Name: Andreu John L. Salvador	Date Performed: 22/09/2023
Course/Section: CPE31S5	Date Submitted: 26/09/2023
Instructor: Engr. Roman Richard	Semester and SY: 2023-2024 1st sem

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.

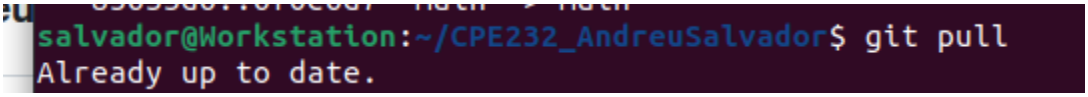
It is a best practice in ansible when you are working in a collaborative environment to use the command git pull. git pull is a Git command used to update the local version of a repository from a remote. By default, git pull does two things. Updates the current local working branch (currently checked 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.

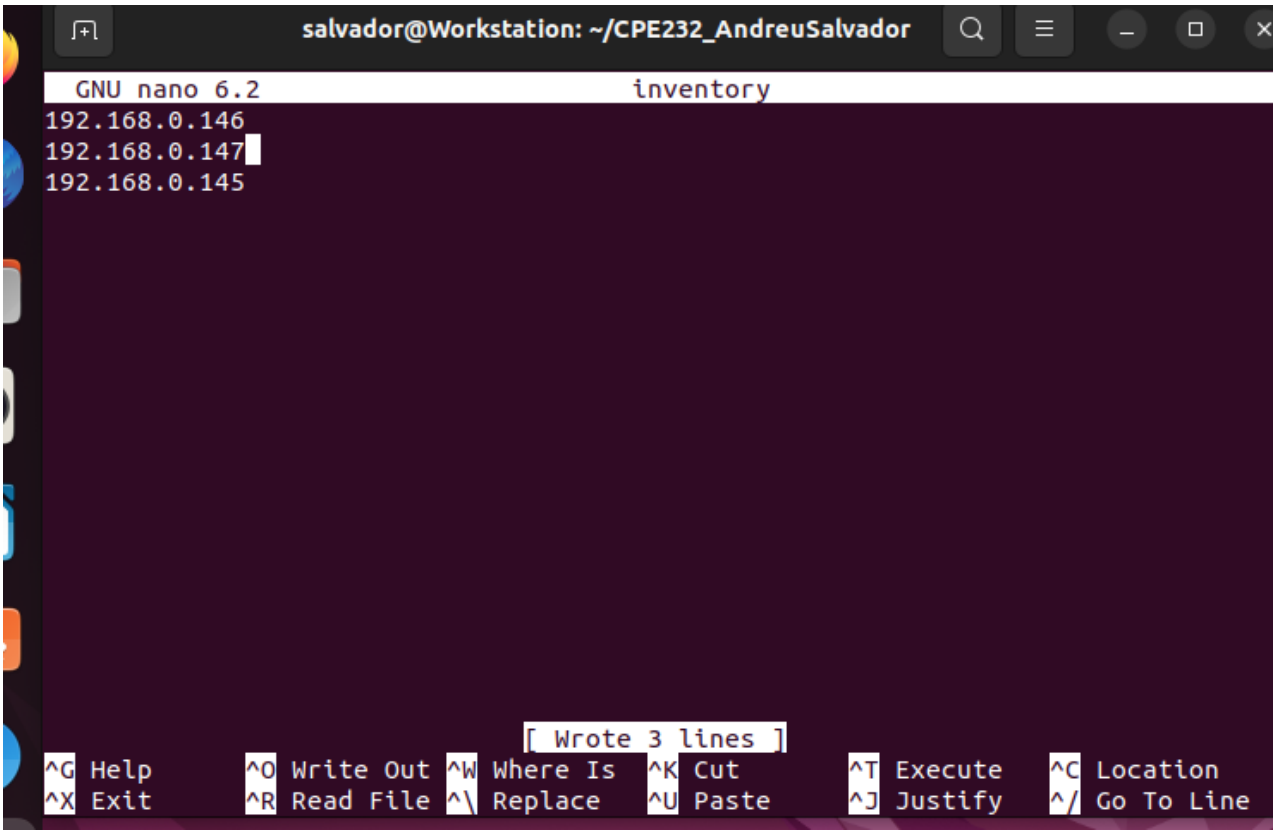
Task 1: Use when command for different distributions

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?



It says that it is already up to date. It means the git is already connected with my workstation.

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."

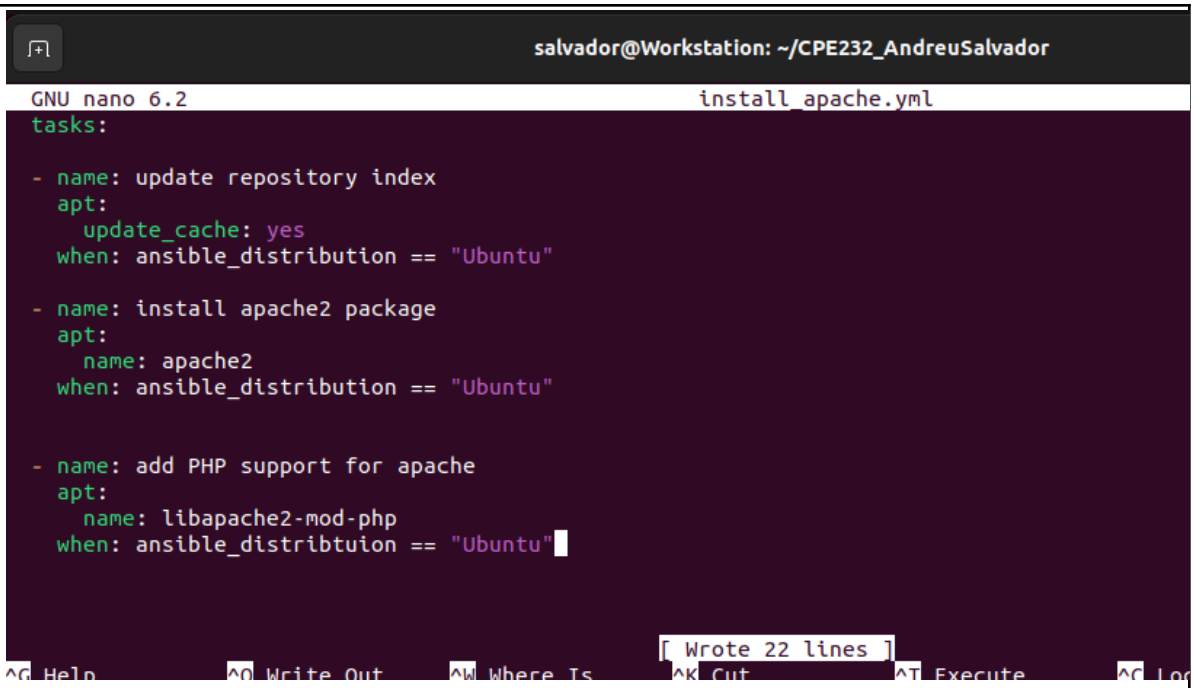


```
salvador@Workstation:~/CPE232_AndreuSalvador$ ansible-playbook -i inventory --ask-become-pass install_apache.yml
BECOME password:

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

TASK [Gathering Facts] *********************************************************************************************
The authenticity of host '192.168.0.146 (192.168.0.146)' can't be established.
ED25519 key fingerprint is SHA256:3Hfw5vuOnI0D+OpuV+naVNPpUjqNCNVEqZx6Wh08qSQ.
This host key is known by the following other names/addresses:
    ~/.ssh/known_hosts:5: [hashed name]
    ~/.ssh/known_hosts:14: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
ok: [192.168.0.145]
ok: [192.168.0.146]
fatal: [192.168.0.147]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to the host via ssh: ssh: connect to host 192.
68.0.147 port 22: Connection timed out", "unreachable": true}

TASK [update repository index] *************************************************************************************
[WARNING]: Updating cache and auto-installing missing dependency: python-apt
fatal: [192.168.0.145]: FAILED! => {"changed": false, "cmd": "apt-get update", "msg": "[Errno 2] No such file or directory", "rc":
}
changed: [192.168.0.146]

TASK [install apache2 package] *************************************************************************************

changed: [192.168.0.146]

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

changed: [192.168.0.146]

PLAY RECAP *********************************************************************************************************
192.168.0.145              : ok=1    changed=0    unreachable=0    failed=1    skipped=0    rescued=0    ignored=0
192.168.0.146              : ok=4    changed=3    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
192.168.0.147              : ok=0    changed=0    unreachable=1    failed=0    skipped=0    rescued=0    ignored=0
```

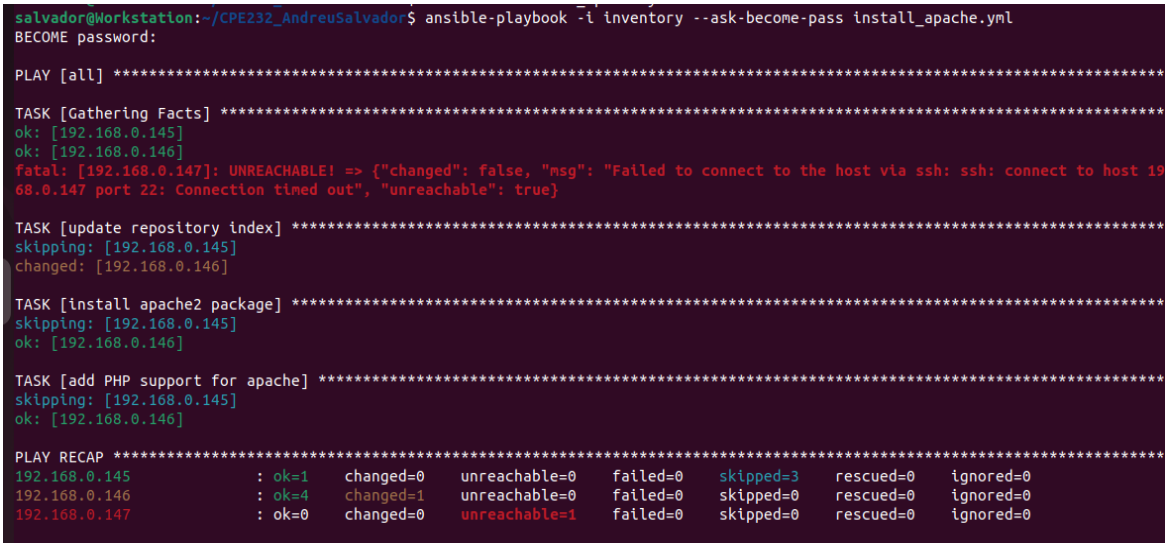
3. Edit the *install_apache.yml* file and insert the lines shown below.

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

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

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

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

Make sure to save the file and exit.



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



The process was successful, it says that there is 1 change within the server which is the CentOS. the 3 commands inside the apache.yml was also successful showing the feedback of ok for the 127.0.0.1 server.
If you have a mix of Debian and Ubuntu servers, you can change the configuration of your playbook like this.

- name: update repository index

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

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

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

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

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

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

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

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

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

  - name: add PHP support for apache
    dnf:
      name: php
      state: latest
    when: ansible_distribution == "CentOS"
```

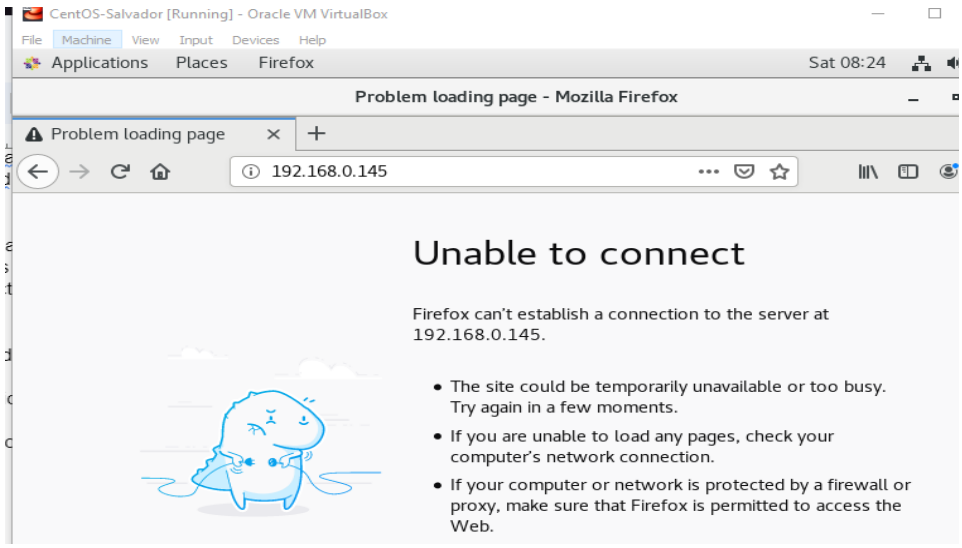
Make sure to save and exit.

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

```
salvador@Workstation:~/CPE232_AndreuSalvador$ ansible-playbook -i inventory --ask-become-pass install_apache.yml
BECOME password:

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

TASK [Gathering Facts] *********************************************************************************************************
ok: [192.168.0.145]
ok: [192.168.0.147]
fatal: [192.168.0.146]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to the host via ssh: ssh: connect to host 19
68.0.146 port 22: Connection timed out", "unreachable": true}

TASK [update repository index] *************************************************************************************************
skipping: [192.168.0.145]
changed: [192.168.0.147]

TASK [install apache2 package] *************************************************************************************************
skipping: [192.168.0.145]
ok: [192.168.0.147]

TASK [add PHP support for apache] **********************************************************************************************
skipping: [192.168.0.145]
ok: [192.168.0.147]

TASK [update repository index] *************************************************************************************************
skipping: [192.168.0.147]
ok: [192.168.0.145]

TASK [install apache2 package] *************************************************************************************************
skipping: [192.168.0.147]
ok: [192.168.0.145]

TASK [add PHP support for apache] **********************************************************************************************
skipping: [192.168.0.147]
ok: [192.168.0.145]

PLAY RECAP *********************************************************************************************************************
192.168.0.145              : ok=4    changed=0    unreachable=0    failed=0    skipped=3    rescued=0    ignored=0
192.168.0.146              : ok=0    changed=0    unreachable=1    failed=0    skipped=0    rescued=0    ignored=0
192.168.0.147              : ok=4    changed=1    unreachable=0    failed=0    skipped=3    rescued=0    ignored=0
```

I switched the servers here since my computer cant run everything all at the same time. the codes inside the playbook did changed or some changes in the server resulting into a ok =4 in the log.

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.

```
[salvador@salvadorcentos ~]$ sudo systemctl status httpd
[sudo] password for salvador:
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; vendor preset: disa
bled)
   Active: inactive (dead)
     Docs: man:httpd(8)
           man:apachectl(8)
[salvador@salvadorcentos ~]$
```

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)

```
salvador@salvadorcentos ~]$ sudo systemctl start httpd
salvador@salvadorcentos ~]$ sudo firewall-cmd --add-port=80/tcp
uccess
salvador@salvadorcentos ~]$
```

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)

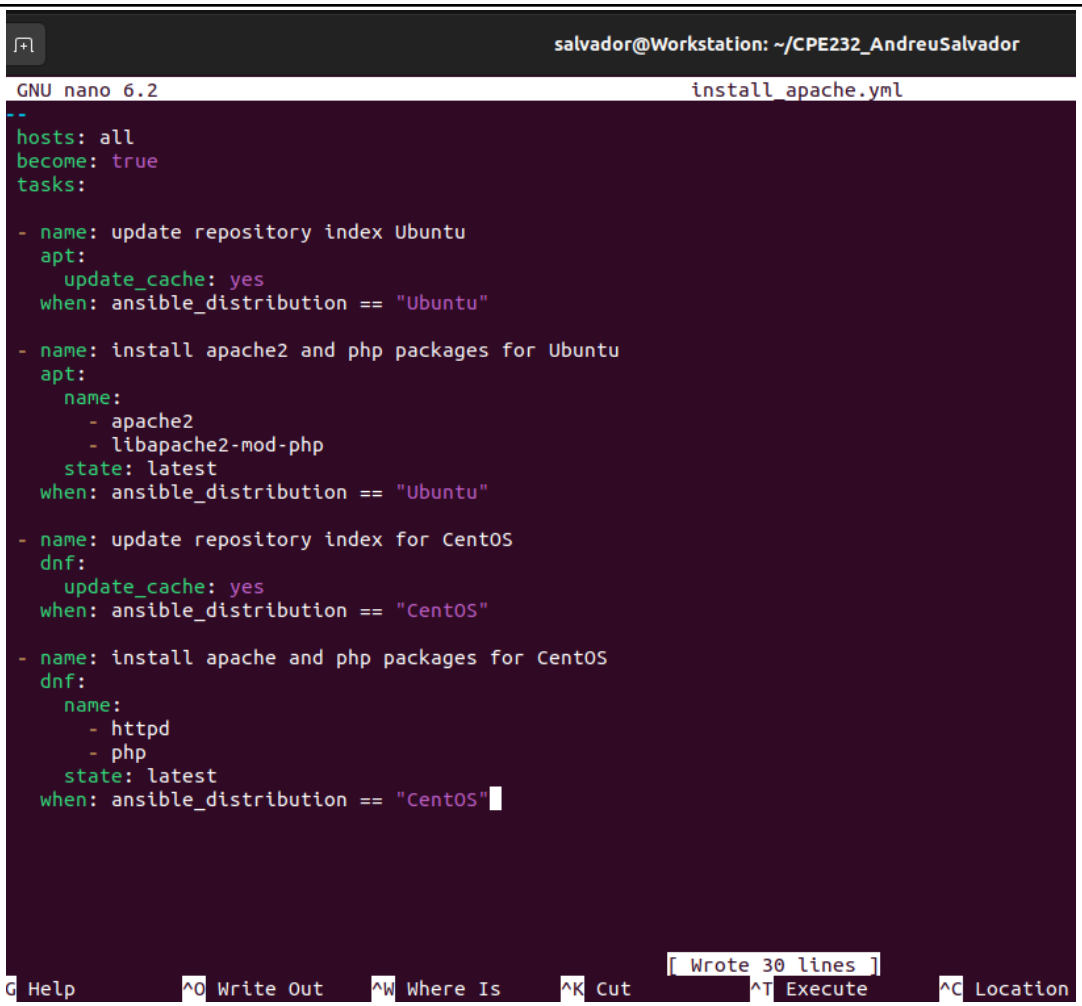
Task 2: Refactoring playbook

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.



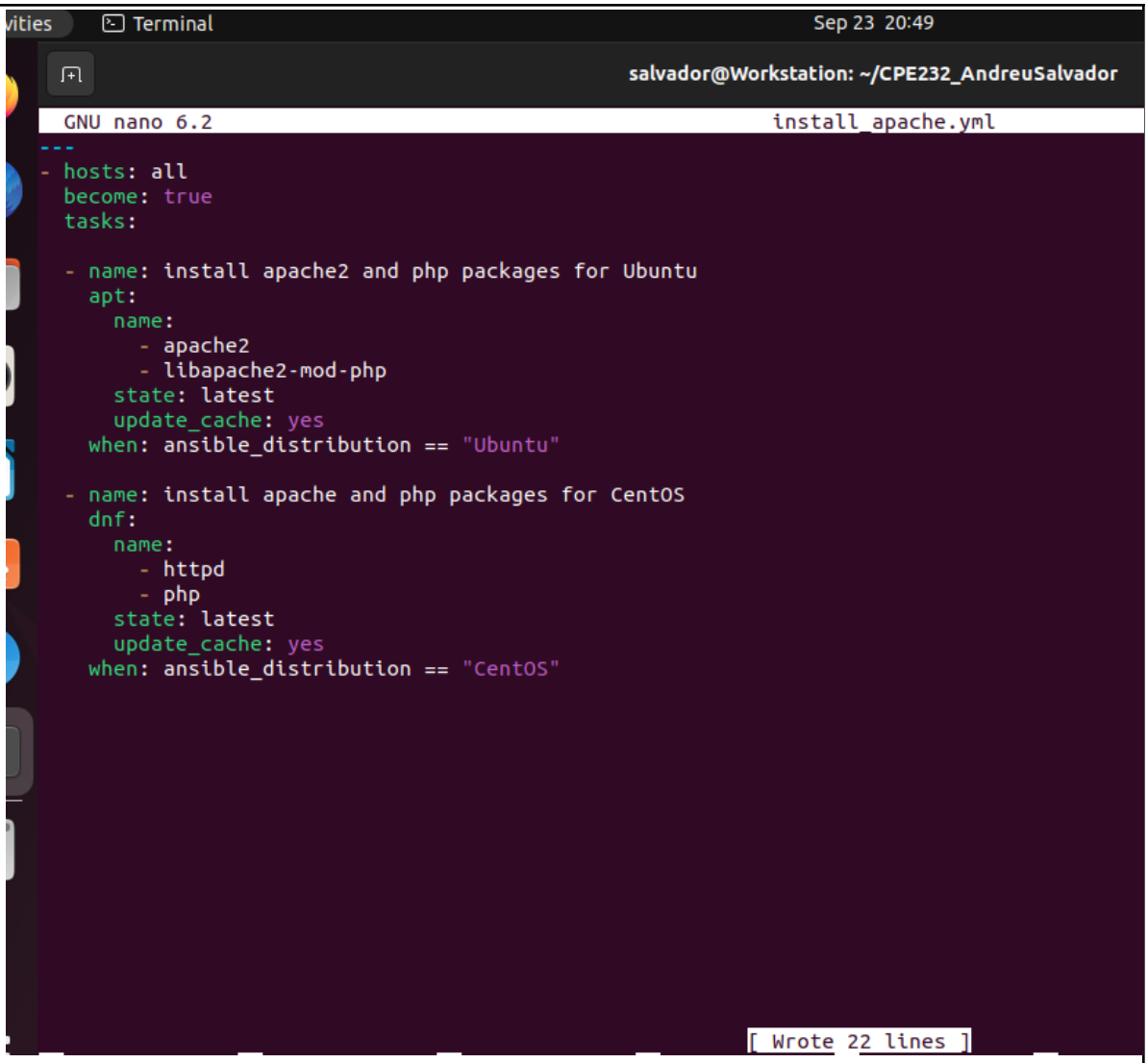
Run *ansible-playbook --ask-become-pass install_apache.yml* and describe the result.

```
salvador@Workstation:~/CPE232_AndreuSalvador$ ansible-playbook -i inventory --ask-become-pass install_apache.yml
BECOME password:

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

TASK [Gathering Facts] *****************************************************************************************************
ok: [192.168.0.145]
ok: [192.168.0.147]
fatal: [192.168.0.146]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to the host via ssh: ssh: connect to h
68.0.146 port 22: Connection timed out", "unreachable": true}

TASK [update repository index Ubuntu] **************************************************************************************
skipping: [192.168.0.145]
changed: [192.168.0.147]

TASK [install apache2 and php packages for Ubuntu] *************************************************************************
skipping: [192.168.0.145]
ok: [192.168.0.147]

TASK [update repository index for CentOS] **********************************************************************************
skipping: [192.168.0.147]
ok: [192.168.0.145]

TASK [install apache and php packages for CentOS] **************************************************************************
skipping: [192.168.0.147]
ok: [192.168.0.145]

PLAY RECAP *****************************************************************************************************************
192.168.0.145              : ok=3    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
192.168.0.146              : ok=0    changed=0    unreachable=1    failed=0    skipped=0    rescued=0    ignored=0
192.168.0.147              : ok=3    changed=1    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
```

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.

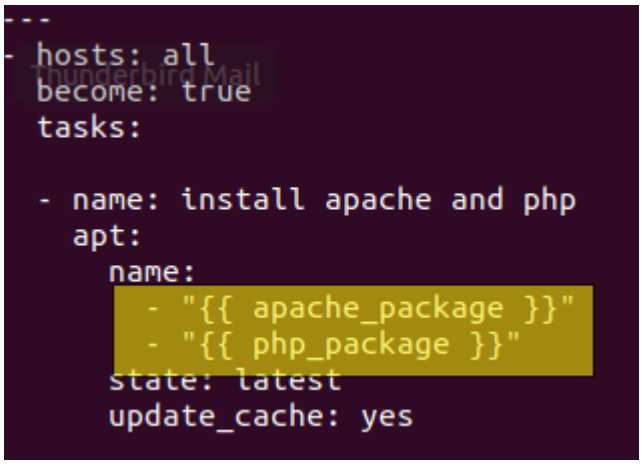


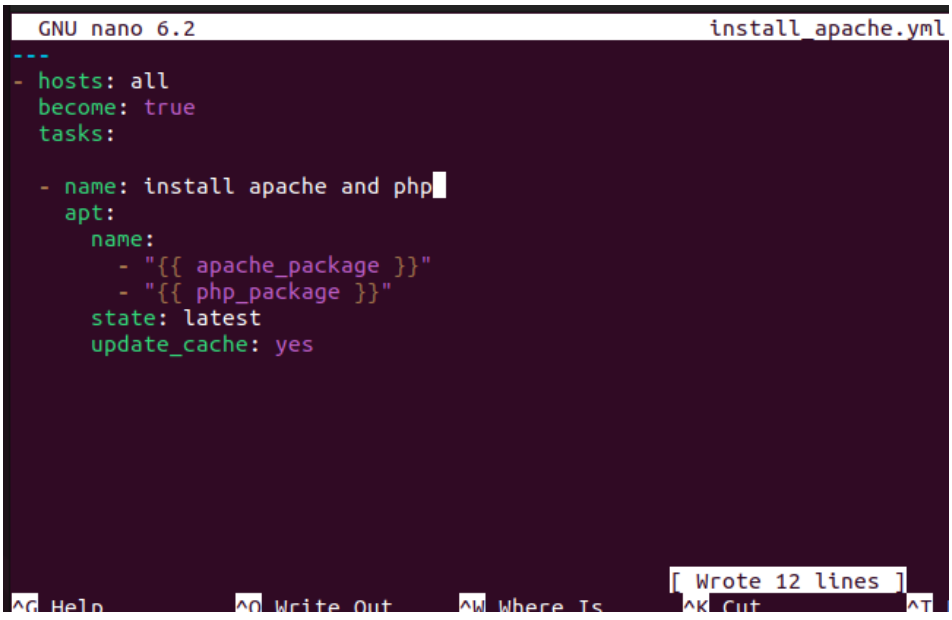
Run *ansible-playbook --ask-become-pass install_apache.yml* and describe the result.

```
salvador@Workstation:~/CPE232_AndreuSalvador$ ansible-playbook -i inventory --ask-become-pass install_apache.yml
BECOME password:

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

TASK [Gathering Facts] *****************************************************************************************************
ok: [192.168.0.145]
ok: [192.168.0.147]
fatal: [192.168.0.146]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to the host via ssh: ssh: connect to host 192.1
68.0.146 port 22: Connection timed out", "unreachable": true}

TASK [install apache2 and php packages for Ubuntu] *************************************************************************
skipping: [192.168.0.145]
ok: [192.168.0.147]

TASK [install apache and php packages for CentOS] **************************************************************************
skipping: [192.168.0.147]
ok: [192.168.0.145]

PLAY RECAP *****************************************************************************************************************
192.168.0.145              : ok=2    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
192.168.0.146              : ok=0    changed=0    unreachable=1    failed=0    skipped=0    rescued=0    ignored=0
192.168.0.147              : ok=2    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
```

the recap says that there are 2 ok means the run was a success.

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.





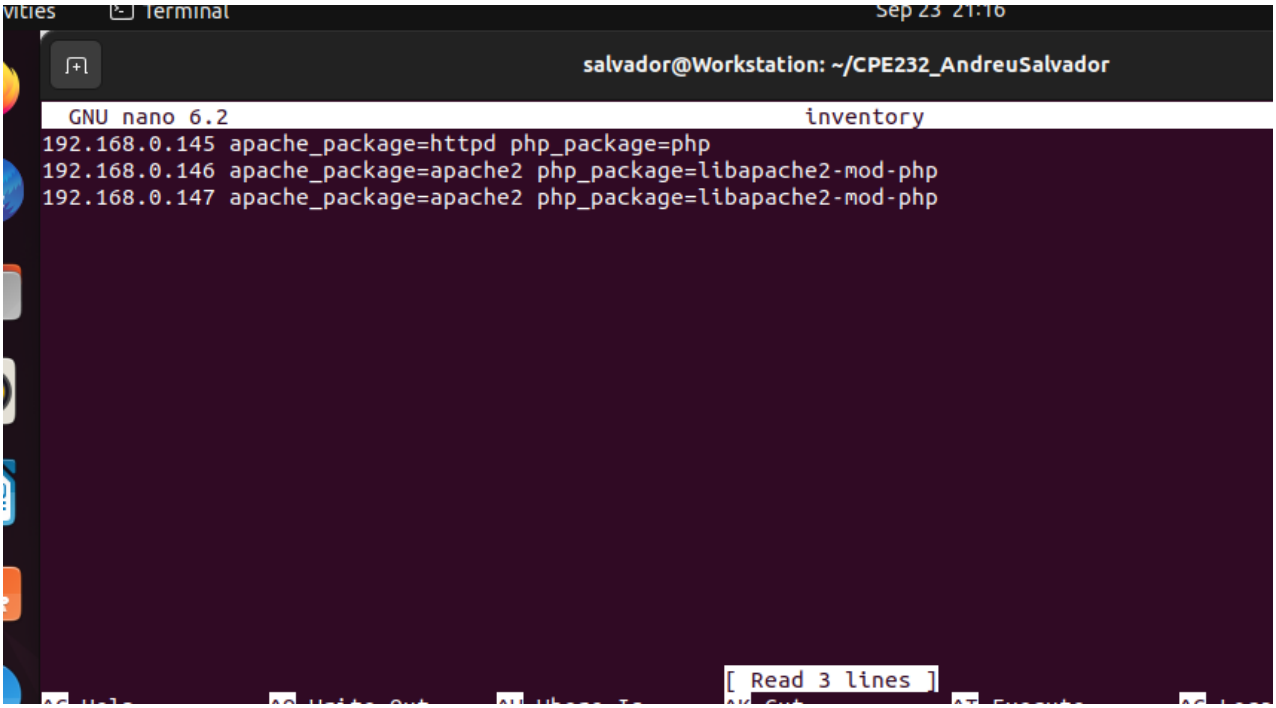
Run *ansible-playbook --ask-become-pass install_apache.yml* and describe the result.

```
salvador@Workstation:~/CPE232_AndreuSalvador$ ansible-playbook -i inventory --ask-become-pass install_apache.yml
BECOME password:

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

TASK [Gathering Facts] *****************************************************************************************
ok: [192.168.0.147]
ok: [192.168.0.145]
fatal: [192.168.0.146]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to the host via ssh: ssh: connect to host 192.1
68.0.146 port 22: Connection timed out", "unreachable": true}

TASK [install apache and php] **********************************************************************************
fatal: [192.168.0.145]: FAILED! => {"msg": "The task includes an option with an undefined variable. The error was: 'apache_package'
is undefined\n\nThe error appears to be in '/home/salvador/CPE232_AndreuSalvador/install_apache.yml': line 6, column 5, but may\nbe
elsewhere in the file depending on the exact syntax problem.\n\nThe offending line appears to be:\n\n\n  - name: install apache and
php\n    ^ here\n"}
fatal: [192.168.0.147]: FAILED! => {"msg": "The task includes an option with an undefined variable. The error was: 'apache_package'
is undefined\n\nThe error appears to be in '/home/salvador/CPE232_AndreuSalvador/install_apache.yml': line 6, column 5, but may\nbe
elsewhere in the file depending on the exact syntax problem.\n\nThe offending line appears to be:\n\n\n  - name: install apache and
php\n    ^ here\n"}

PLAY RECAP *****************************************************************************************************
192.168.0.145              : ok=1    changed=0    unreachable=0    failed=1    skipped=0    rescued=0    ignored=0
192.168.0.146              : ok=0    changed=0    unreachable=1    failed=0    skipped=0    rescued=0    ignored=0
192.168.0.147              : ok=1    changed=0    unreachable=0    failed=1    skipped=0    rescued=0    ignored=0
```

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.



192.168.0.145 is my CentOS server.

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 6.2                                  install_apache.yml *
---
- hosts: all
  become: true
  tasks:

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

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

```
ties  Terminal                    Sep 23 21:15
salvador@Workstation: ~/CPE232_AndreuSalvador
192.168.0.146              : ok=0    changed=0    unreachable=1    failed=0    skipped=0    rescued=0    ignored=0

salvador@Workstation:~/CPE232_AndreuSalvador$ sudo nano inventory
salvador@Workstation:~/CPE232_AndreuSalvador$ ansible-playbook -i inventory --ask-become-pass install_apache.yml
BECOME password:

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

TASK [Gathering Facts] *********************************************************************************************
ok: [192.168.0.145]
ok: [192.168.0.147]
fatal: [192.168.0.146]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to the host via ssh: ssh: connect
to host 192.168.0.146 port 22: Connection timed out", "unreachable": true}

TASK [install apache and php] **************************************************************************************
ok: [192.168.0.147]
ok: [192.168.0.145]

PLAY RECAP *********************************************************************************************************
192.168.0.145              : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
192.168.0.146              : ok=0    changed=0    unreachable=1    failed=0    skipped=0    rescued=0    ignored=0
192.168.0.147              : 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.

First, I grouped the CentOS in the hosts directory of the ansible so that i can easily access it. I named the group as Server.

```
  GNU nano 6.2                                    /etc/ansible/hosts
[localhost]
127.0.0.1 ansible_connection=local

[Server]
192.168.0.145
```

Just like in the previous codes, I wrote the code that will update the repository index and install apache package in CentOS. i put package instead of apt or dnf since this code will automatically know which Operating system the server is wherein it will identify the OS as a dnf if it is a CentOS and apt if it is Ubuntu or others.

```
  GNU nano 6.2                                    supple.yml
---
- hosts: Server
  become: true
  tasks:

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

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

The result shows that the CentOS server has already installed the apache2 package and the repository was already updated hence showing an ok in the play recap.

```
salvador@Workstation:~/CPE232_AndreuSalvador$ ansible-playbook --ask-become-pass supple.yml
BECOME password:

PLAY [Server] ******************************************************************************************************

TASK [Gathering Facts] *********************************************************************************************
ok: [192.168.0.145]

TASK [update repository index] *************************************************************************************
ok: [192.168.0.145]

TASK [install apache2 package] *************************************************************************************
ok: [192.168.0.145]

PLAY RECAP *********************************************************************************************************
192.168.0.145              : ok=3    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

Reflections:

Answer the following:

1. Why do you think refactoring of playbook codes is important?

Its significance is needed not just for programmers but for other people to see. Refactoring of the playbook codes will let us read it more easily or the readability of the codes will be much easier and less bugs and confusion will be developed so that our codes won't be complex to understand.

2. When do we use the “when” command in playbook?

we use when for platform specific tasks. to identify the OS and not construct a invalid argument within the playbook the when is used, for example is when installing a package for a certain OS and you have a different OS with each of the hosts included in your playbook. this will give you no running error and the process will not result in a invalid code.