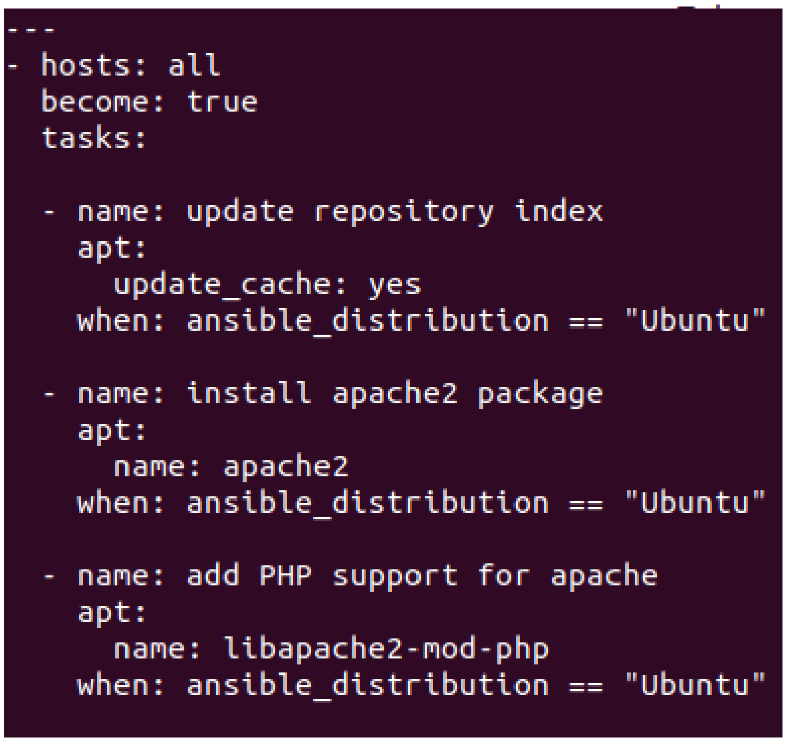
Task 1: Use when command for different distributions

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?



```
joshxh@managenode:~/CPE232_DARROCA$ git pull
Already up to date.
```

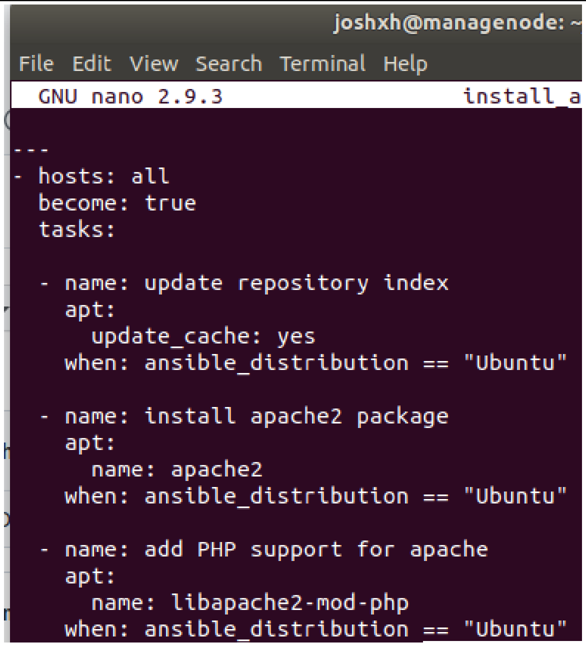
2. Edit the inventory file and add the IP address of the Centos VM. Issue the command we used to execute the playbook (the one we used in the last activity): *ansible-playbook --ask-become-pass install_apache.yml*. After executing this command, you may notice that it did not become successful in the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

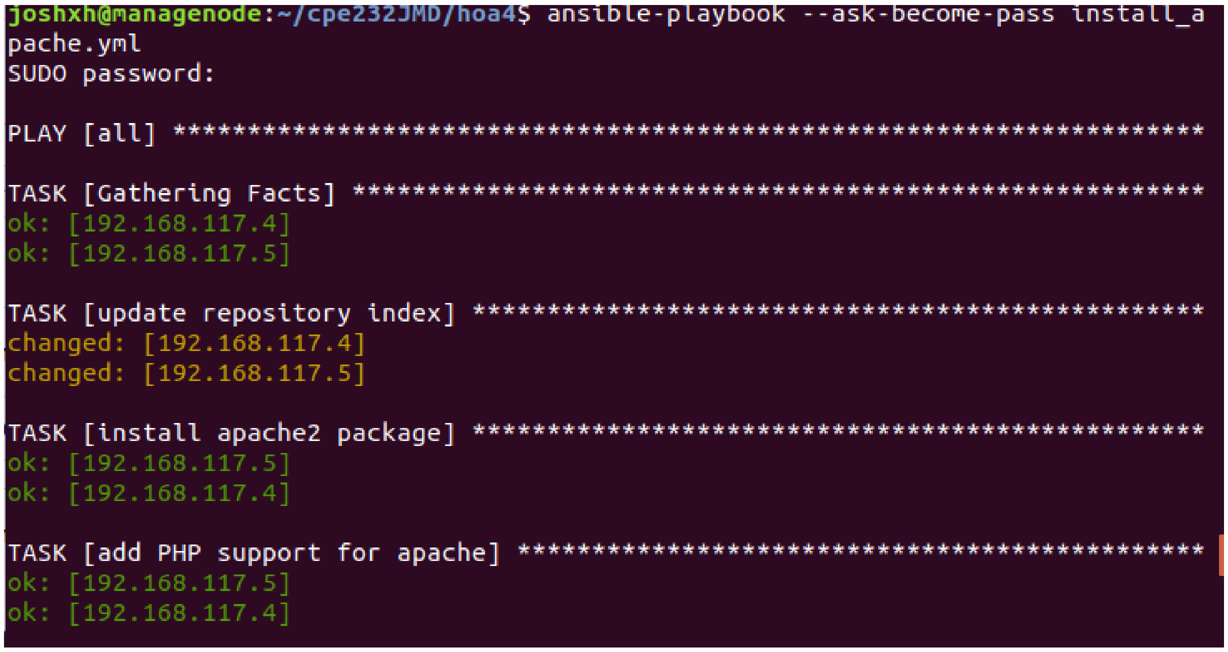
Input



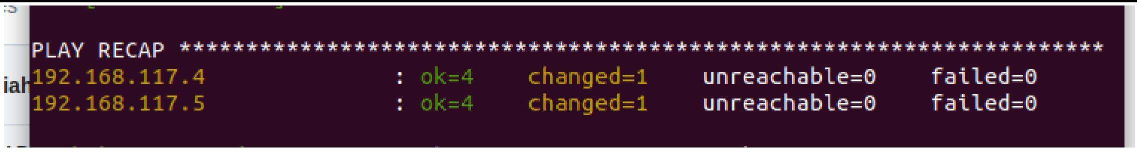
```
  GNU nano 2.9.3                    install_a

---
- 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.

Process

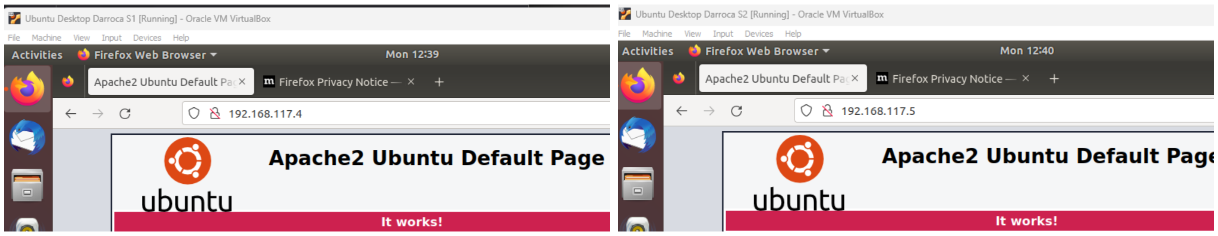


```
joshxh@managenode:~/cpe232JMD/hoa4$ ansible-playbook --ask-become-pass install_a
pache.yml
SUDO password:

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

TASK [Gathering Facts] *********************************************************
ok: [192.168.117.4]
ok: [192.168.117.5]

TASK [update repository index] *************************************************
changed: [192.168.117.4]
changed: [192.168.117.5]

TASK [install apache2 package] *************************************************
ok: [192.168.117.5]
ok: [192.168.117.4]

TASK [add PHP support for apache] **********************************************
ok: [192.168.117.5]
ok: [192.168.117.4]
```



```
PLAY RECAP ************************************************************************
192.168.117.4              : ok=4    changed=1    unreachable=0    failed=0
192.168.117.5              : ok=4    changed=1    unreachable=0    failed=0
```

Output

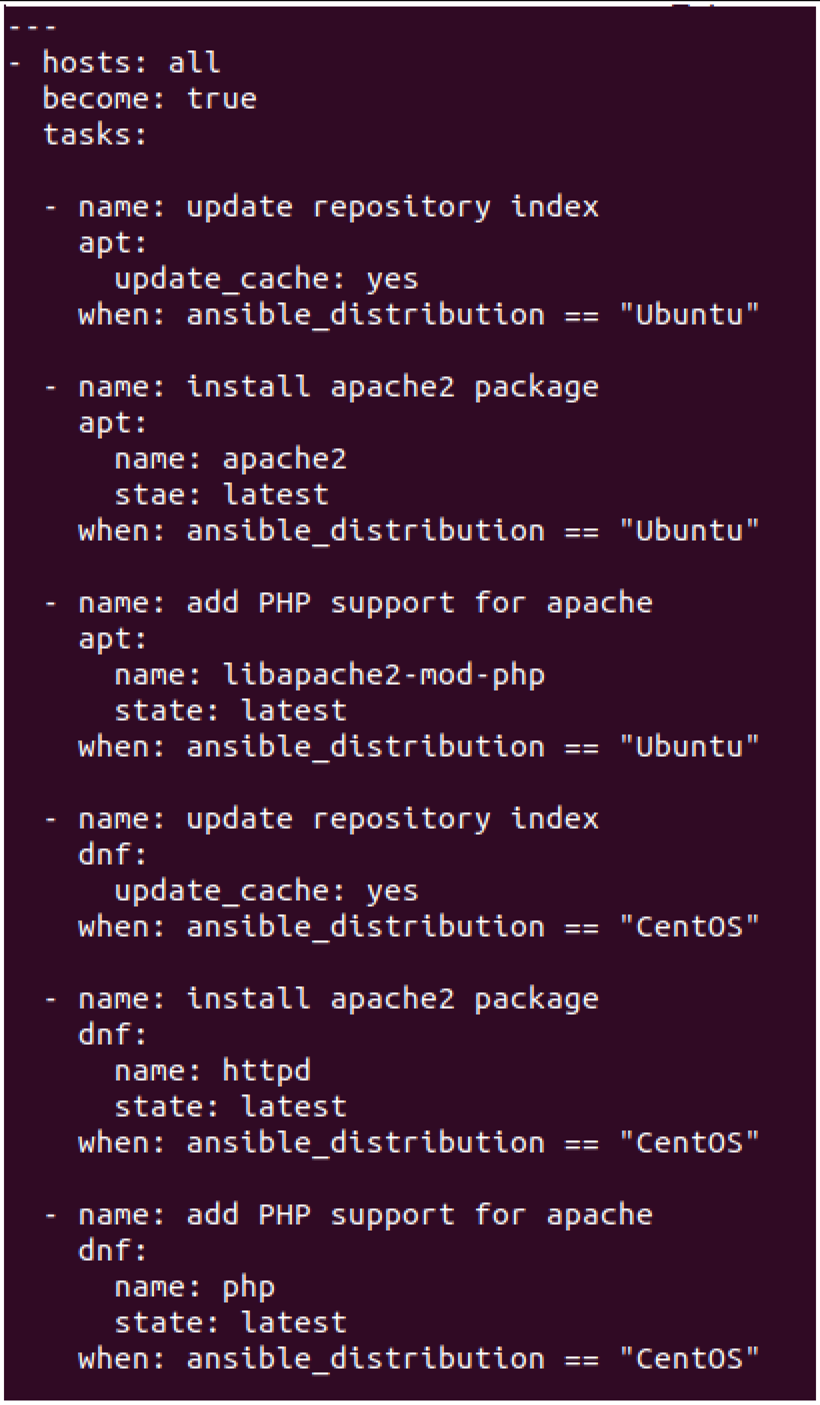

Ubuntu Desktop Darroca S1 [Running] - Oracle VM VirtualBox
Activities
Firefox Web Browser
Mon 12:39
Apache2 Ubuntu Default Pa
Firefox Privacy Notice
192.168.117.4
Apache2 Ubuntu Default Page
ubuntu
It works!
Ubuntu Desktop Darroca S2 [Running] - Oracle VM VirtualBox
Activities
Firefox Web Browser
Mon 12:40
Apache2 Ubuntu Default Pa
Firefox Privacy Notice
192.168.117.5
Apache2 Ubuntu Default Page
ubuntu
It works!

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.

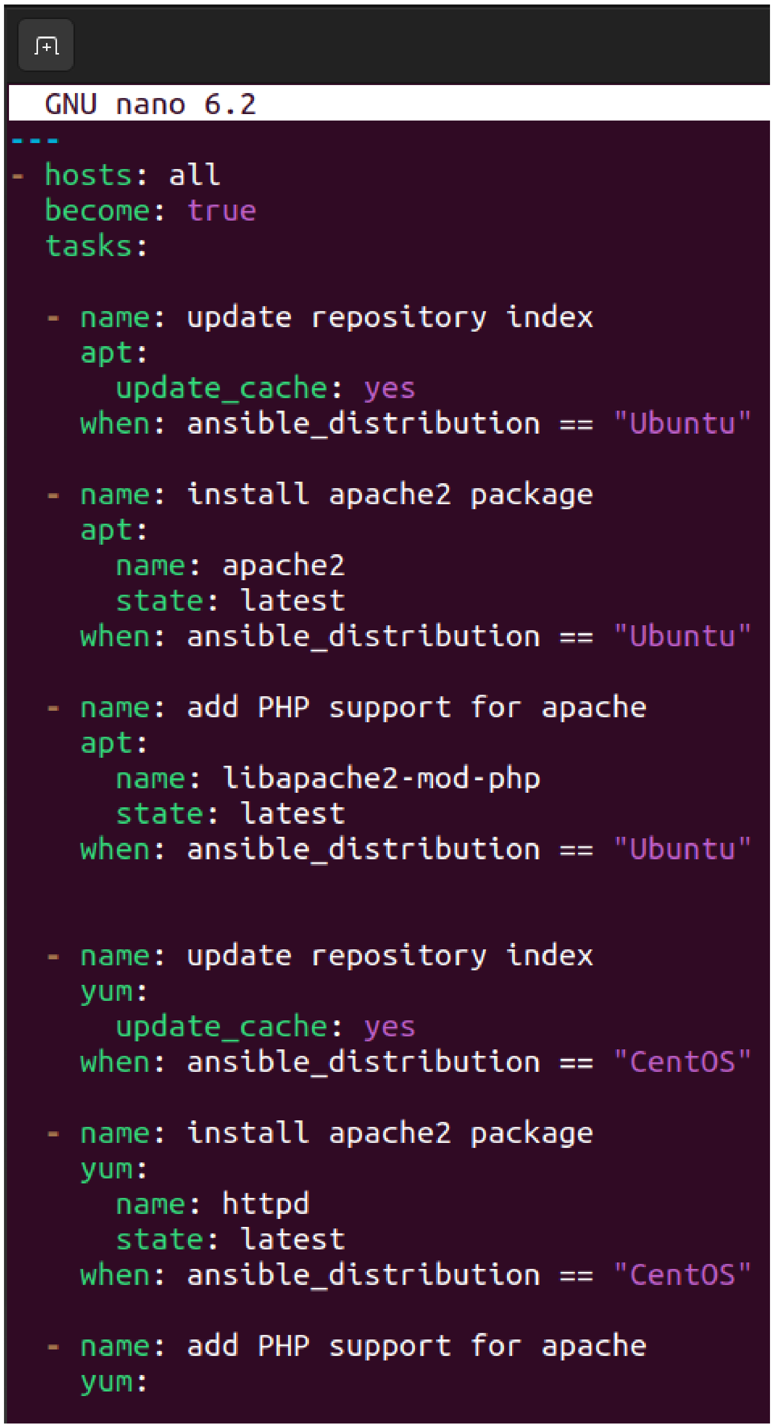
4. Edit the *install_apache.yml* file and insert the lines shown below.



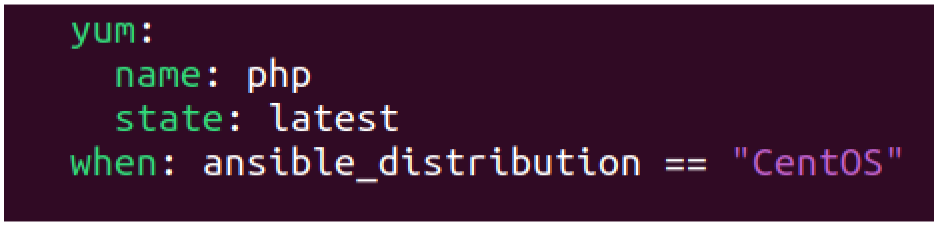
```
---
- hosts: all
  become: true
  tasks:

  - 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.

Input



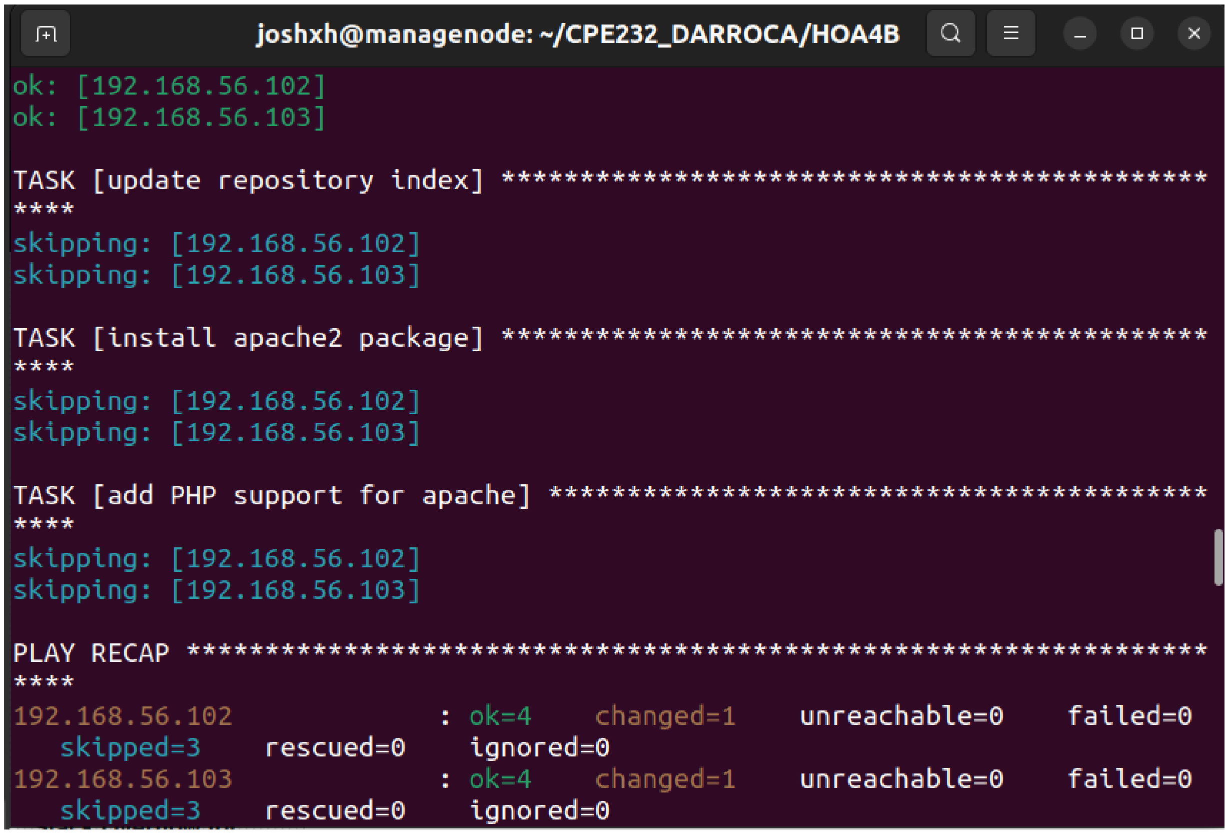
```
GNU nano 6.2
---
- hosts: all
  become: true
  tasks:

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

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

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


  - name: update repository index
    yum:
      update_cache: yes
    when: ansible_distribution == "CentOS"

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

  - name: add PHP support for apache
    yum:
```



```
    yum:
      name: php
      state: latest
    when: ansible_distribution == "CentOS"
```

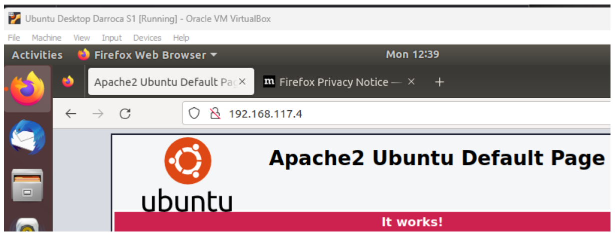
Run *ansible-playbook --ask-become-pass install_apache.yml* and describe the result.

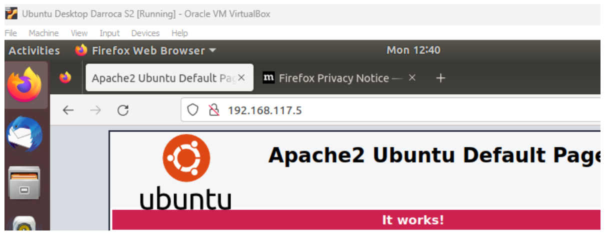
Process



```
joshxh@managenode: ~/CPE232_DARROCA/HOA4B
ok: [192.168.56.102]
ok: [192.168.56.103]

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

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

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

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
```

Output


Ubuntu Desktop Darroca S1 [Running] - Oracle VM VirtualBox
Activities
Firefox Web Browser
Mon 12:39
Apache2 Ubuntu Default Pa
Firefox Privacy Notice
192.168.117.4
Apache2 Ubuntu Default Page
ubuntu
It works!


Ubuntu Desktop Darroca S2 [Running] - Oracle VM VirtualBox
Activities
Firefox Web Browser
Mon 12:40
Apache2 Ubuntu Default Pa
Firefox Privacy Notice
192.168.117.5
Apache2 Ubuntu Default Page
ubuntu
It works!

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.

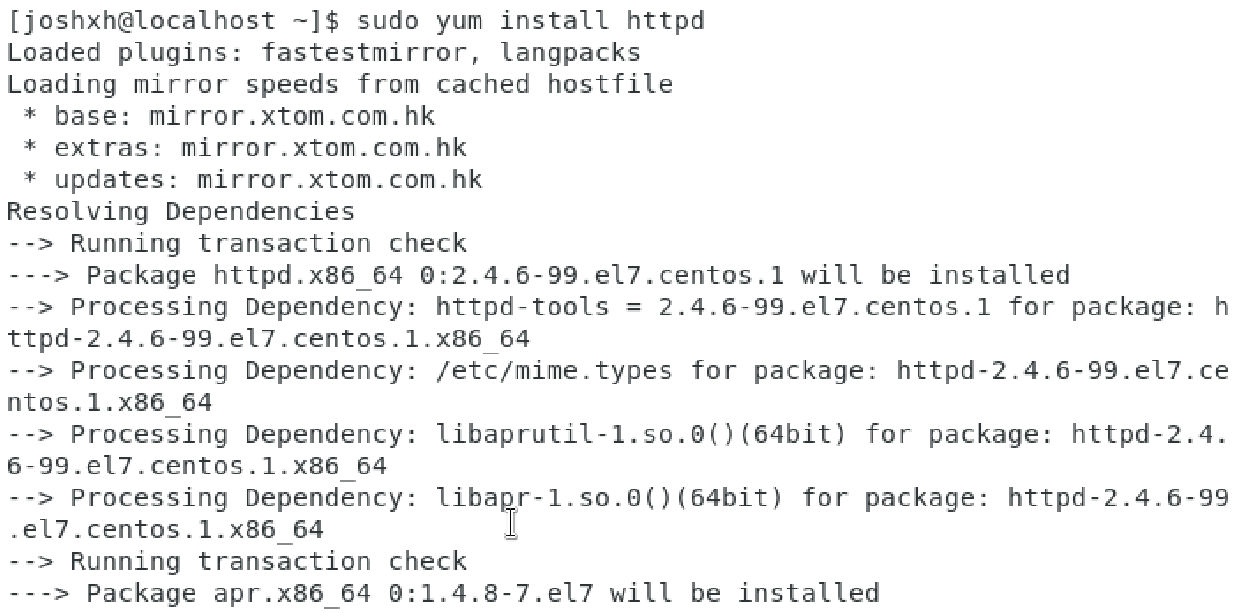
   5.1 To activate, go to the CentOS VM terminal and enter the following:
   systemctl status httpd

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

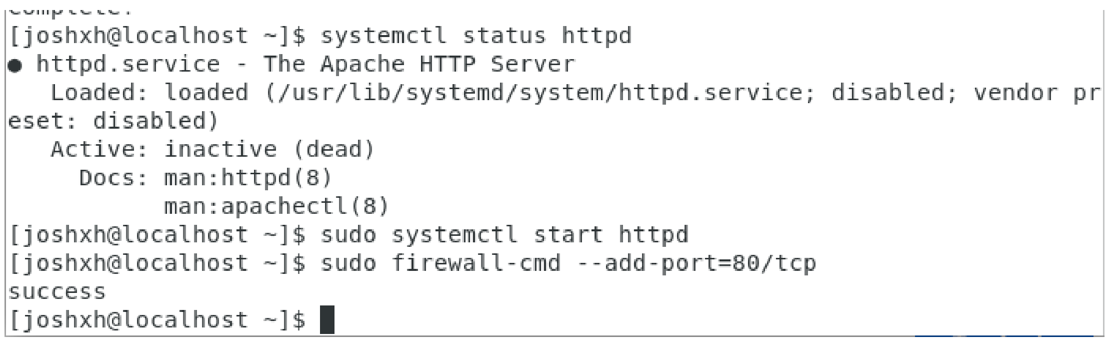
5.2 Issue the following command to start the service:
sudo systemctl start httpd
(When prompted, enter the sudo password)
sudo firewall-cmd --add-port=80/tcp
(The result should be a success)

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)

Input

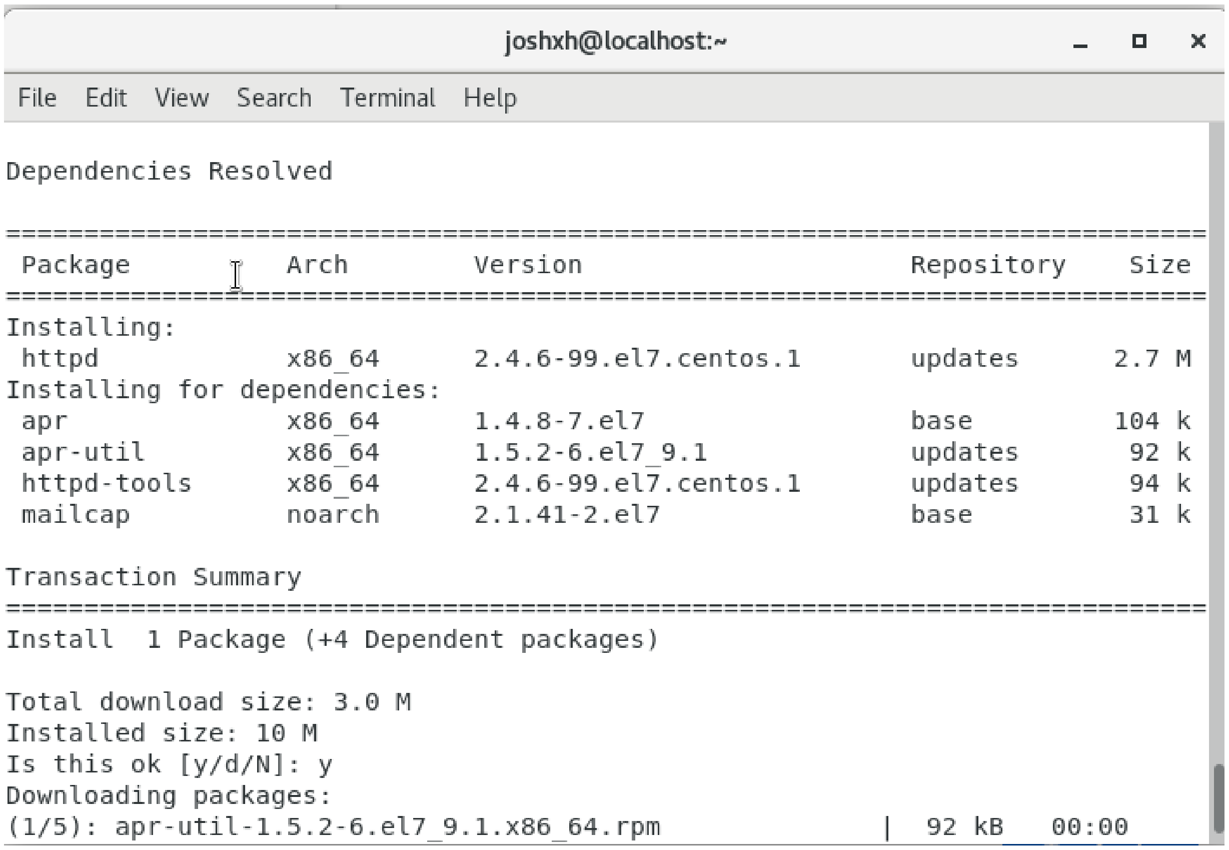


```
[joshxh@localhost ~]$ sudo yum install httpd
Loaded plugins: fastestmirror, langpacks
Loading mirror speeds from cached hostfile
 * base: mirror.xtom.com.hk
 * extras: mirror.xtom.com.hk
 * updates: mirror.xtom.com.hk
Resolving Dependencies
--> Running transaction check
---> Package httpd.x86_64 0:2.4.6-99.el7.centos.1 will be installed
--> Processing Dependency: httpd-tools = 2.4.6-99.el7.centos.1 for package: h
ttpd-2.4.6-99.el7.centos.1.x86_64
--> Processing Dependency: /etc/mime.types for package: httpd-2.4.6-99.el7.ce
ntos.1.x86_64
--> Processing Dependency: libaprutil-1.so.0()(64bit) for package: httpd-2.4.
6-99.el7.centos.1.x86_64
--> Processing Dependency: libapr-1.so.0()(64bit) for package: httpd-2.4.6-99
.el7.centos.1.x86_64
--> Running transaction check
---> Package apr.x86_64 0:1.4.8-7.el7 will be installed
```

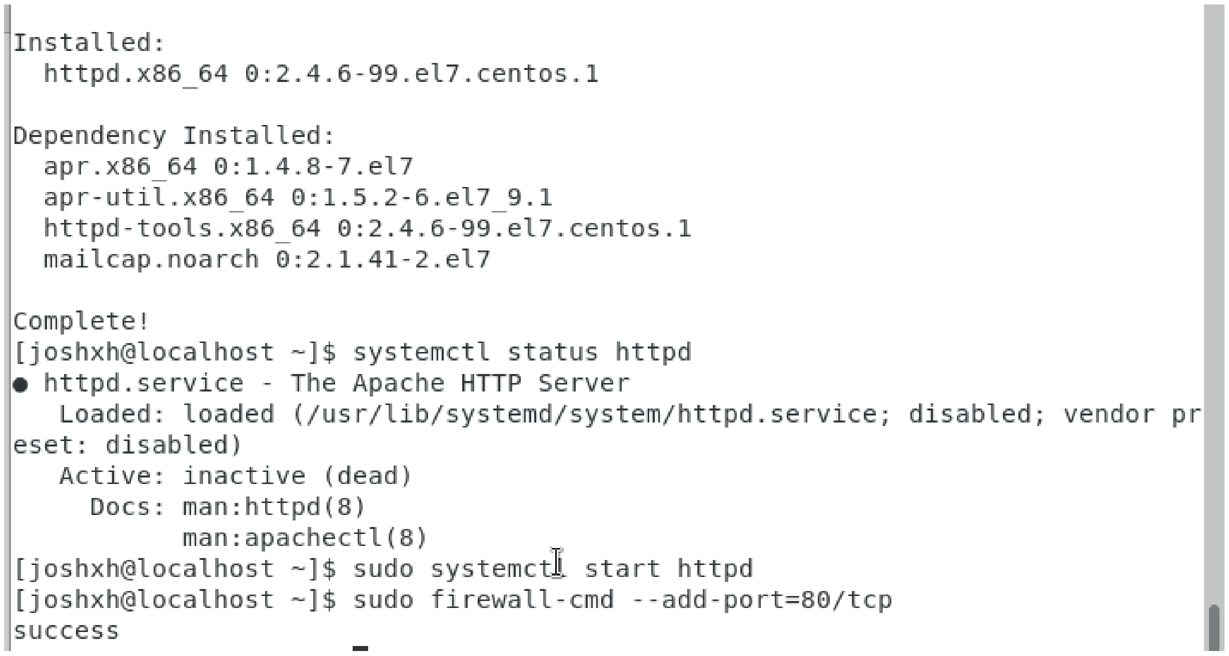


```
[joshxh@localhost ~]$ systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; vendor pr
eset: disabled)
   Active: inactive (dead)
     Docs: man:httpd(8)
           man:apachectl(8)
[joshxh@localhost ~]$ sudo systemctl start httpd
[joshxh@localhost ~]$ sudo firewall-cmd --add-port=80/tcp
success
[joshxh@localhost ~]$ 
```

Process

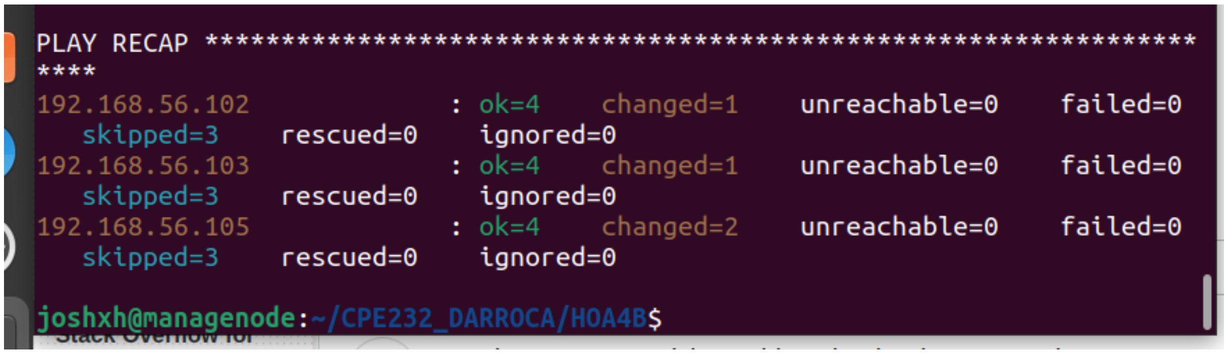


```
Dependencies Resolved

================================================================================
 Package          Arch        Version                      Repository    Size
================================================================================
Installing:
 httpd            x86_64      2.4.6-99.el7.centos.1        updates      2.7 M
Installing for dependencies:
 apr              x86_64      1.4.8-7.el7                  base         104 k
 apr-util         x86_64      1.5.2-6.el7_9.1              updates       92 k
 httpd-tools      x86_64      2.4.6-99.el7.centos.1        updates       94 k
 mailcap          noarch      2.1.41-2.el7                 base          31 k

Transaction Summary
================================================================================
Install  1 Package (+4 Dependent packages)

Total download size: 3.0 M
Installed size: 10 M
Is this ok [y/d/N]: y
Downloading packages:
(1/5): apr-util-1.5.2-6.el7_9.1.x86_64.rpm             |  92 kB   00:00
```

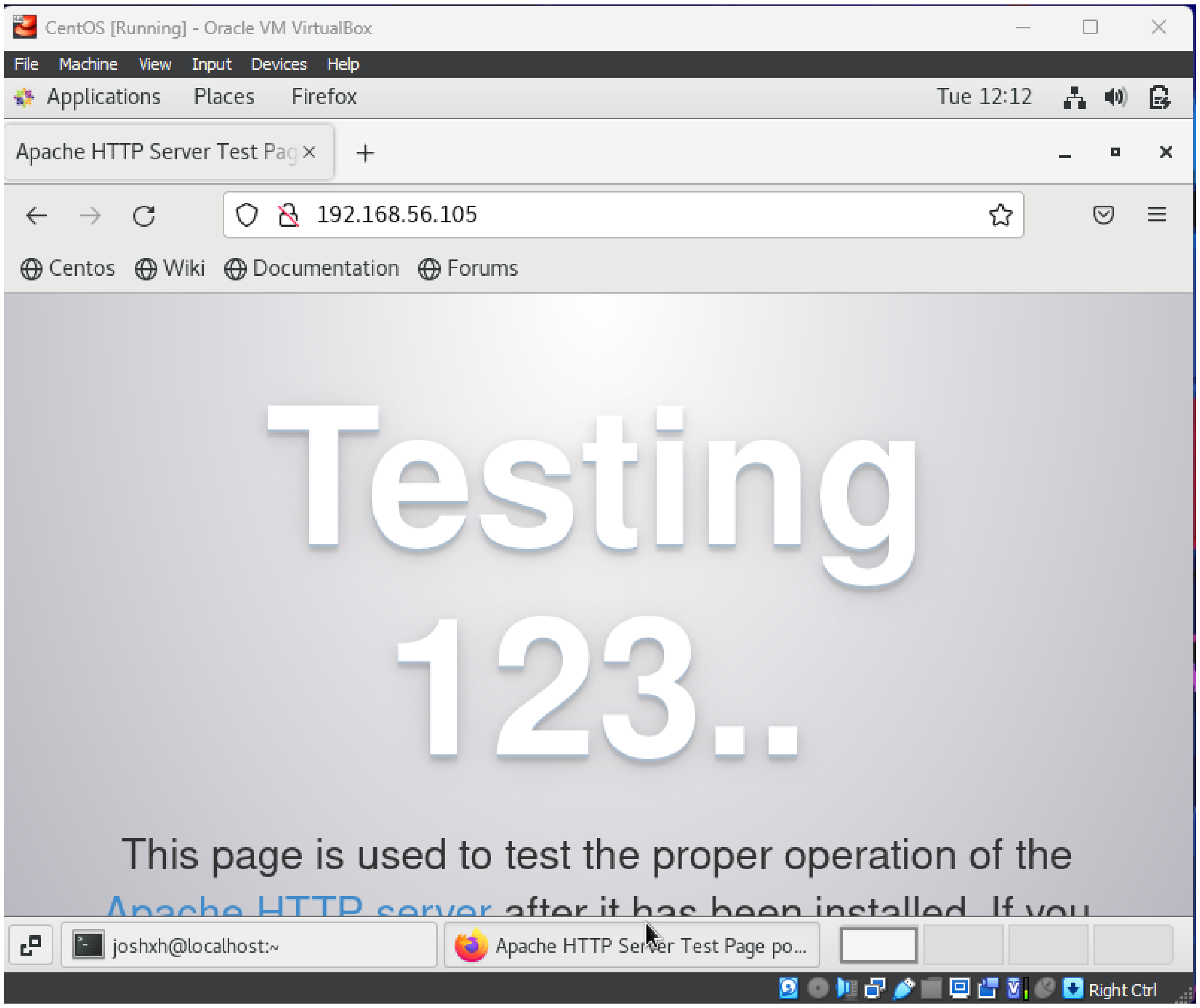


```
Installed:
  httpd.x86_64 0:2.4.6-99.el7.centos.1

Dependency Installed:
  apr.x86_64 0:1.4.8-7.el7
  apr-util.x86_64 0:1.5.2-6.el7_9.1
  httpd-tools.x86_64 0:2.4.6-99.el7.centos.1
  mailcap.noarch 0:2.1.41-2.el7

Complete!
[joshxh@localhost ~]$ systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; vendor pr
eset: disabled)
   Active: inactive (dead)
     Docs: man:httpd(8)
           man:apachectl(8)
[joshxh@localhost ~]$ sudo systemctl start httpd
[joshxh@localhost ~]$ sudo firewall-cmd --add-port=80/tcp
success
```

Output



```
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
192.168.56.105             : ok=4    changed=2    unreachable=0    failed=0
   skipped=3    rescued=0    ignored=0

joshxh@managenode:~/CPE232_DARROCA/HOA4B$
```


CentOS [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places Firefox
Tue 12:12
Apache HTTP Server Test Pag
192.168.56.105
Centos Wiki Documentation Forums
joshxh@localhost:~
Apache HTTP Server Test Page po...
Right Ctrl

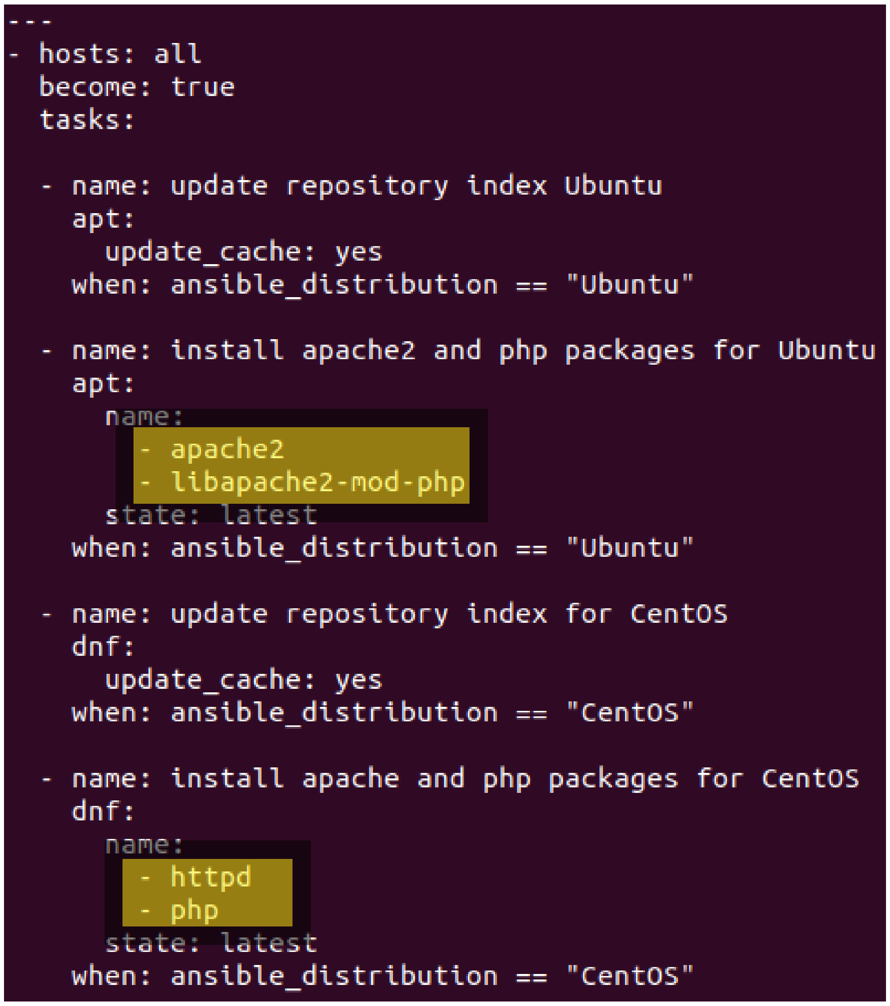
Testing 123..

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

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.

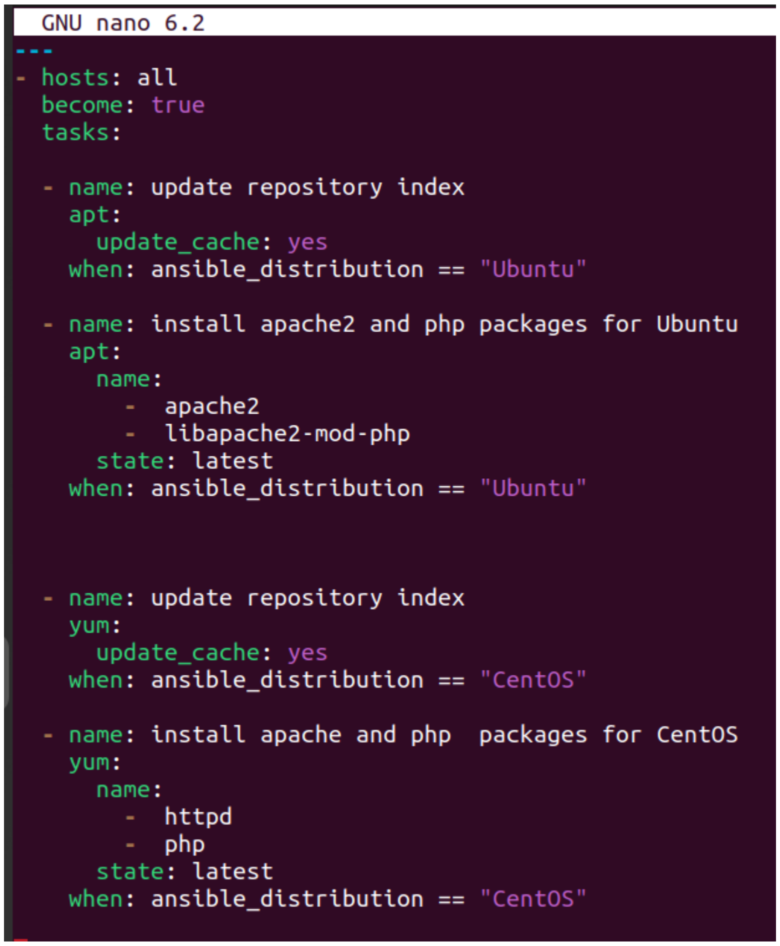
1. Edit the playbook *install_apache.yml*. Currently, we have three tasks targeting our Ubuntu machines and 3 tasks targeting our CentOS machine. Right now, we try to consolidate some tasks that are typically the same. For example, we can consolidate two plays that install packages. We can do that by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

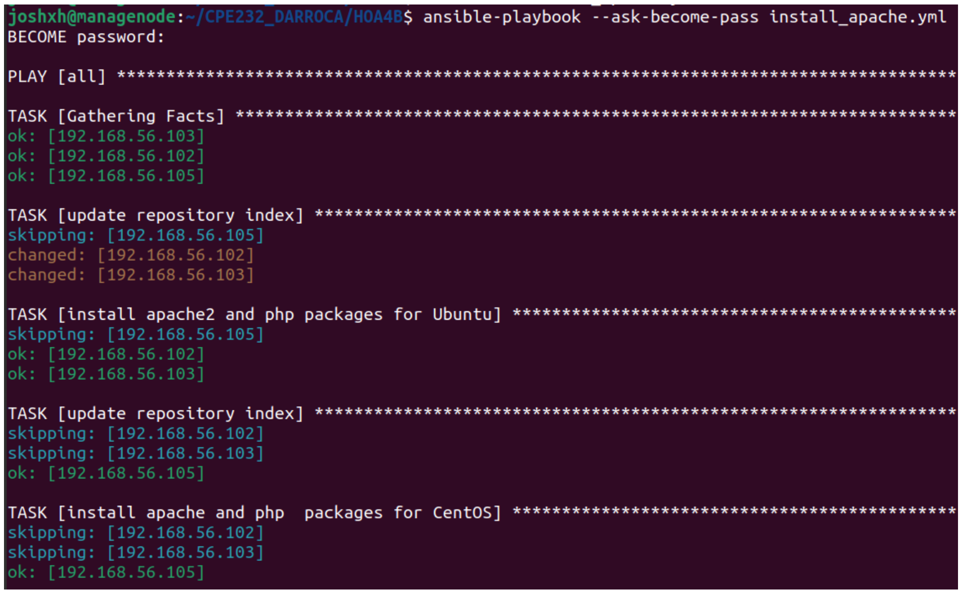
Input



```
 GNU nano 6.2
---
- hosts: all
  become: true
  tasks:

  - name: update repository index
    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
    yum:
      update_cache: yes
    when: ansible_distribution == "CentOS"

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

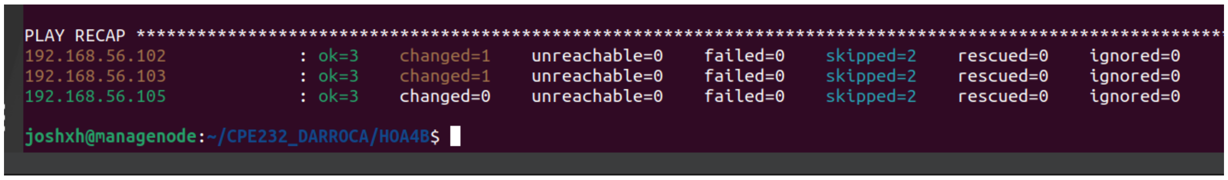
Run *ansible-playbook --ask-become-pass install_apache.yml* and describe the result.

Process



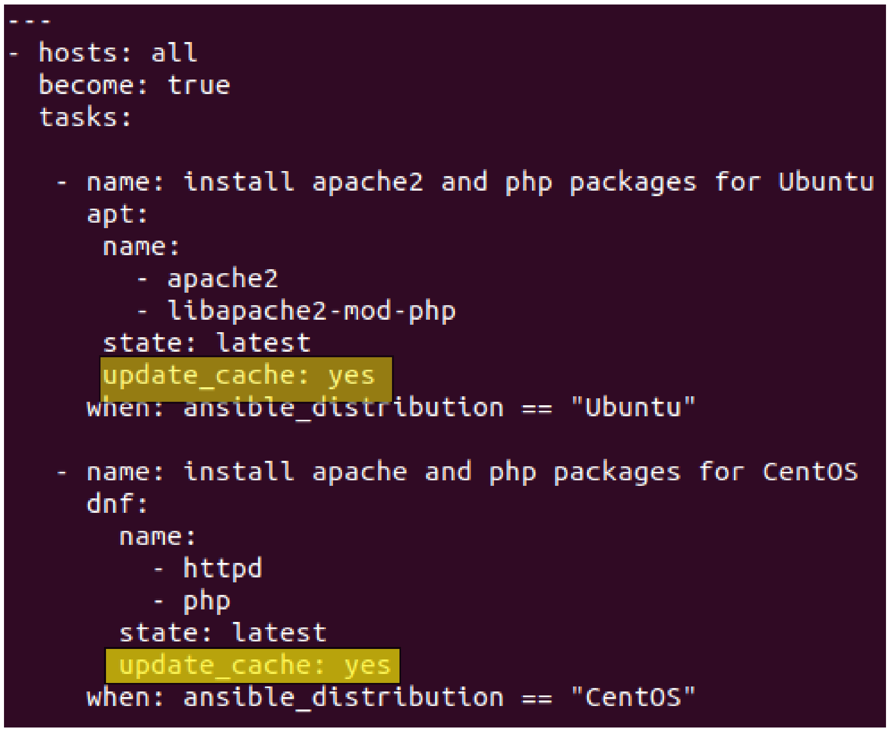
```
joshxh@managenode:~/CPE232_DARROCA/HOA4B$ ansible-playbook --ask-become-pass install_apache.yml
BECOME password:

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

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

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

TASK [install apache2 and php packages for Ubuntu] *******************************************
skipping: [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: [192.168.56.105]

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

Output



```
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
192.168.56.105             : ok=3    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0

joshxh@managenode:~/CPE232_DARROCA/HOA4B$
```

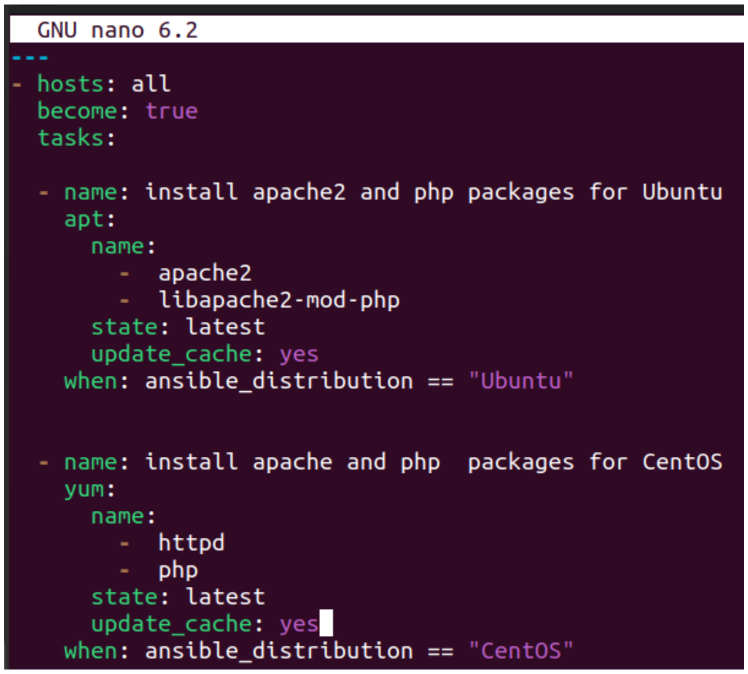
2. Edit the playbook *install_apache.yml* again. In task 2.1, we consolidated the plays into one play. This time we can actually consolidated everything in just 2 plays. This can be done by removing the update repository play and putting the command *update_cache: yes* below the command *state: latest*.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

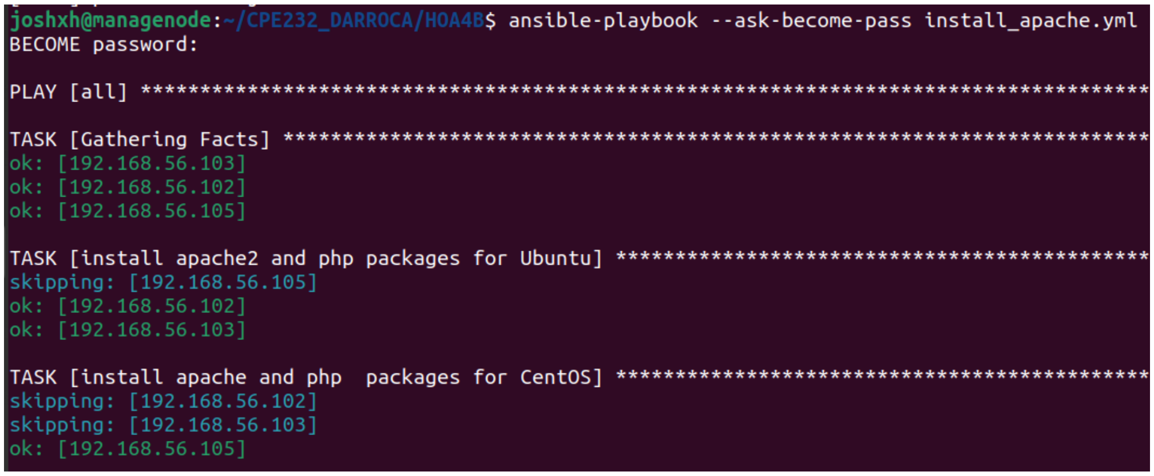
Input



```
 GNU nano 6.2
---
- hosts: all
  become: true
  tasks:

  - name: install apache2 and php packages for Ubuntu
    apt:
      name:
        -  apache2
        -  libapache2-mod-php
      state: latest
      update_cache: yes
    when: ansible_distribution == "Ubuntu"


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

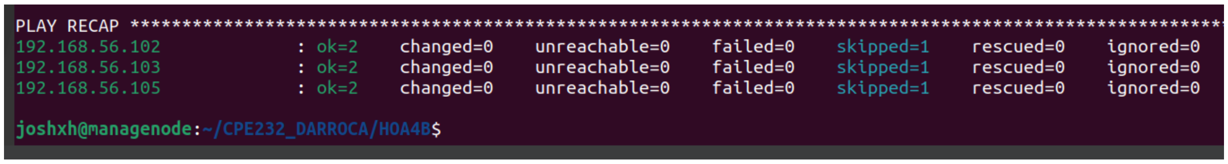
Run *ansible-playbook --ask-become-pass install_apache.yml* and describe the result.

Process



```
joshxh@managenode:~/CPE232_DARROCA/HOA4B$ ansible-playbook --ask-become-pass install_apache.yml
BECOME password:

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

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

TASK [install apache2 and php packages for Ubuntu] ***********************************
skipping: [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: [192.168.56.105]
```

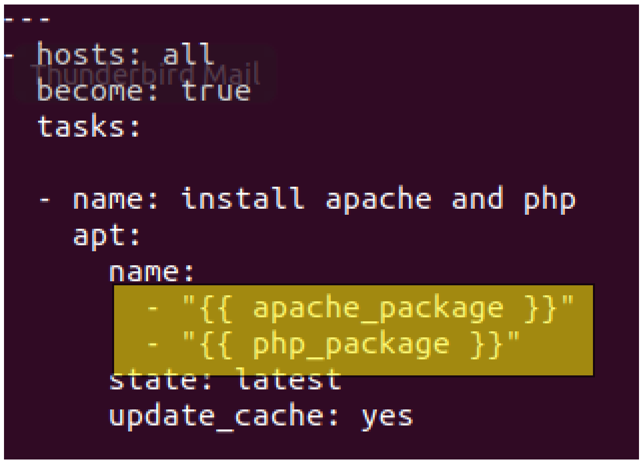
Output



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

joshxh@managenode:~/CPE232_DARROCA/HOA4B$
```

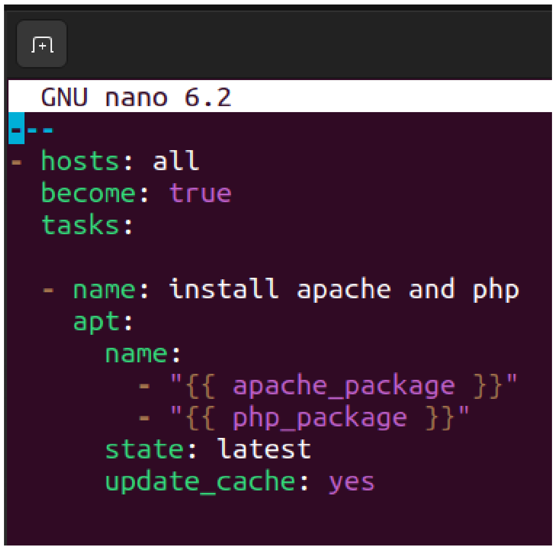
3. Finally, we can consolidate these 2 plays in just 1 play. This can be done by declaring variables that will represent the packages that we want to install. Basically, the apache_package and php_package are variables. The names are arbitrary, which means we can choose different names. We also take 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
```

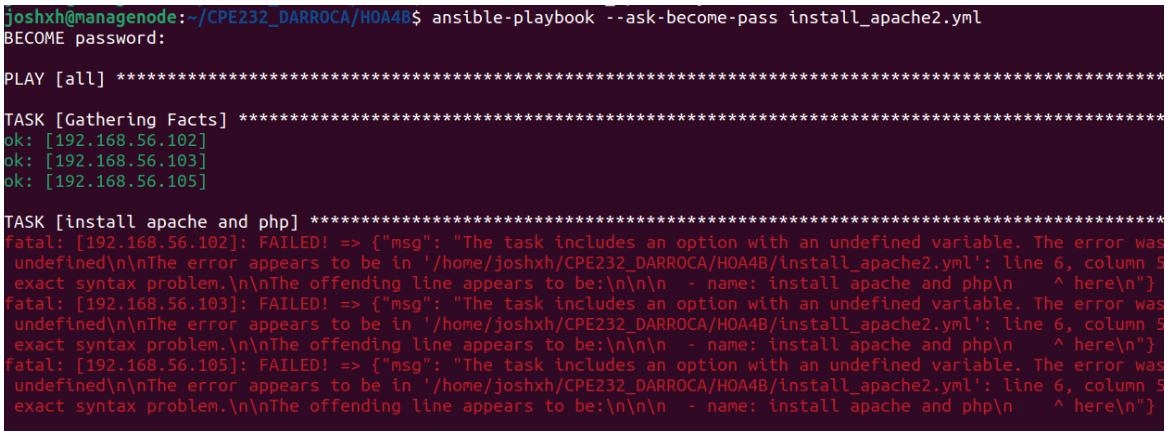
Input



```
 GNU nano 6.2
---
- hosts: all
  become: true
  tasks:

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

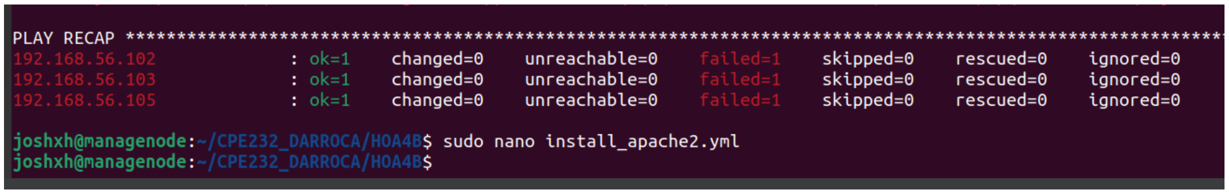
Run *ansible-playbook --ask-become-pass install_apache.yml* and describe the result.

Process



```
joshxh@managenode:~/CPE232_DARROCA/HOA4B$ ansible-playbook --ask-become-pass install_apache2.yml
BECOME password:

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

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

TASK [install apache and php] **************************************************************************
fatal: [192.168.56.102]: FAILED! => {"msg": "The task includes an option with an undefined variable. The error was
 undefined\n\nThe error appears to be in '/home/joshxh/CPE232_DARROCA/HOA4B/install_apache2.yml': line 6, column 5
 exact syntax problem.\n\nThe offending line appears to be:\n\n\n  - name: install apache and php\n    ^ here\n"}
fatal: [192.168.56.103]: FAILED! => {"msg": "The task includes an option with an undefined variable. The error was
 undefined\n\nThe error appears to be in '/home/joshxh/CPE232_DARROCA/HOA4B/install_apache2.yml': line 6, column 5
 exact syntax problem.\n\nThe offending line appears to be:\n\n\n  - name: install apache and php\n    ^ here\n"}
fatal: [192.168.56.105]: FAILED! => {"msg": "The task includes an option with an undefined variable. The error was
 undefined\n\nThe error appears to be in '/home/joshxh/CPE232_DARROCA/HOA4B/install_apache2.yml': line 6, column 5
 exact syntax problem.\n\nThe offending line appears to be:\n\n\n  - name: install apache and php\n    ^ here\n"}
```

Output



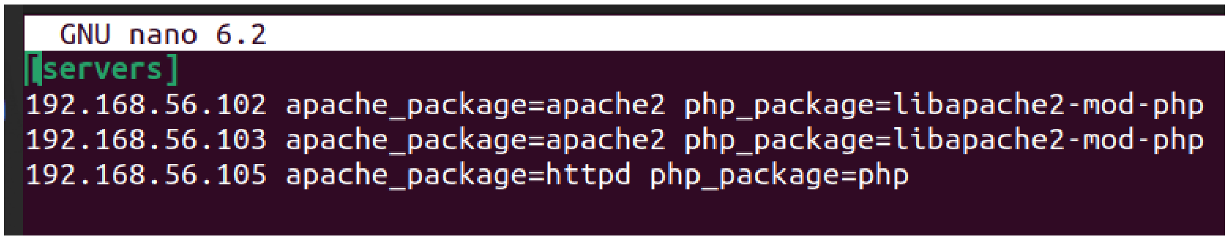
```
PLAY RECAP ****************************************************************************************
192.168.56.102             : ok=1    changed=0    unreachable=0    failed=1    skipped=0    rescued=0    ignored=0
192.168.56.103             : ok=1    changed=0    unreachable=0    failed=1    skipped=0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

joshxh@managenode:~/CPE232_DARROCA/HOA4B$ sudo nano install_apache2.yml
joshxh@managenode:~/CPE232_DARROCA/HOA4B$
```

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
```

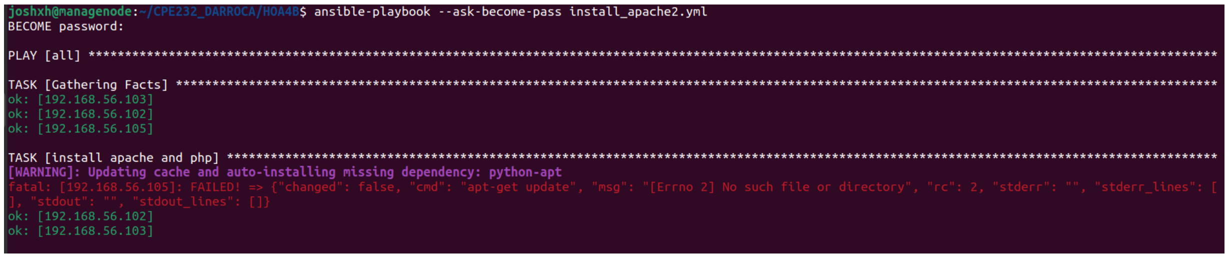
Make sure to save the *inventory* file and exit.

Input



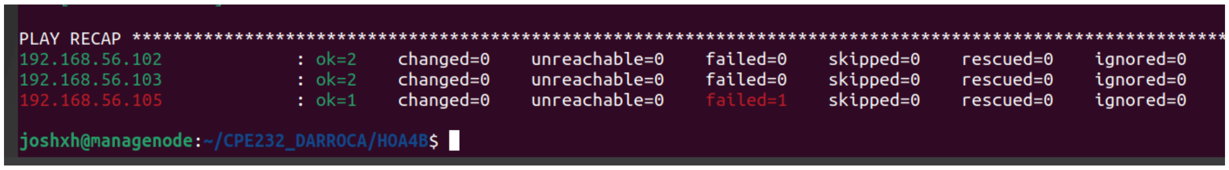
```
  GNU nano 6.2
[servers]
192.168.56.102 apache_package=apache2 php_package=libapache2-mod-php
192.168.56.103 apache_package=apache2 php_package=libapache2-mod-php
192.168.56.105 apache_package=httpd php_package=php
```

Process



```
joshxh@managenode:~/CPE232_DARROCA/HOA4B$ ansible-playbook --ask-become-pass install_apache2.yml
BECOME password:

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

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

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

Output

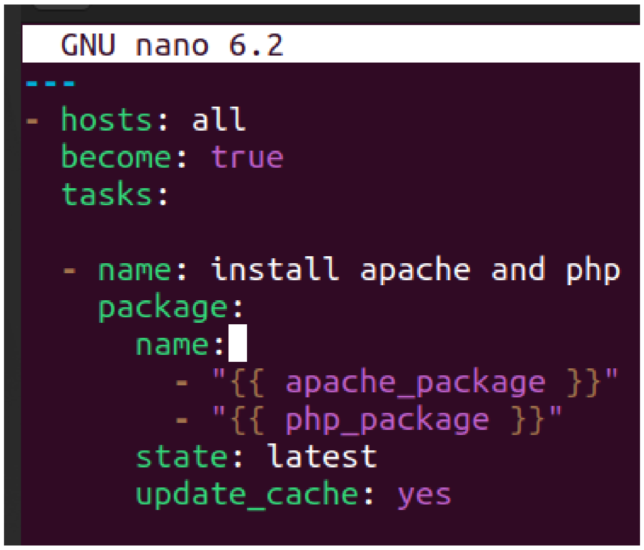


```
PLAY RECAP ****************************************************************************************
192.168.56.102             : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
192.168.56.103             : ok=2    changed=0    unreachable=0    failed=0    skipped=0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

joshxh@managenode:~/CPE232_DARROCA/HOA4B$
```

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]()

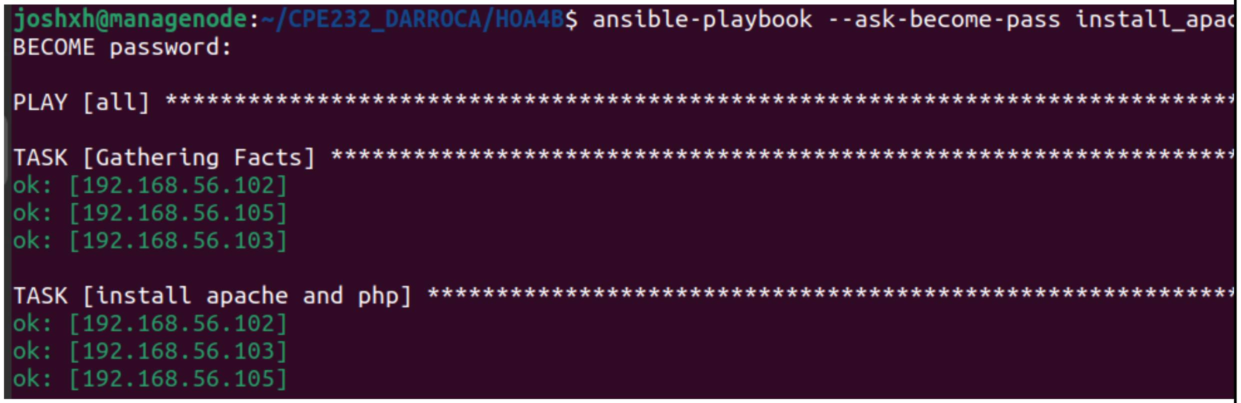
Input



```
GNU nano 6.2
---
- hosts: all
  become: true
  tasks:

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

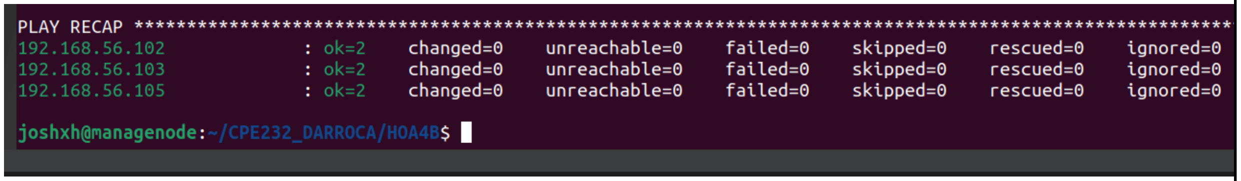
Run *ansible-playbook --ask-become-pass install_apache.yml* and describe the result.

Process



```
joshxh@managenode:~/CPE232_DARROCA/HOA4B$ ansible-playbook --ask-become-pass install_apac
BECOME password:

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

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

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

Output



```
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
192.168.56.105             : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

joshxh@managenode:~/CPE232_DARROCA/HOA4B$
```

Reflections:

Answer the following:

1. Why do you think refactoring of playbook codes is important?
    - It makes consolidation of playbooks more convenient and less tedious. If we were to type an individual command every time, imagine the time spent on typing and checking that. Compared to refactoring, we can just type multiple names of what we want to do in a single command. Good thing that the package command exists, it will replace the ‘apt’ or ‘dnf/’yum’ so that you will not worry about what OS you’re using.

2. When do we use the “when” command in playbook?
    - We use the ‘when’ command in times where we declare what type of installation process we want. If we declare the command as ‘apt’ in the script, we will use ‘when’ to indicate that its distribution is a Debian distribution or specifically, Ubuntu. On the other hand, if we declare it as ‘dnf/yum’, we will indicate that its distribution is affiliated with the RedHat distribution or in this activity, CentOS.

Conclusion:

In this activity where we are tasked to perform scripts in order to install apache, httpd, and their php support. I was able to perform all of the required task from using the ‘when’ command in the playbook across Debian and RedHat distributions to applying refactoring techniques in cleaning up playbook codes. In spite of all the errors that I’ve faced in this activity, I was still able to accomplish a late submission. One of the errors that became a roadblock for me is having a functional CentOS 7 because some of that OS that I have installed always gave me the error that prevented me from moving forward. Thankfully, after several re-installation of CentOS 7, I was able to move on and it finally worked. Because of this occurrence, it dawned on me that no matter how hard you try, sometimes the problem is just doing things from the start all over again. All in all, this activity helped me to learn more about system administrations especially in the field of making playbooks and how to make them less tedious to do and I hope I can learn more in the future.