Name: John Era Roldan	**Date Performed: 9/12/2025**
Course/Section: CPE212 CPE31S2	**Date Submitted: 9/12/2025**
Instructor: Engr. Robin Valenzuela	**Semester and SY:**

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**:

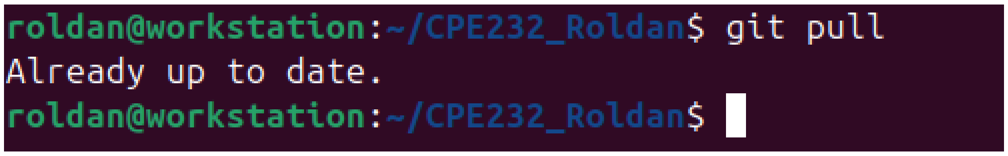
We are going to look at a way that we can differentiate a playbook by a host in terms of which distribution the host is running. It's very common in most Linux shops to run multiple distributions, for example, Ubuntu shop or Debian shop and you need a different distribution for a one off-case or perhaps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

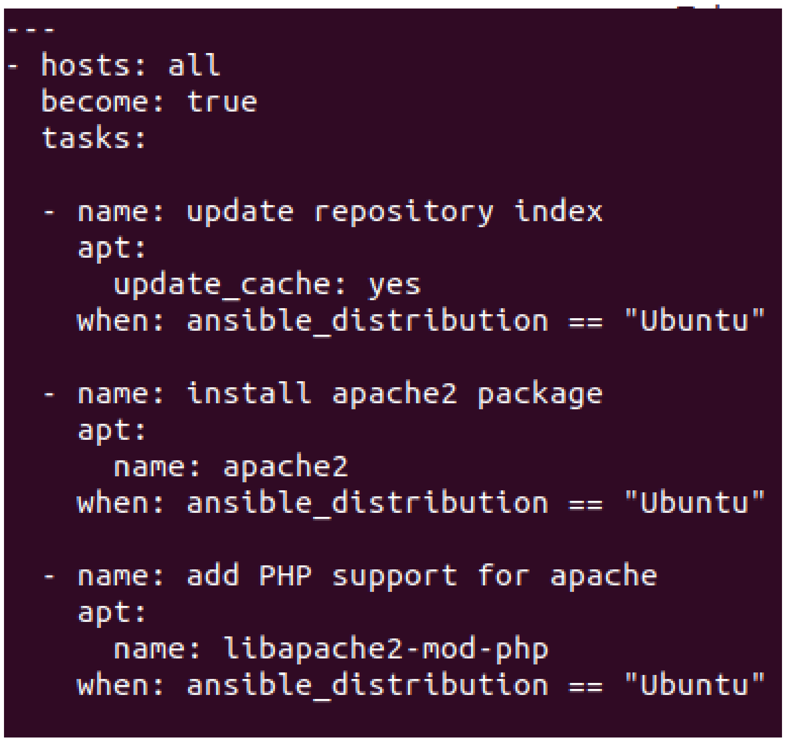
Task 1: Use when command for different distributions

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?



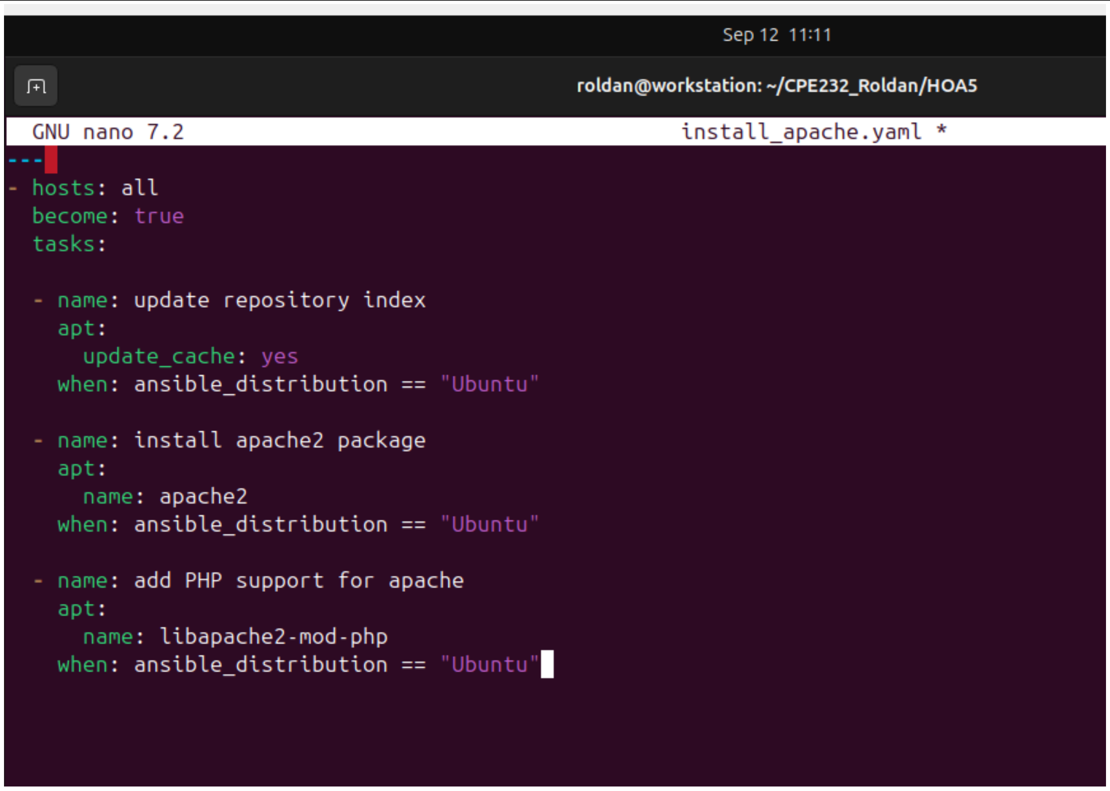
```
roldan@workstation:~/CPE232_Roldan$ git pull
Already up to date.
roldan@workstation:~/CPE232_Roldan$ 
```

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.



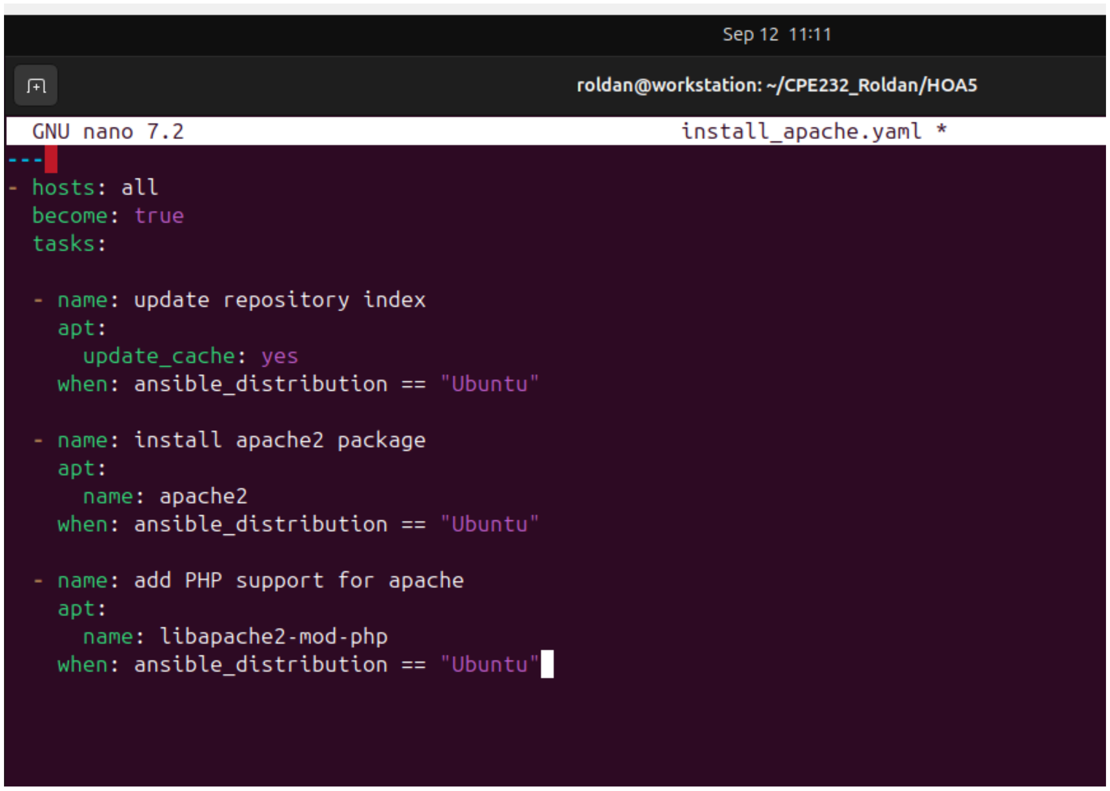
```
---
- hosts: all
  become: true
  tasks:

  - 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.

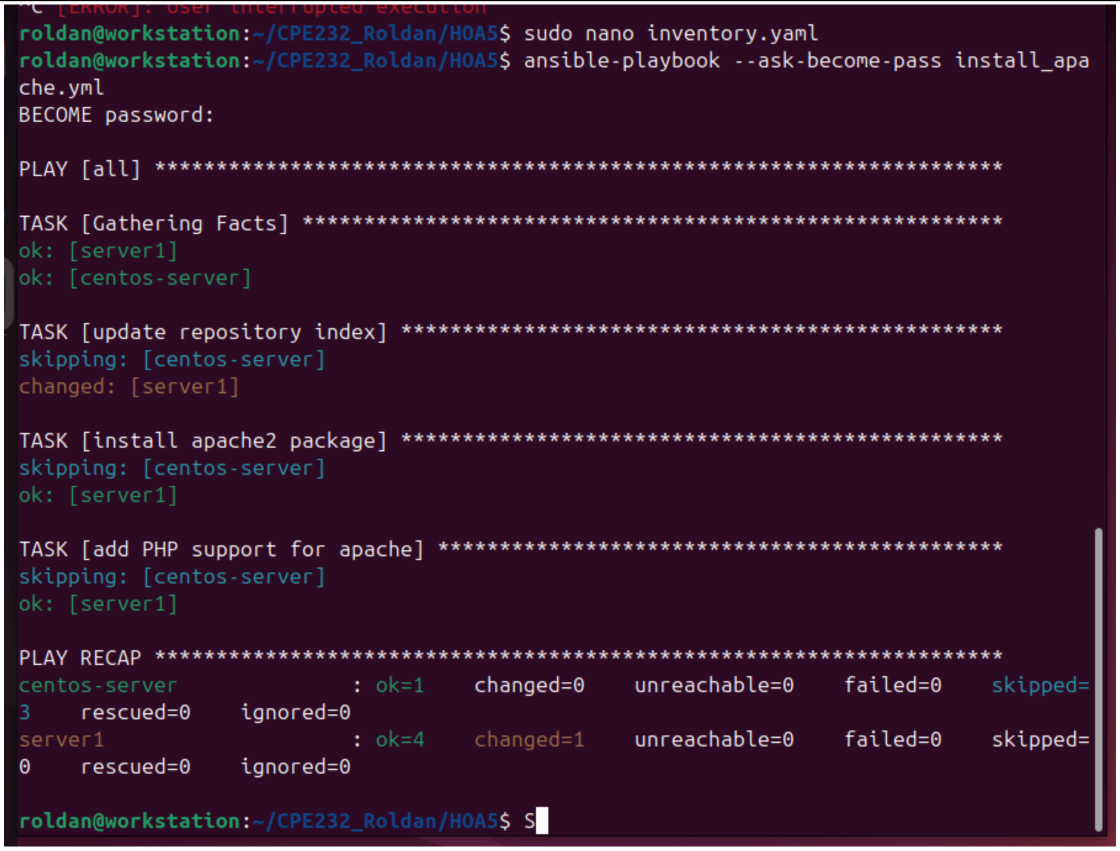


```
Sep 12  11:11
roldan@workstation: ~/CPE232_Roldan/HOA5
GNU nano 7.2                                        install_apache.yaml *
---
- hosts: all
  become: true
  tasks:

  - 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.



```
Sep 12  11:11
roldan@workstation: ~/CPE232_Roldan/HOA5
GNU nano 7.2                                        install_apache.yaml *
---
- 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"
```



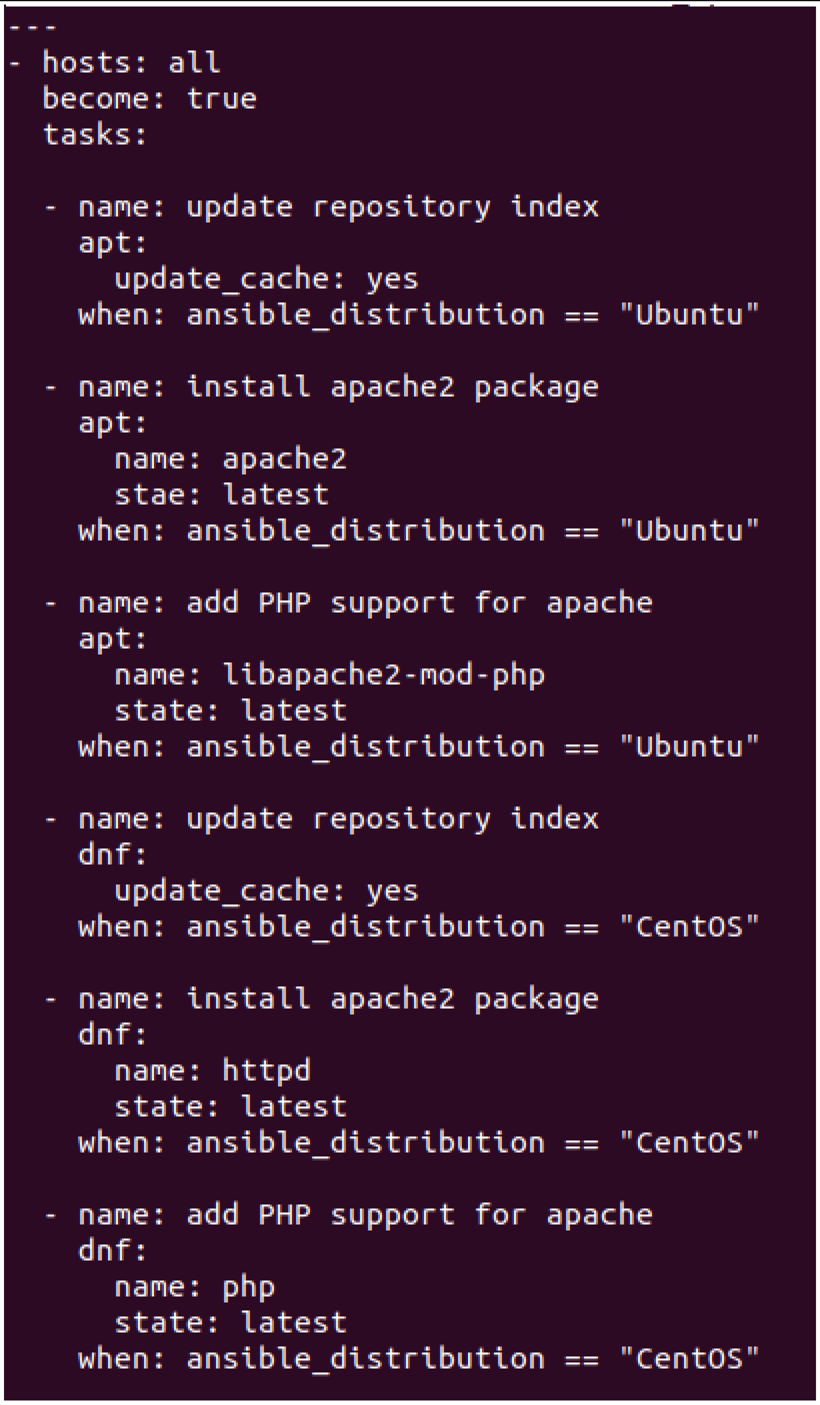
```
^C [ERROR]: User interrupted execution
roldan@workstation:~/CPE232_Roldan/HOA5$ sudo nano inventory.yaml
roldan@workstation:~/CPE232_Roldan/HOA5$ ansible-playbook --ask-become-pass install_apa
che.yml
BECOME password:

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

TASK [Gathering Facts] *********************************************************
ok: [server1]
ok: [centos-server]

TASK [update repository index] *************************************************
skipping: [centos-server]
changed: [server1]

TASK [install apache2 package] *************************************************
skipping: [centos-server]
ok: [server1]

TASK [add PHP support for apache] **********************************************
skipping: [centos-server]
ok: [server1]

PLAY RECAP *********************************************************************
centos-server              : ok=1    changed=0    unreachable=0    failed=0    skipped=
3    rescued=0    ignored=0
server1                    : ok=4    changed=1    unreachable=0    failed=0    skipped=
0    rescued=0    ignored=0

roldan@workstation:~/CPE232_Roldan/HOA5$ S
```

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

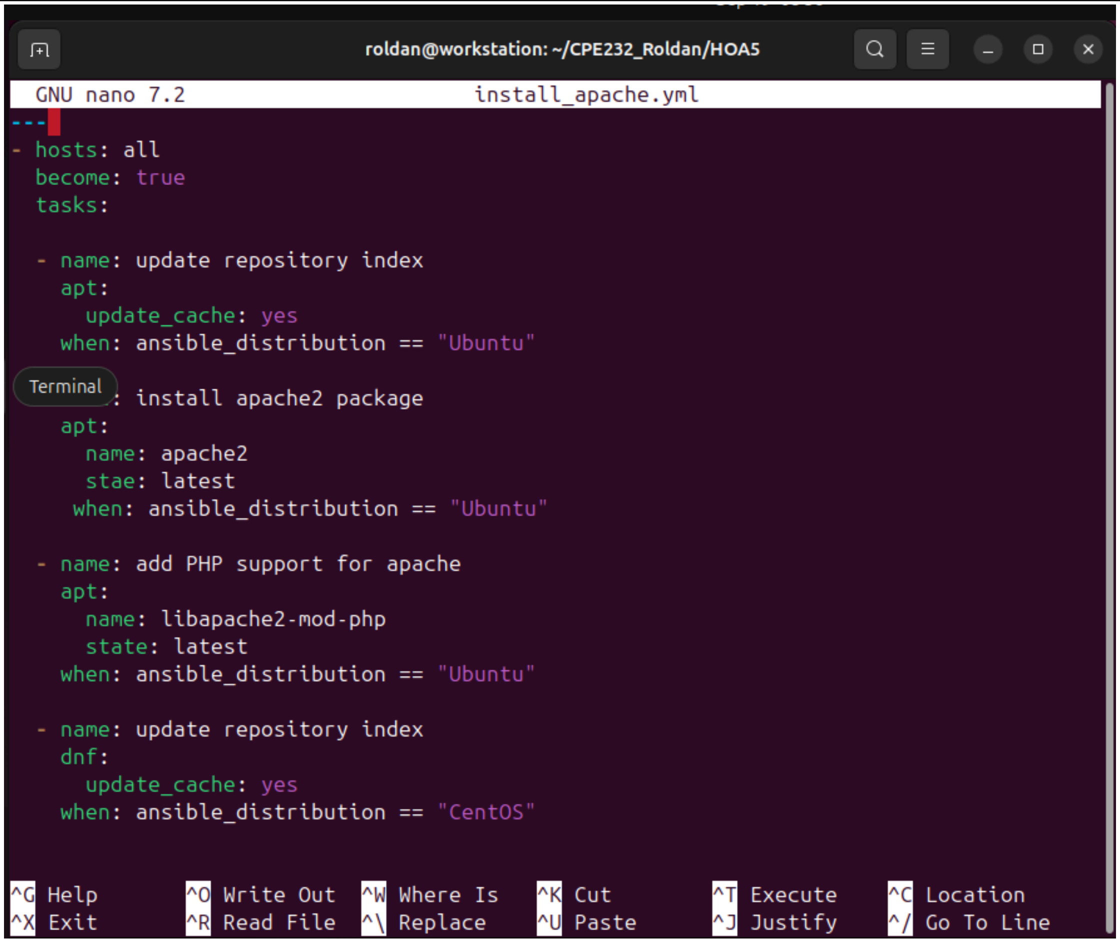
```
-   name: update repository index
    apt:
       update_cache: yes
    when: ansible_distribution in [“Debian”, “Ubuntu]
```

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

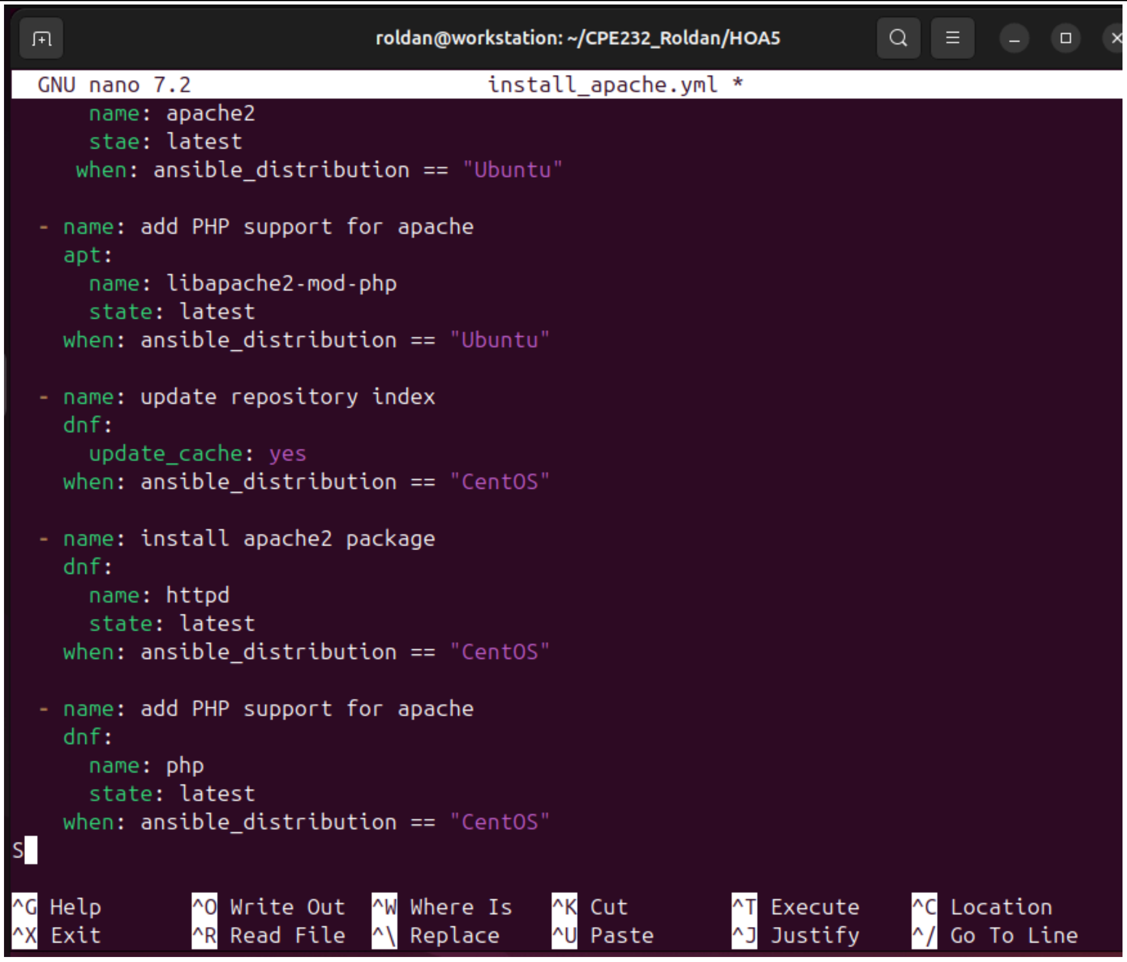
4. Edit the *install_apache.yml* file and insert the lines shown below.


```
---
- 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.

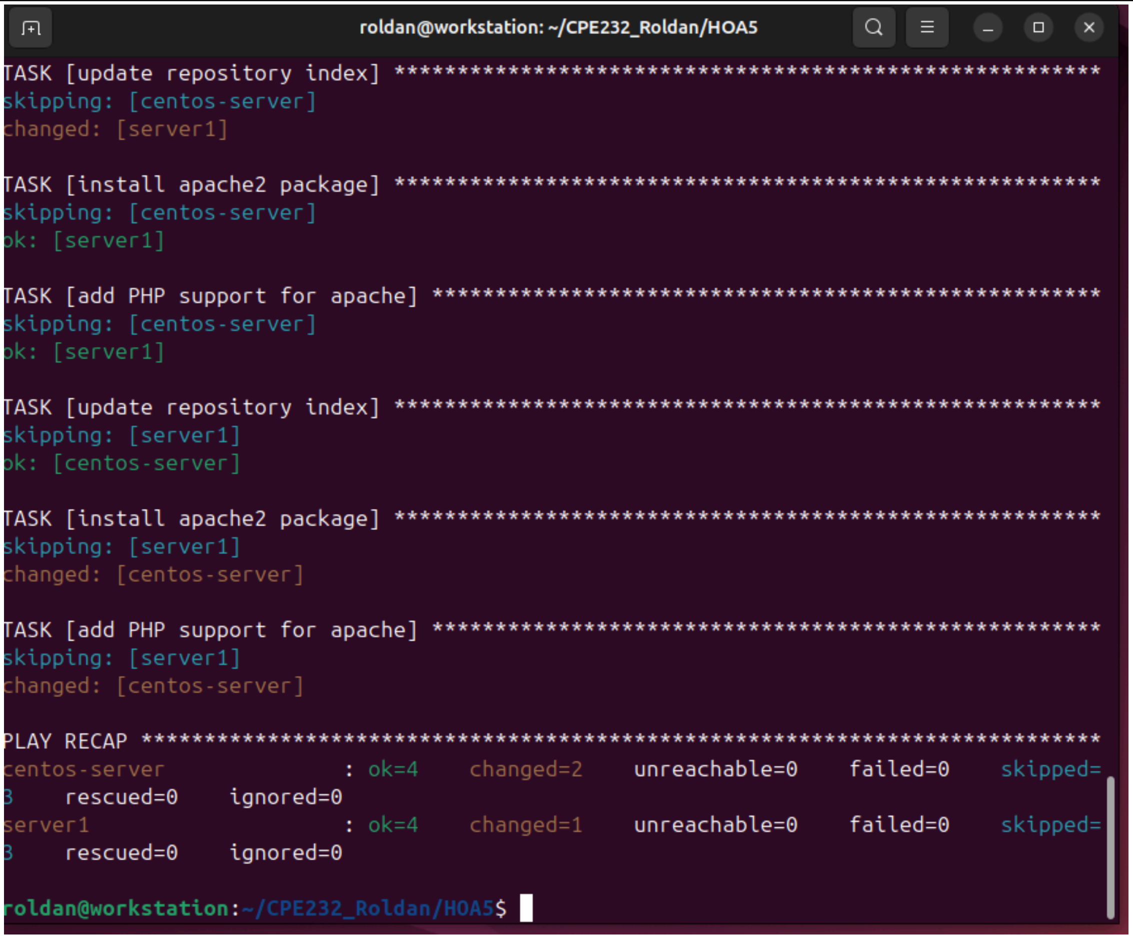


```
  GNU nano 7.2                       install_apache.yml
---
- hosts: all
  become: true
  tasks:

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

         : 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"


^G Help        ^O Write Out   ^W Where Is    ^K Cut         ^T Execute     ^C Location
^X Exit        ^R Read File   ^\ Replace     ^U Paste       ^J Justify     ^/ Go To Line
```



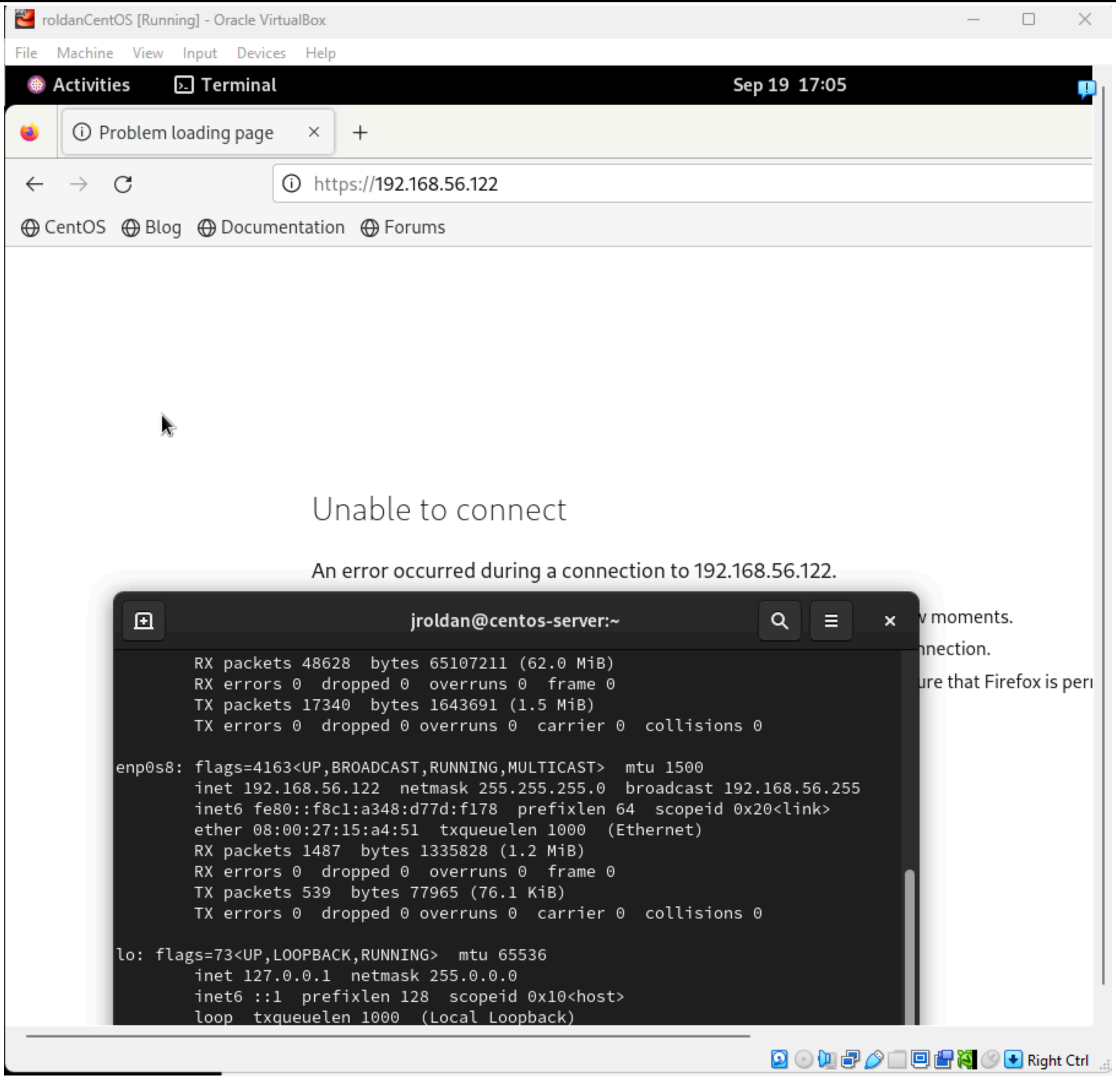
```
roldan@workstation: ~/CPE232_Roldan/HOA5

GNU nano 7.2                              install_apache.yml *
      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"
S

^G Help         ^O Write Out   ^W Where Is    ^K Cut         ^T Execute     ^C Location
^X Exit         ^R Read File   ^\ Replace     ^U Paste       ^J Justify     ^/ Go To Line
```

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



```
roldan@workstation: ~/CPE232_Roldan/HOA5
TASK [update repository index] ****************************************************
skipping: [centos-server]
changed: [server1]

TASK [install apache2 package] ****************************************************
skipping: [centos-server]
ok: [server1]

TASK [add PHP support for apache] *************************************************
skipping: [centos-server]
ok: [server1]

TASK [update repository index] ****************************************************
skipping: [server1]
ok: [centos-server]

TASK [install apache2 package] ****************************************************
skipping: [server1]
changed: [centos-server]

TASK [add PHP support for apache] *************************************************
skipping: [server1]
changed: [centos-server]

PLAY RECAP ************************************************************************
centos-server              : ok=4    changed=2    unreachable=0    failed=0    skipped=
3    rescued=0    ignored=0
server1                    : ok=4    changed=1    unreachable=0    failed=0    skipped=
3    rescued=0    ignored=0

roldan@workstation:~/CPE232_Roldan/HOA5$
```

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.

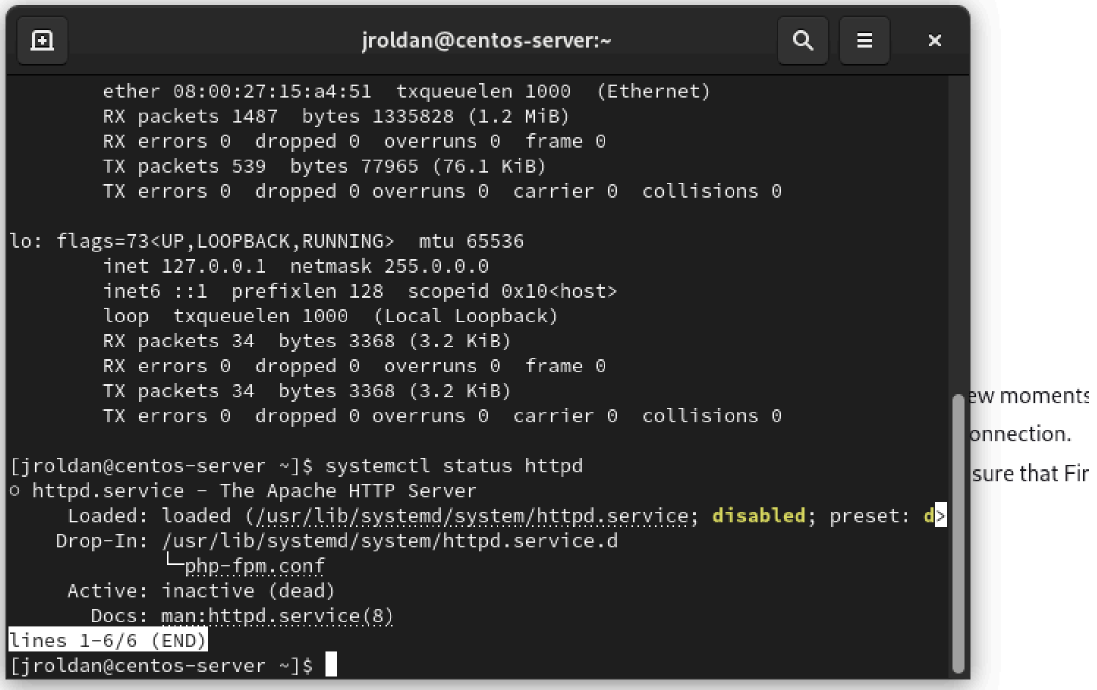

roldanCentOS [Running] - Oracle VirtualBox
File Machine View Input Devices Help
Activities Terminal Sep 19 17:05
Problem loading page
https://192.168.56.122
CentOS Blog Documentation Forums
jroldan@centos-server:~
Right Ctrl

Unable to connect

An error occurred during a connection to 192.168.56.122.

```
        RX packets 48628  bytes 65107211 (62.0 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 17340  bytes 1643691 (1.5 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.56.122  netmask 255.255.255.0  broadcast 192.168.56.255
        inet6 fe80::f8c1:a348:d77d:f178  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:15:a4:51  txqueuelen 1000  (Ethernet)
        RX packets 1487  bytes 1335828 (1.2 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 539  bytes 77965 (76.1 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
```

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.



```
        ether 08:00:27:15:a4:51  txqueuelen 1000  (Ethernet)
        RX packets 1487  bytes 1335828 (1.2 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 539  bytes 77965 (76.1 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 34  bytes 3368 (3.2 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 34  bytes 3368 (3.2 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

[jroldan@centos-server ~]$ 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: inactive (dead)
       Docs: man:httpd.service(8)
lines 1-6/6 (END)
[jroldan@centos-server ~]$
```

ew moments
onnection.
sure that Fir

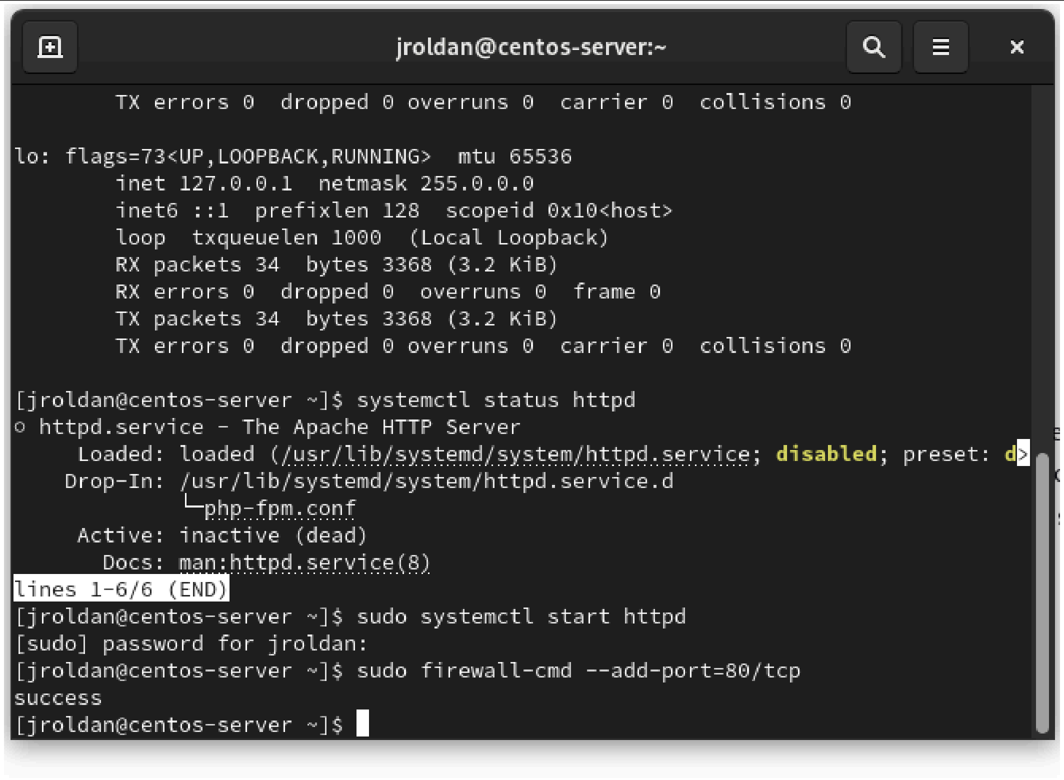
5.2 Issue the following command to start the service:

sudo systemctl start httpd

(When prompted, enter the sudo password)

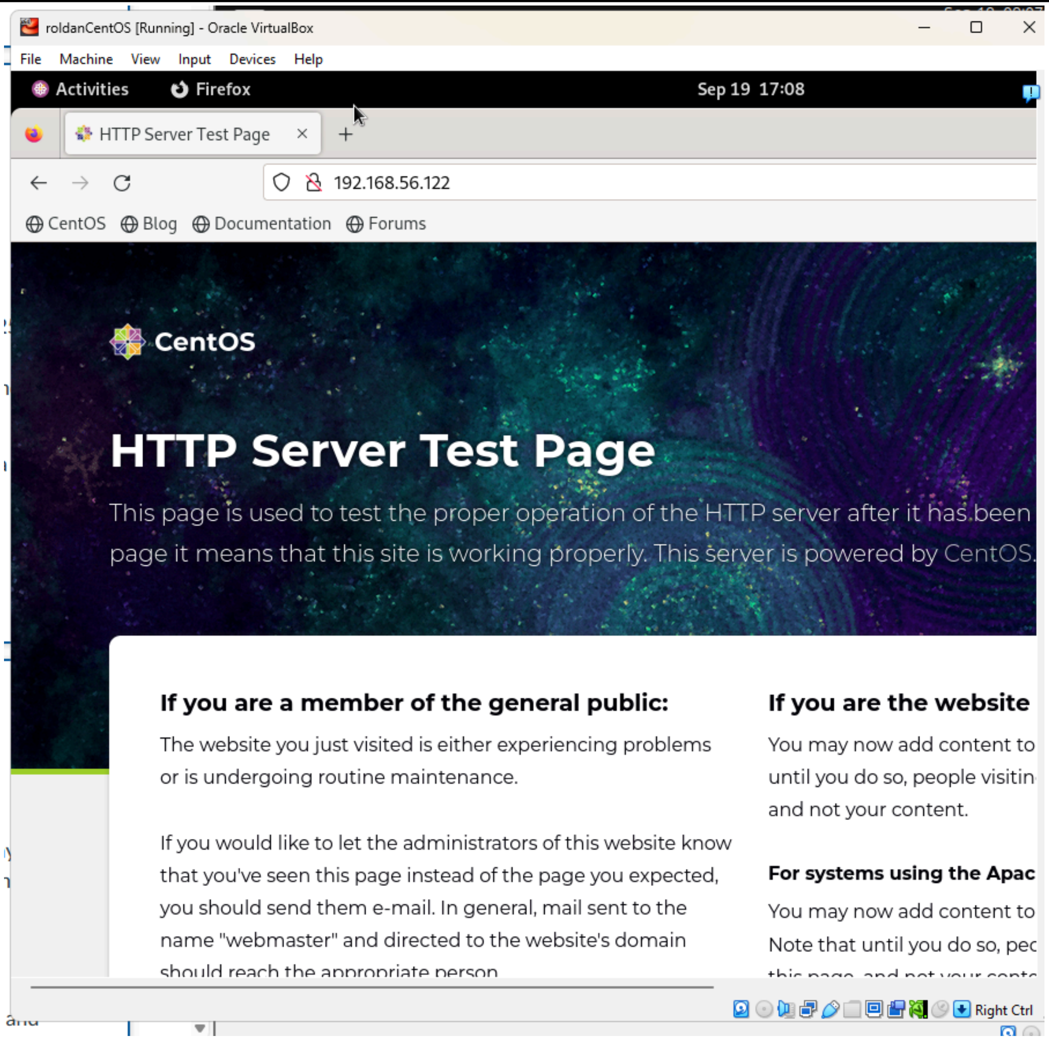
sudo firewall-cmd --add-port=80/tcp

(The result should be a success)



```
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 34  bytes 3368 (3.2 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 34  bytes 3368 (3.2 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

[jroldan@centos-server ~]$ 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: inactive (dead)
       Docs: man:httpd.service(8)
lines 1-6/6 (END)
[jroldan@centos-server ~]$ sudo systemctl start httpd
[sudo] password for jroldan:
[jroldan@centos-server ~]$ sudo firewall-cmd --add-port=80/tcp
success
[jroldan@centos-server ~]$ 
```

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)


roldanCentOS [Running] - Oracle VirtualBox

File Machine View Input Devices Help

Activities Firefox Sep 19 17:08

HTTP Server Test Page

192.168.56.122

CentOS Blog Documentation Forums

CentOS

HTTP Server Test Page

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

If you are the website

You may now add content to
until you do so, people visitin
and not your content.

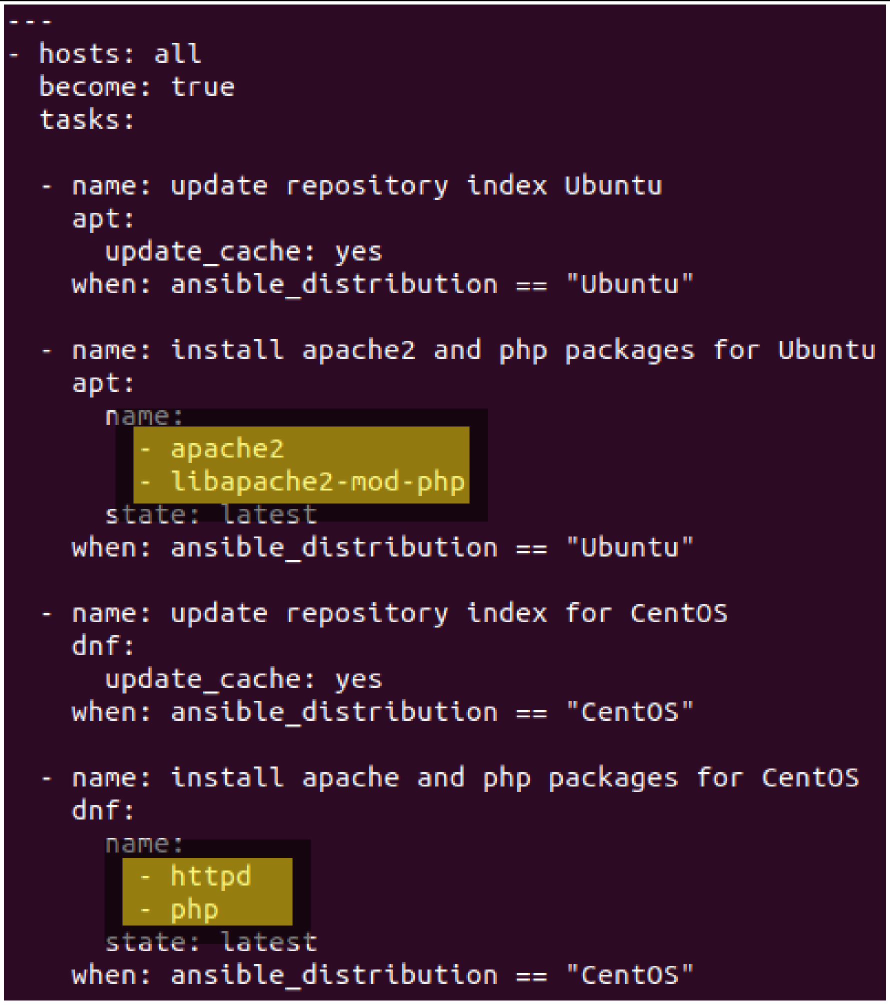
For systems using the Apac

You may now add content to
Note that until you do so, peo

Task 2: Refactoring playbook

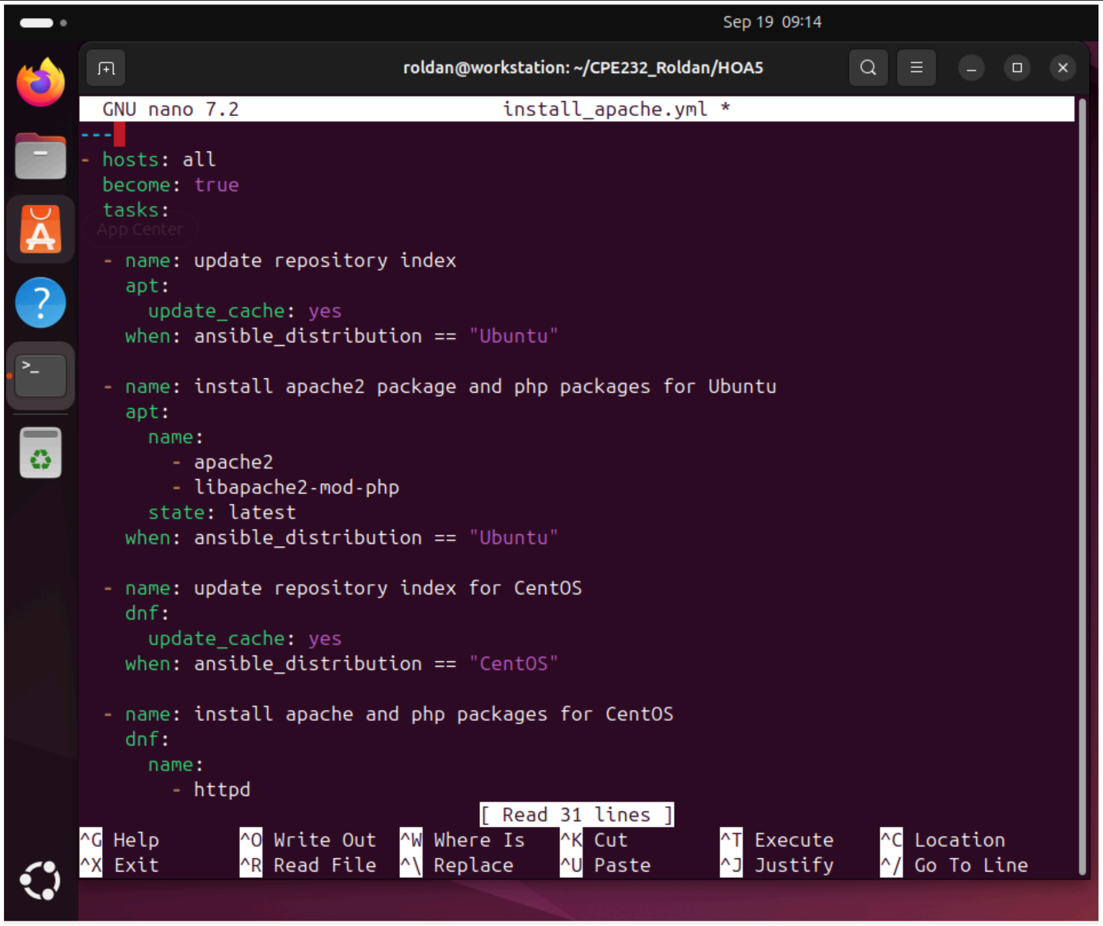
This time, we want to make sure that our playbook is efficient and that the codes are easier to read. This will also makes run ansible more quickly if it has to execute fewer tasks to do the same thing.

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:

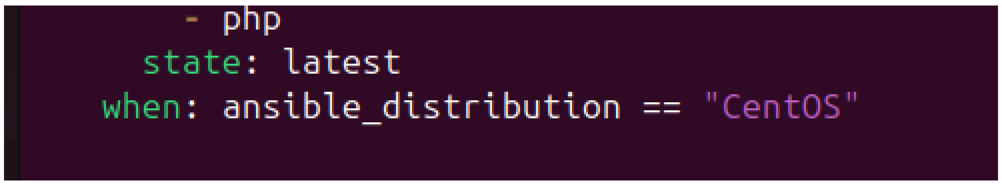


```
---
- hosts: all
  become: true
  tasks:

  - 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.

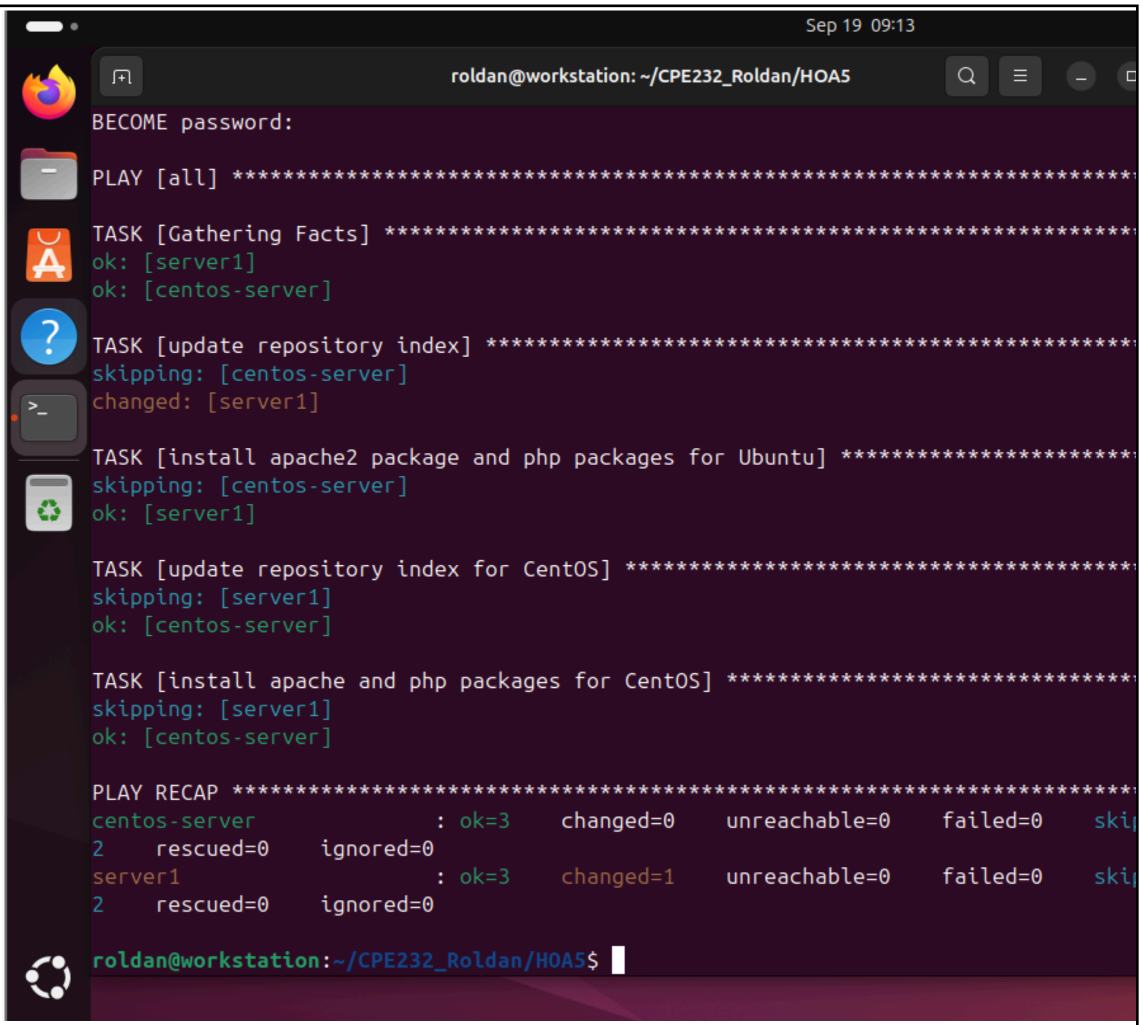


```
GNU nano 7.2                    install_apache.yml *
---
- hosts: all
  become: true
  tasks:

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

  - name: install apache2 package 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
                         [ Read 31 lines ]
^G Help       ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location
^X Exit       ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^/ Go To Line
```



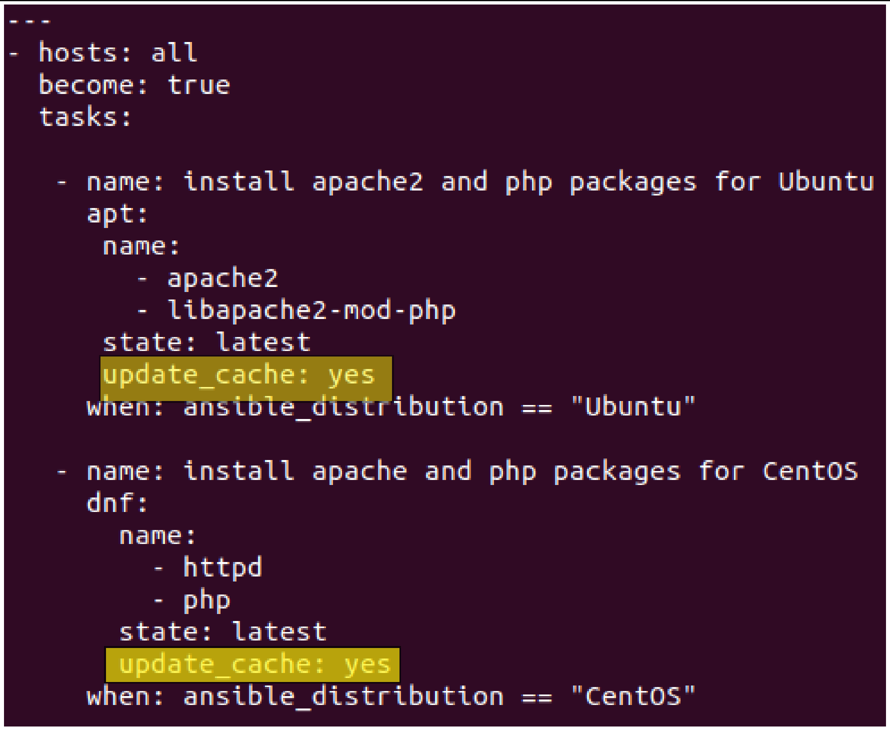
```
        - php
      state: latest
    when: ansible_distribution == "CentOS"
```

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



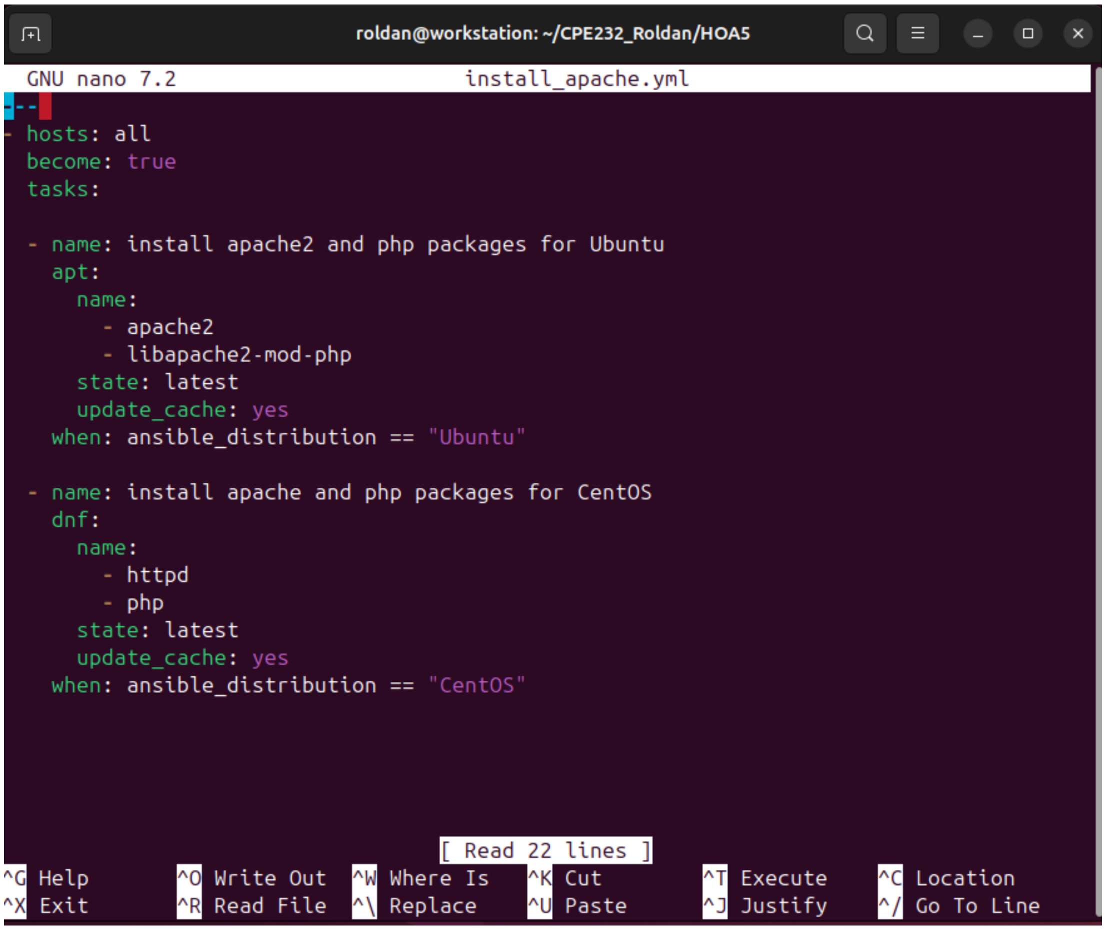
```
BECOME password:

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

TASK [Gathering Facts] *********************************************************
ok: [server1]
ok: [centos-server]

TASK [update repository index] *************************************************
skipping: [centos-server]
changed: [server1]

TASK [install apache2 package and php packages for Ubuntu] *********************
skipping: [centos-server]
ok: [server1]

TASK [update repository index for CentOS] **************************************
skipping: [server1]
ok: [centos-server]

TASK [install apache and php packages for CentOS] ******************************
skipping: [server1]
ok: [centos-server]

PLAY RECAP *********************************************************************
centos-server              : ok=3    changed=0    unreachable=0    failed=0    ski
2    rescued=0    ignored=0
server1                    : ok=3    changed=1    unreachable=0    failed=0    ski
2    rescued=0    ignored=0

roldan@workstation:~/CPE232_Roldan/HOA5$
```

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:



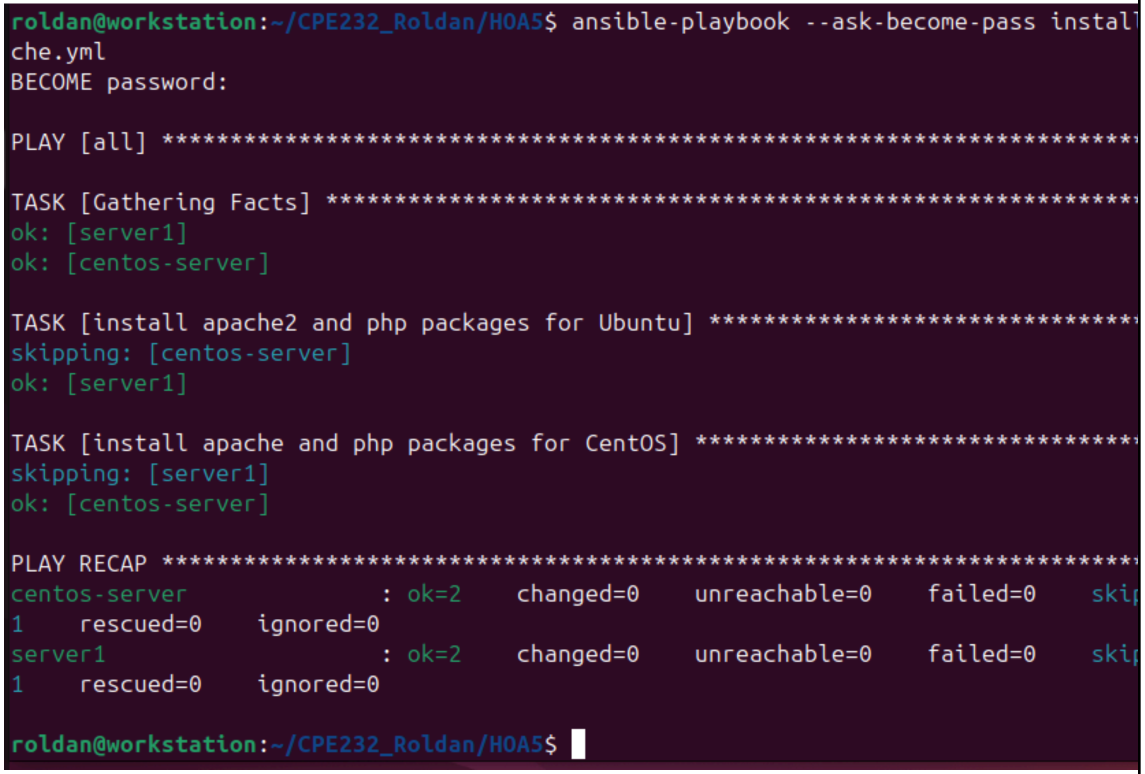
```
---
- hosts: all
  become: true
  tasks:

    - 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.



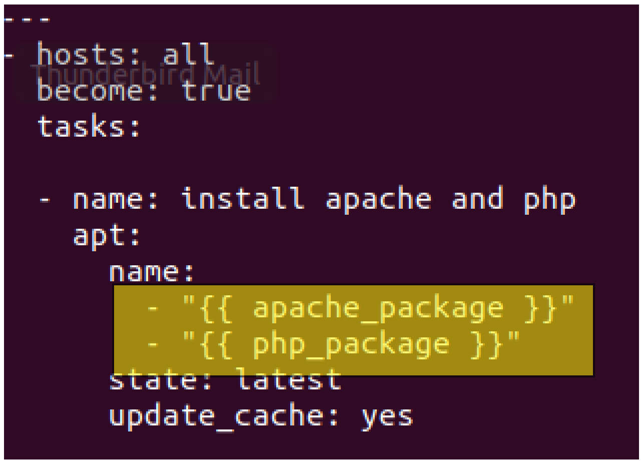
```
roldan@workstation: ~/CPE232_Roldan/HOA5

  GNU nano 7.2                         install_apache.yml
---
- hosts: all
  become: true
  tasks:

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

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

                                  [ Read 22 lines ]
^G Help          ^O Write Out   ^W Where Is    ^K Cut         ^T Execute     ^C Location
^X Exit          ^R Read File   ^\ Replace     ^U Paste       ^J Justify     ^/ Go To Line
```

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

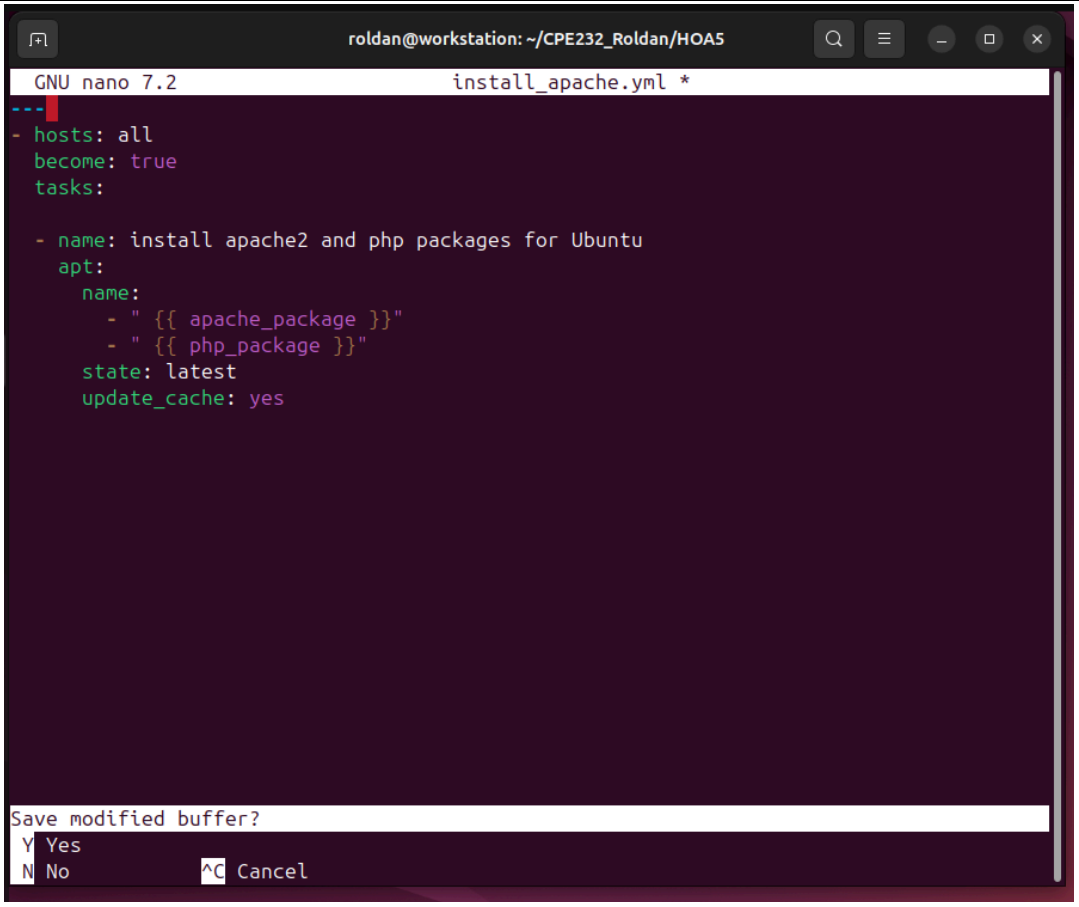


```
roldan@workstation:~/CPE232_Roldan/HOA5$ ansible-playbook --ask-become-pass install
che.yml
BECOME password:

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

TASK [Gathering Facts] *******************************************************
ok: [server1]
ok: [centos-server]

TASK [install apache2 and php packages for Ubuntu] ***************************
skipping: [centos-server]
ok: [server1]

TASK [install apache and php packages for CentOS] ****************************
skipping: [server1]
ok: [centos-server]

PLAY RECAP *******************************************************************
centos-server              : ok=2    changed=0    unreachable=0    failed=0    ski
1    rescued=0    ignored=0
server1                    : ok=2    changed=0    unreachable=0    failed=0    ski
1    rescued=0    ignored=0

roldan@workstation:~/CPE232_Roldan/HOA5$
```

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.



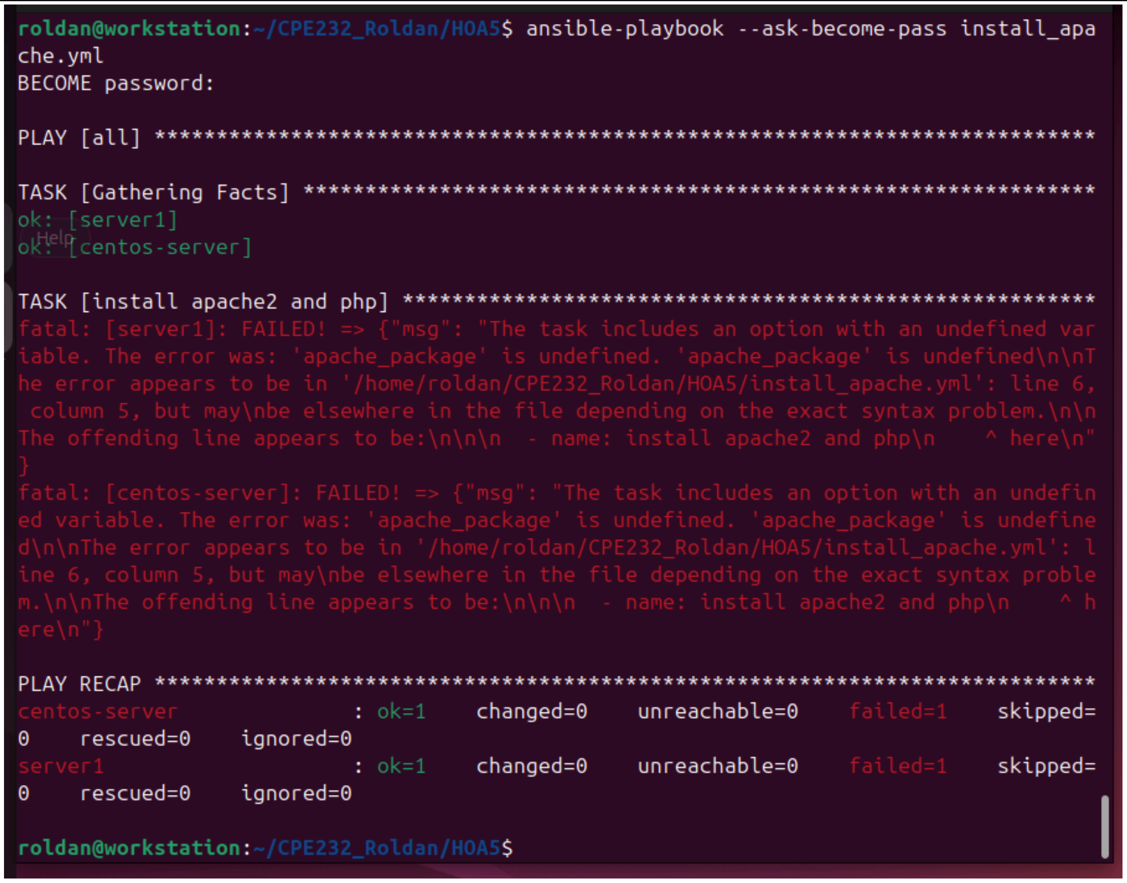
```
---
- hosts: all
  become: true
  tasks:

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



```
GNU nano 7.2                          install_apache.yml *
---
- hosts: all
  become: true
  tasks:

  - name: install apache2 and php packages for Ubuntu
    apt:
      name:
        - " {{ apache_package }}"
        - " {{ php_package }}"
      state: latest
      update_cache: yes
```

```
Save modified buffer?
 Y Yes
 N No           ^C Cancel
```

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

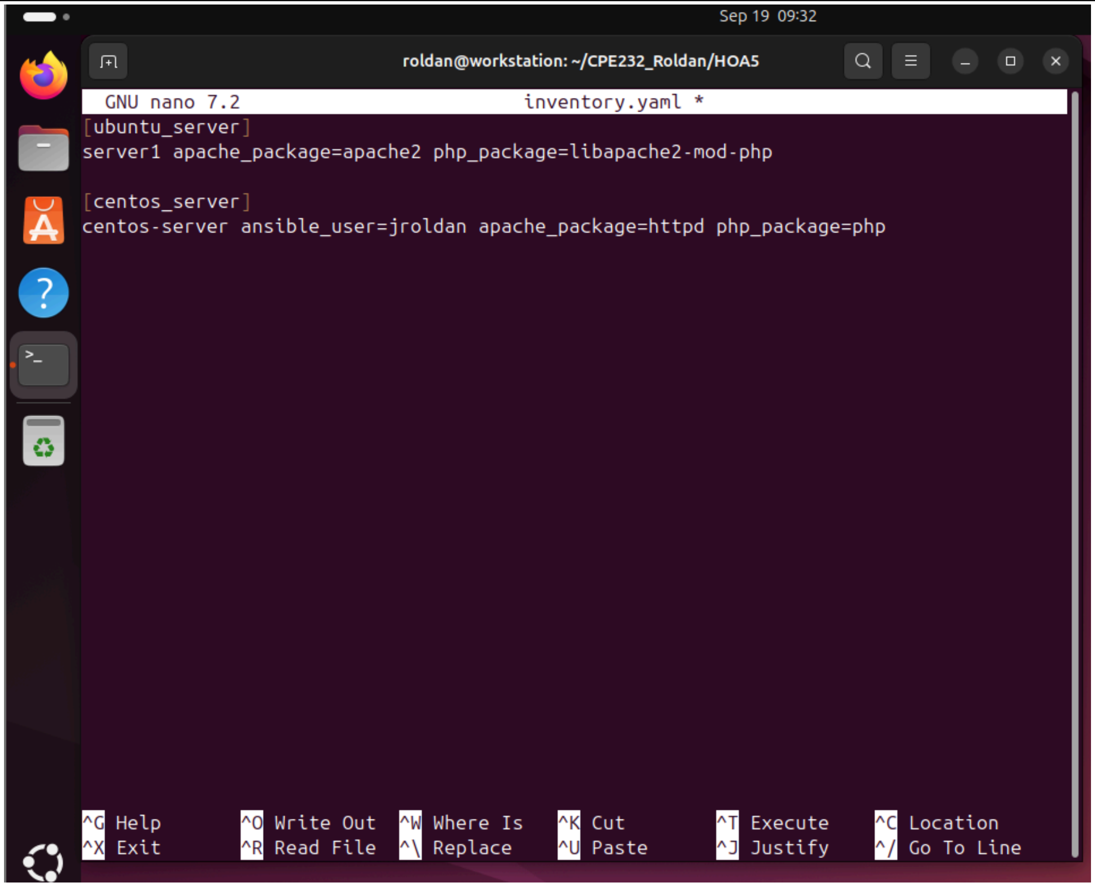


```
roldan@workstation:~/CPE232_Roldan/HOA5$ ansible-playbook --ask-become-pass install_apa
che.yml
BECOME password:

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

TASK [Gathering Facts] **************************************************************
ok: [server1]
ok: [centos-server]

TASK [install apache2 and php] ******************************************************
fatal: [server1]: FAILED! => {"msg": "The task includes an option with an undefined var
iable. The error was: 'apache_package' is undefined. 'apache_package' is undefined\n\nT
he error appears to be in '/home/roldan/CPE232_Roldan/HOA5/install_apache.yml': line 6,
 column 5, but may\nbe elsewhere in the file depending on the exact syntax problem.\n\n
The offending line appears to be:\n\n\n  - name: install apache2 and php\n    ^ here\n"
}
fatal: [centos-server]: FAILED! => {"msg": "The task includes an option with an undefin
ed variable. The error was: 'apache_package' is undefined. 'apache_package' is undefine
d\n\nThe error appears to be in '/home/roldan/CPE232_Roldan/HOA5/install_apache.yml': l
ine 6, column 5, but may\nbe elsewhere in the file depending on the exact syntax proble
m.\n\nThe offending line appears to be:\n\n\n  - name: install apache2 and php\n    ^ h
ere\n"}

PLAY RECAP **************************************************************************
centos-server              : ok=1    changed=0    unreachable=0    failed=1    skipped=
0    rescued=0    ignored=0
server1                    : ok=1    changed=0    unreachable=0    failed=1    skipped=
0    rescued=0    ignored=0

roldan@workstation:~/CPE232_Roldan/HOA5$
```

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:

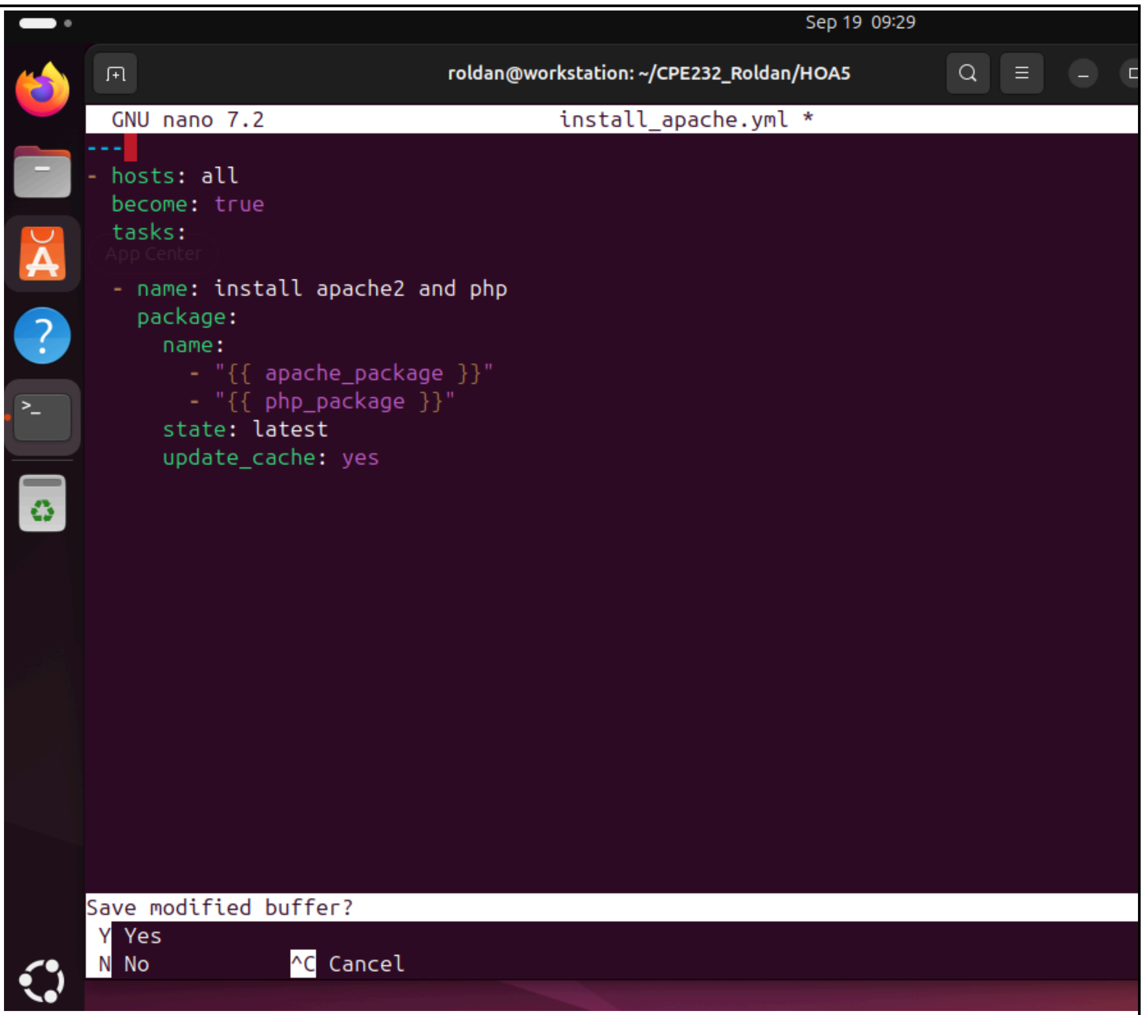
```
192.168.56.120 apache_package=apache2 php_package=libapache2-mod-php
192.168.56.121 apache_package=apache2 php_package=libapache2-mod-php
192.168.56.122 apache_package=httpd php_package=php
```

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



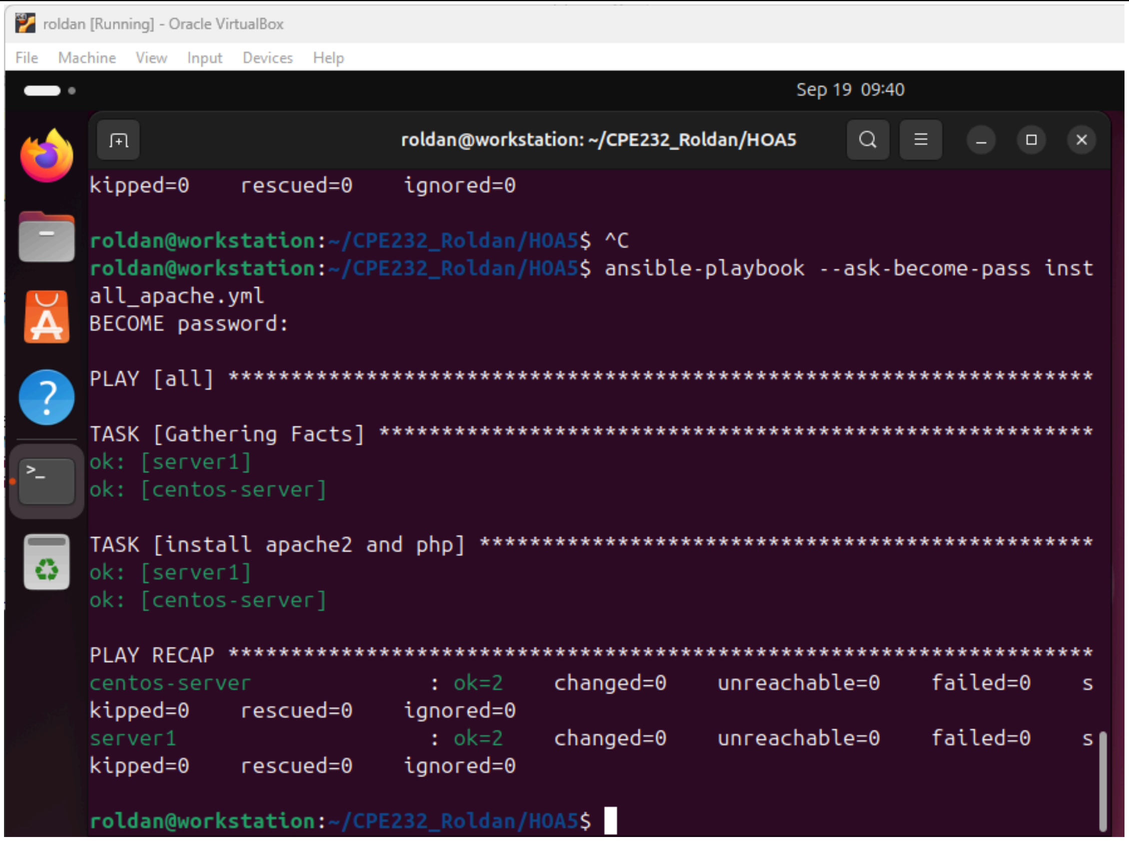
```
  GNU nano 7.2                     inventory.yaml *
[ubuntu_server]
server1 apache_package=apache2 php_package=libapache2-mod-php

[centos_server]
centos-server ansible_user=jroldan apache_package=httpd php_package=php
















^G Help        ^O Write Out   ^W Where Is    ^K Cut         ^T Execute     ^C Location
^X Exit        ^R Read File   ^\ Replace     ^U Paste       ^J Justify     ^/ Go To Line
```

Finally, we still have one more thing to change in our *install_apache.yml* file. In task 2.3, you may notice that the package is assign as apt, which will not run in CentOS. Replace the *apt* with *package*. Package is a module in ansible that is generic, which is going to use whatever package manager the underlying host or the target server uses. For Ubuntu it will automatically use *apt*, and for CentOS it will automatically use *dnf*. Make sure to save the file and exit. For more details about the ansible package, you may refer to this documentation: ansible.builtin.package – Generic OS package manager — Ansible Documentation



```
  GNU nano 7.2                          install_apache.yml *
---
- hosts: all
  become: true
  tasks:

  - name: install apache2 and php
    package:
      name:
        - "{{ apache_package }}"
        - "{{ php_package }}"
      state: latest
      update_cache: yes

Save modified buffer?
 Y Yes
 N No           ^C Cancel
```

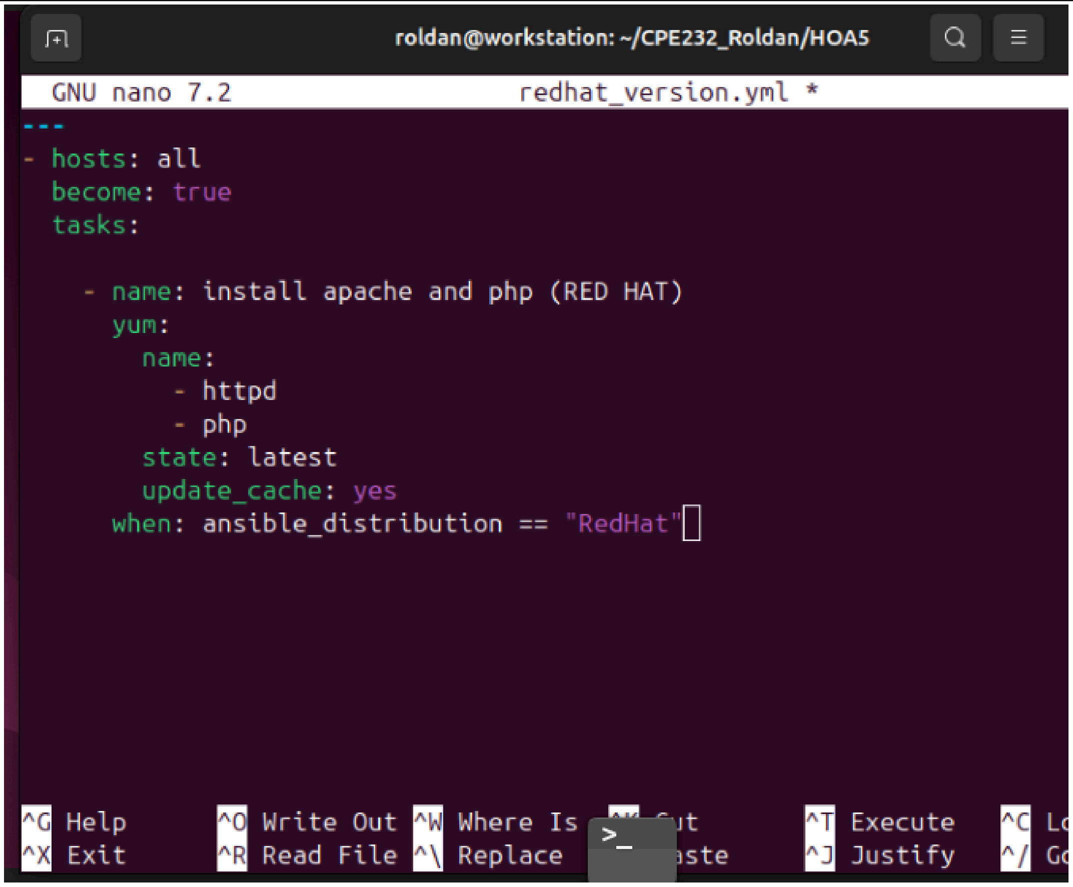
Run *ansible-playbook --ask-become-pass install_apache.yml* and describe the result.


roldan [Running] - Oracle VirtualBox
File Machine View Input Devices Help
Sep 19 09:40
roldan@workstation: ~/CPE232_Roldan/HOA5

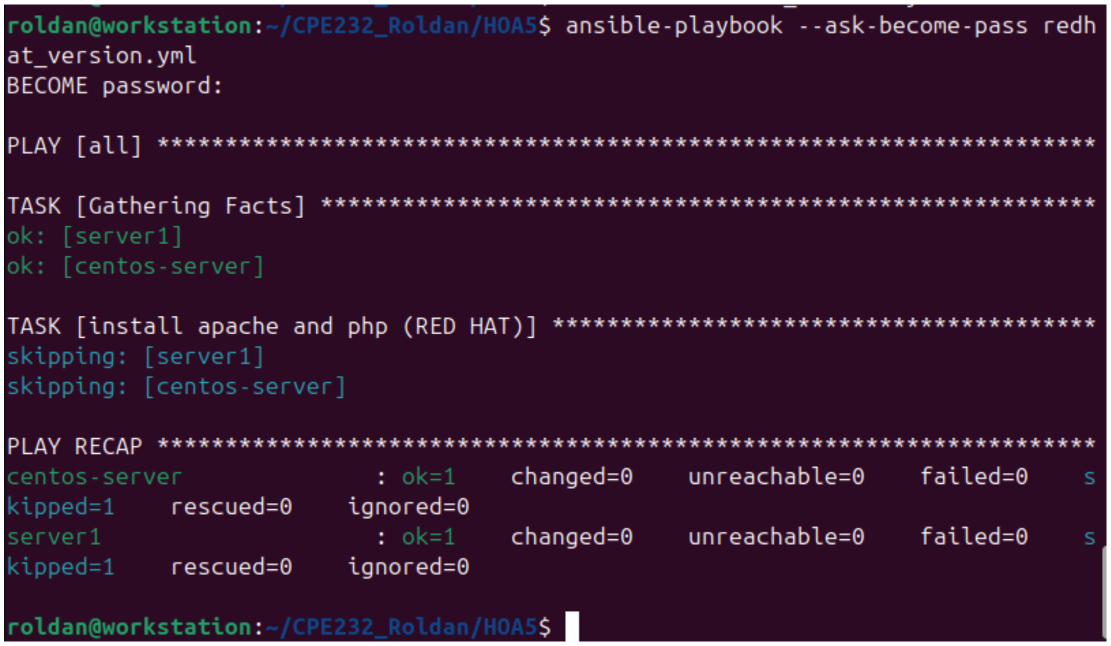
```
kipped=0    rescued=0    ignored=0

roldan@workstation:~/CPE232_Roldan/HOA5$ ^C
roldan@workstation:~/CPE232_Roldan/HOA5$ ansible-playbook --ask-become-pass inst
all_apache.yml
BECOME password:

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

TASK [Gathering Facts] *********************************************************
ok: [server1]
ok: [centos-server]

TASK [install apache2 and php] *************************************************
ok: [server1]
ok: [centos-server]

PLAY RECAP *********************************************************************
centos-server              : ok=2    changed=0    unreachable=0    failed=0    s
kipped=0    rescued=0    ignored=0
server1                    : ok=2    changed=0    unreachable=0    failed=0    s
kipped=0    rescued=0    ignored=0

roldan@workstation:~/CPE232_Roldan/HOA5$ 
```

Supplementary Activity:

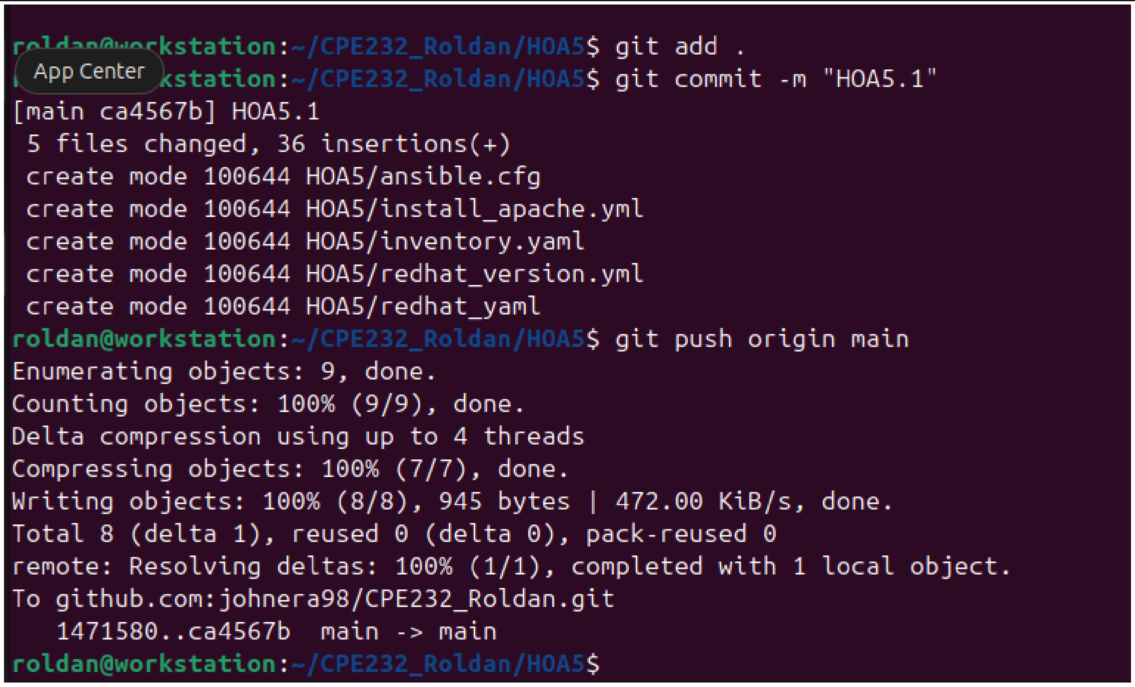
1. Create a playbook that could do the previous tasks in Red Hat OS.



```
roldan@workstation: ~/CPE232_Roldan/HOA5
  GNU nano 7.2                    redhat_version.yml *
---
- hosts: all
  become: true
  tasks:

    - name: install apache and php (RED HAT)
      yum:
        name:
          - httpd
          - php
        state: latest
        update_cache: yes
      when: ansible_distribution == "RedHat"
```



```
roldan@workstation:~/CPE232_Roldan/HOA5$ ansible-playbook --ask-become-pass redh
at_version.yml
BECOME password:

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

TASK [Gathering Facts] **********************************************************
ok: [server1]
ok: [centos-server]

TASK [install apache and php (RED HAT)] *****************************************
skipping: [server1]
skipping: [centos-server]

PLAY RECAP **********************************************************************
centos-server              : ok=1    changed=0    unreachable=0    failed=0    s
kipped=1    rescued=0    ignored=0
server1                    : ok=1    changed=0    unreachable=0    failed=0    s
kipped=1    rescued=0    ignored=0

roldan@workstation:~/CPE232_Roldan/HOA5$
```



```
roldan@workstation:~/CPE232_Roldan/HOA5$ git add .
roldan@workstation:~/CPE232_Roldan/HOA5$ git commit -m "HOA5.1"
[main ca4567b] HOA5.1
 5 files changed, 36 insertions(+)
 create mode 100644 HOA5/ansible.cfg
 create mode 100644 HOA5/install_apache.yml
 create mode 100644 HOA5/inventory.yaml
 create mode 100644 HOA5/redhat_version.yml
 create mode 100644 HOA5/redhat_yaml
roldan@workstation:~/CPE232_Roldan/HOA5$ git push origin main
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 4 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (8/8), 945 bytes | 472.00 KiB/s, done.
Total 8 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:johnera98/CPE232_Roldan.git
   1471580..ca4567b  main -> main
roldan@workstation:~/CPE232_Roldan/HOA5$
```

Reflections:

Answer the following:

1. Why do you think refactoring of playbook codes is important?
   - The refactoring of playbook code is valuable since it assists in maintaining the code to be clean, structured, and simple to comprehend. As refactoring your workspace means you can find what you want more easily and have a more productive time than getting distracted by mess, refactoring your playbook simplifies updating the playbook down the line. It is also useful in preventing mistakes by ensuring the code is made clearer and better, therefore not consuming time on removing problems that should have been avoided.

2. When do we use the "when" command in playbook?
   - When command We add conditions to tasks in a playbook through the use of the command when. It is as though you are saying, only do this, unless this particular situation is some other situation. To illustrate, in case you need to use specific servers or a specific value of a variable, when a variable equals a specific value, the when command can be used to heavily manage which task is run and it makes your playbook smarter and flexible.