Name: ROALLOS, Jean Gabriel Vincent G.	**Date Performed:** 08 / 26 / 2025
Course/Section: CPE212 - CPE31S2	**Date Submitted:** 09 / 05 / 2025
Instructor: Engr. Robin Valenzuela	**Semester & SY:** 1st Sem, 2025 - 2026

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.

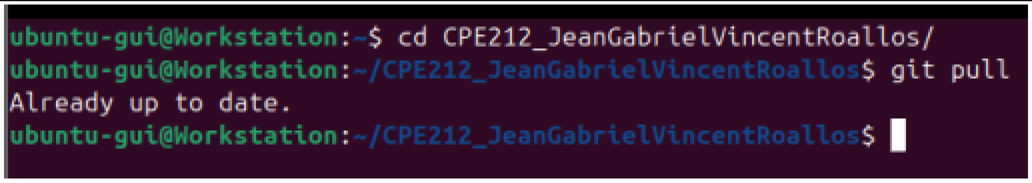
It is a best practice in ansible when you are working in a collaborative environment to use the command git pull. git pull is a Git command used to update the local version of a repository from a remote. By default, git pull does two things. Updates the current local working branch (currently checked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

No CentOS Stream 9 at the time of executing the HOA

Task 1: Use when command for different distributions

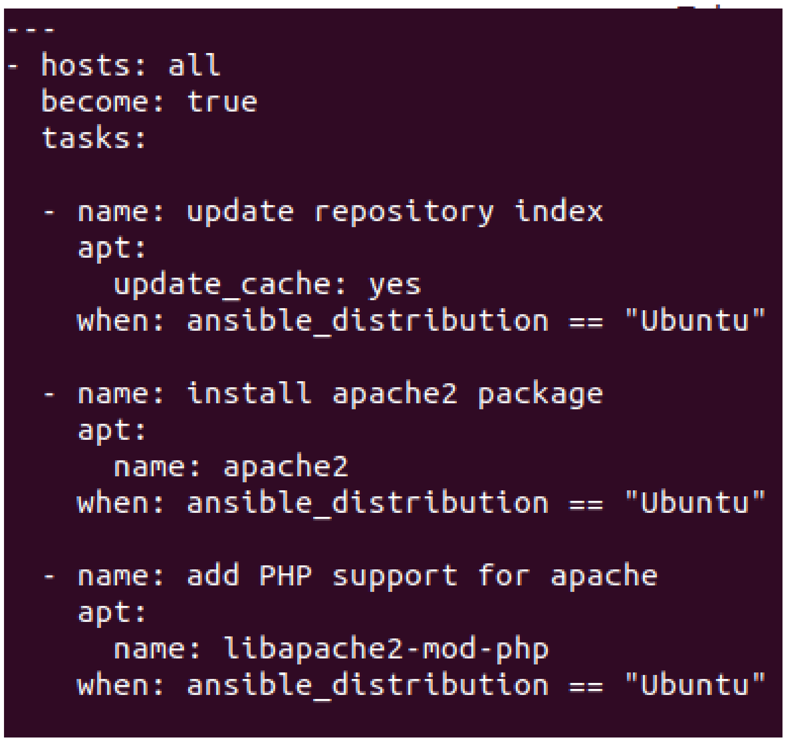
1. In the local machine, make sure you are in the local repository directory (*CPE232_yourname*). Issue the command git pull. When prompted, enter the correct passphrase or password. Describe what happened when you issue this command. Did something happen? Why?



```
ubuntu-gui@Workstation:~$ cd CPE212_JeanGabrielVincentRoallos/
ubuntu-gui@Workstation:~/CPE212_JeanGabrielVincentRoallos$ git pull
Already up to date.
ubuntu-gui@Workstation:~/CPE212_JeanGabrielVincentRoallos$ 
```

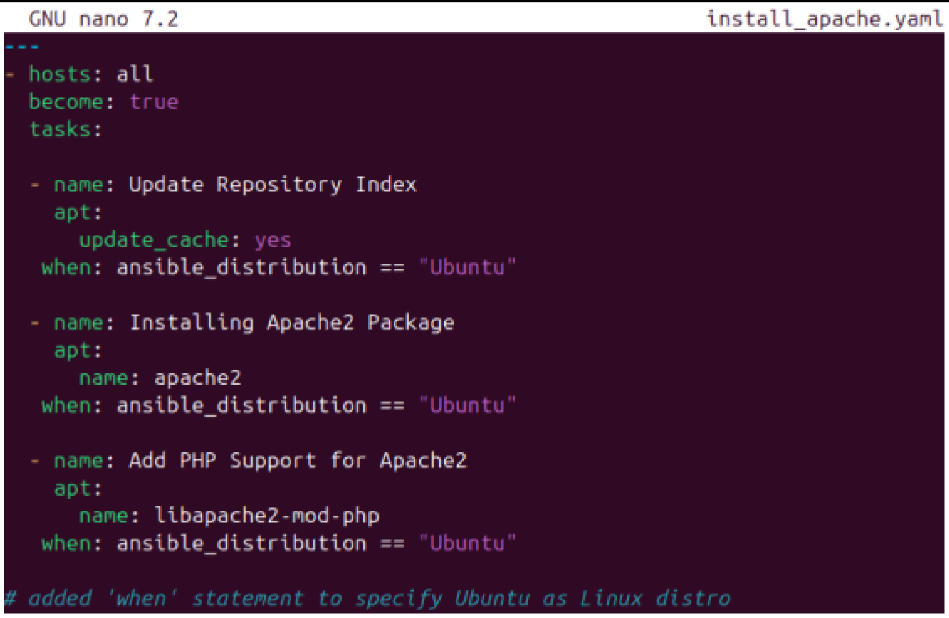
2. Edit the inventory file and add the IP address of the Centos VM. Issue the command we used to execute the playbook (the one we used in the last activity): *ansible-playbook --ask-become-pass install_apache.yml*. After executing this command, you may notice that it did not become successful in the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



```
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: Installing Apache2 Package
    apt:
      name: apache2
   when: ansible_distribution == "Ubuntu"

  - name: Add PHP Support for Apache2
    apt:
      name: libapache2-mod-php
   when: ansible_distribution == "Ubuntu"

# added 'when' statement to specify Ubuntu as Linux distro
```

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

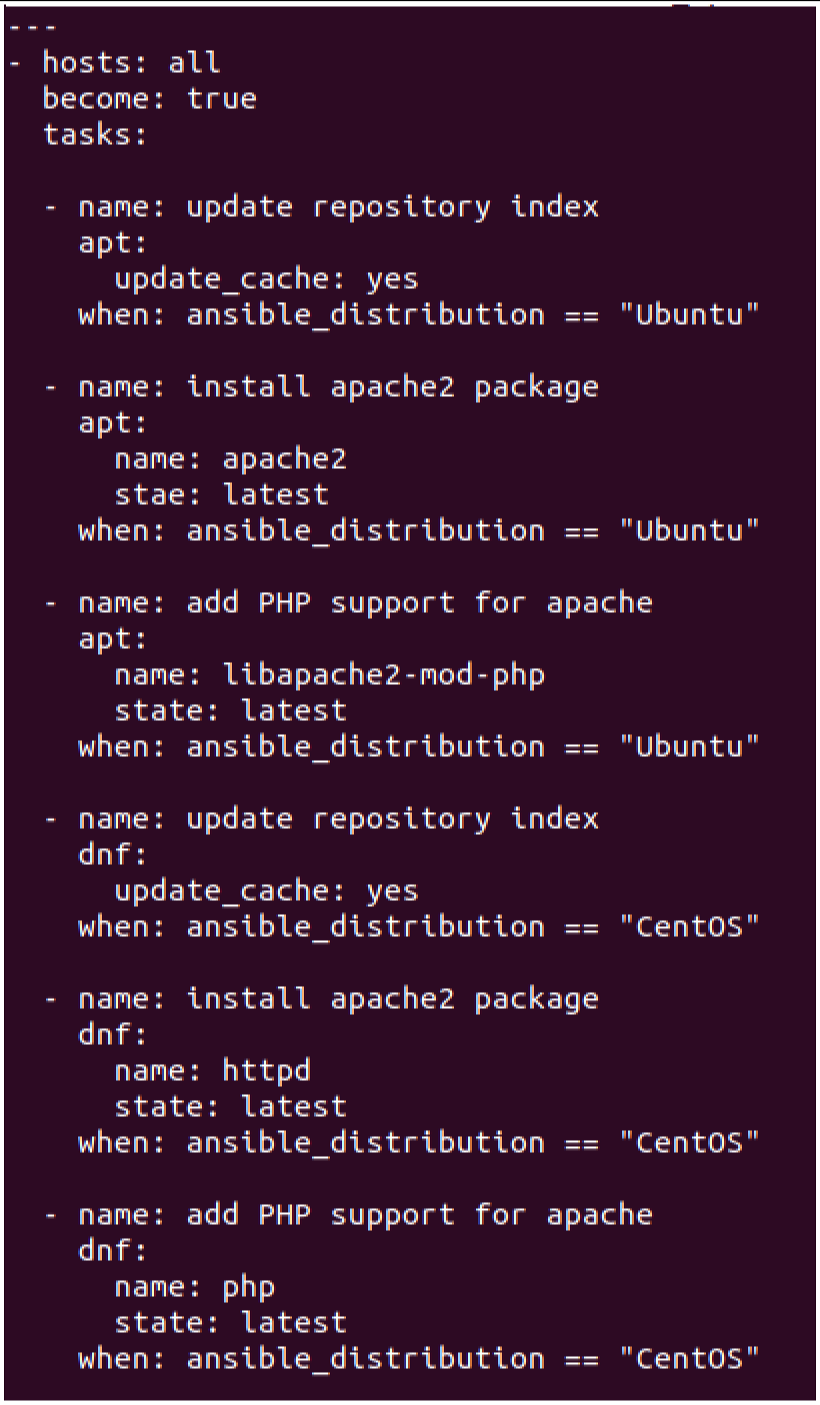
By implementing the 'when' statement, it makes sure to only implement the aforementioned commands when the Linux distribution matches the specified distro.

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]
```

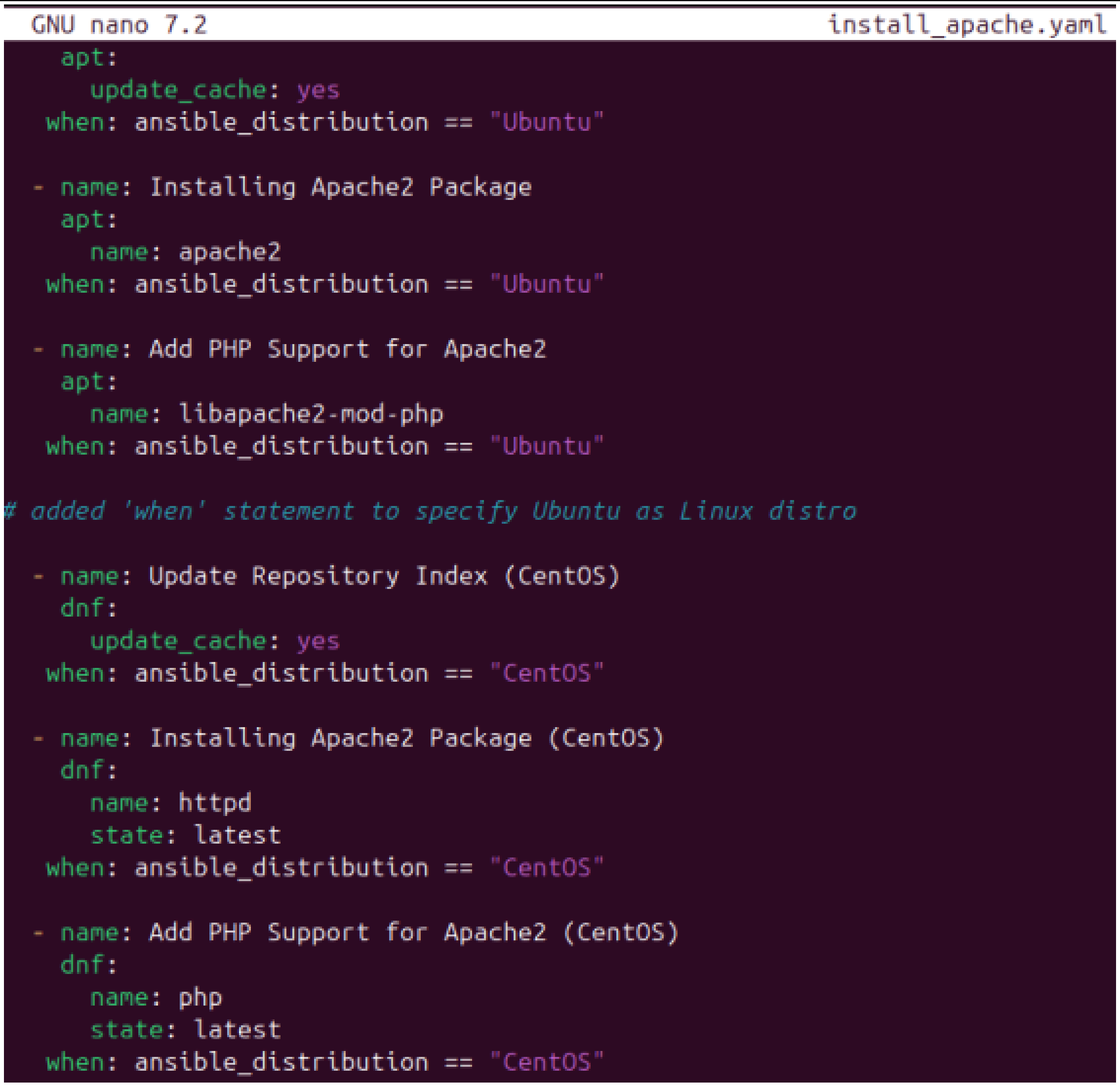
Note: This will work also if you try. Notice the changes are highlighted.

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



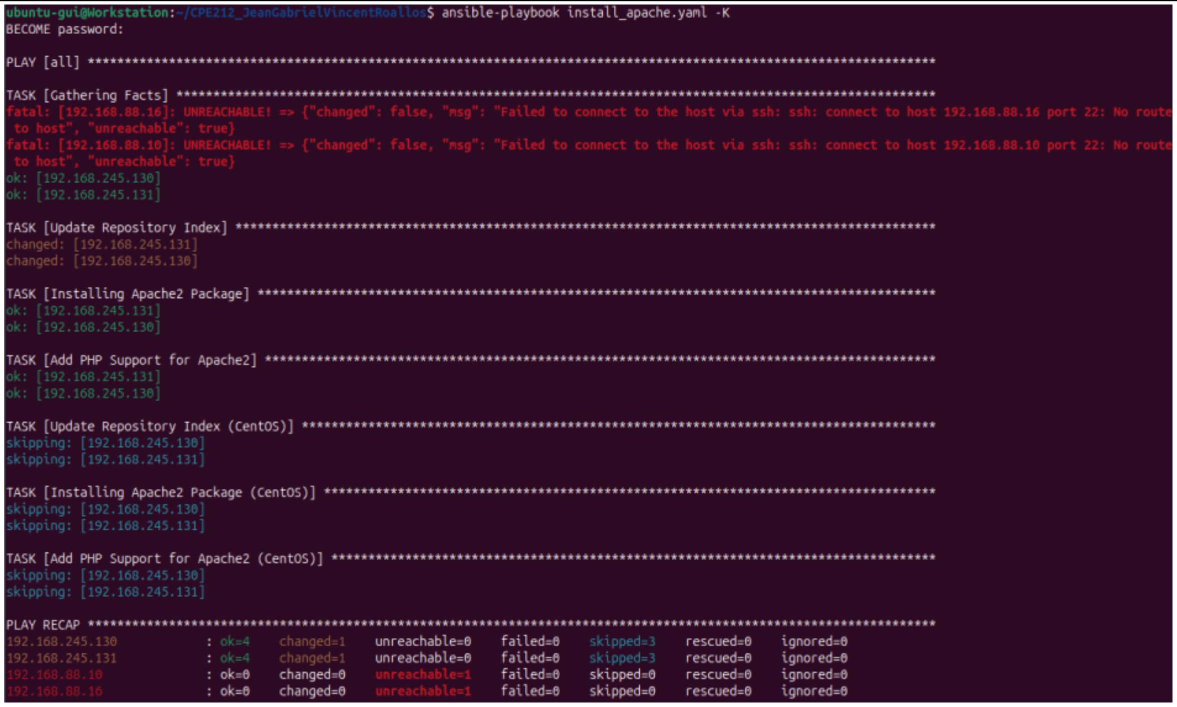
```
---
- hosts: all
  become: true
  tasks:

  - 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.yaml
    apt:
      update_cache: yes
   when: ansible_distribution == "Ubuntu"

  - name: Installing Apache2 Package
    apt:
      name: apache2
   when: ansible_distribution == "Ubuntu"

  - name: Add PHP Support for Apache2
    apt:
      name: libapache2-mod-php
   when: ansible_distribution == "Ubuntu"

# added 'when' statement to specify Ubuntu as Linux distro

  - name: Update Repository Index (CentOS)
    dnf:
      update_cache: yes
   when: ansible_distribution == "CentOS"

  - name: Installing Apache2 Package (CentOS)
    dnf:
      name: httpd
      state: latest
   when: ansible_distribution == "CentOS"

  - name: Add PHP Support for Apache2 (CentOS)
    dnf:
      name: php
      state: latest
   when: ansible_distribution == "CentOS"
```

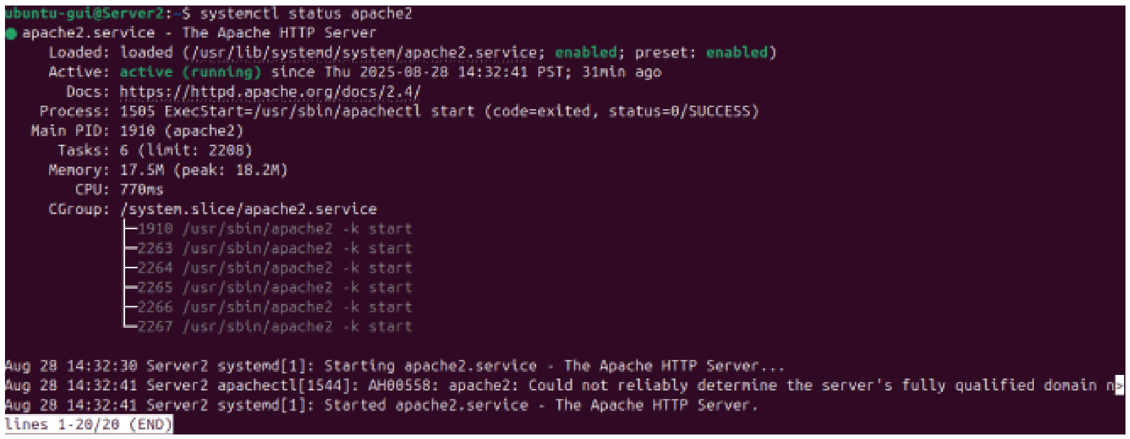
Run *ansible-playbook --ask-become-pass install_apache.yml* and describe the result.



```
ubuntu-gui@Workstation:~/CPE212_JeanGabrielVincentRoallos$ ansible-playbook install_apache.yaml -K
BECOME password:

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

TASK [Gathering Facts] *****************************************************************************************
fatal: [192.168.88.16]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to the host via ssh: ssh: connect to host 192.168.88.16 port 22: No route
 to host", "unreachable": true}
fatal: [192.168.88.10]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to the host via ssh: ssh: connect to host 192.168.88.10 port 22: No route
 to host", "unreachable": true}
ok: [192.168.245.130]
ok: [192.168.245.131]

TASK [Update Repository Index] *********************************************************************************
changed: [192.168.245.131]
changed: [192.168.245.130]

TASK [Installing Apache2 Package] ******************************************************************************
ok: [192.168.245.131]
ok: [192.168.245.130]

TASK [Add PHP Support for Apache2] *****************************************************************************
ok: [192.168.245.131]
ok: [192.168.245.130]

TASK [Update Repository Index (CentOS)] ************************************************************************
skipping: [192.168.245.130]
skipping: [192.168.245.131]

TASK [Installing Apache2 Package (CentOS)] *********************************************************************
skipping: [192.168.245.130]
skipping: [192.168.245.131]

TASK [Add PHP Support for Apache2 (CentOS)] ********************************************************************
skipping: [192.168.245.130]
skipping: [192.168.245.131]

PLAY RECAP *****************************************************************************************************
192.168.245.130            : ok=4    changed=1    unreachable=0    failed=0    skipped=3    rescued=0    ignored=0
192.168.245.131            : ok=4    changed=1    unreachable=0    failed=0    skipped=3    rescued=0    ignored=0
192.168.88.10              : ok=0    changed=0    unreachable=1    failed=0    skipped=0    rescued=0    ignored=0
192.168.88.16              : ok=0    changed=0    unreachable=1    failed=0    skipped=0    rescued=0    ignored=0
```

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.



```
ubuntu-gui@Server2:~$ systemctl status apache2
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; preset: enabled)
     Active: active (running) since Thu 2025-08-28 14:32:41 PST; 31min ago
       Docs: https://httpd.apache.org/docs/2.4/
    Process: 1505 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
   Main PID: 1910 (apache2)
      Tasks: 6 (limit: 2208)
     Memory: 17.5M (peak: 18.2M)
        CPU: 770ms
     CGroup: /system.slice/apache2.service
             ├─1910 /usr/sbin/apache2 -k start
             ├─2263 /usr/sbin/apache2 -k start
             ├─2264 /usr/sbin/apache2 -k start
             ├─2265 /usr/sbin/apache2 -k start
             ├─2266 /usr/sbin/apache2 -k start
             └─2267 /usr/sbin/apache2 -k start

Aug 28 14:32:30 Server2 systemd[1]: Starting apache2.service - The Apache HTTP Server...
Aug 28 14:32:41 Server2 apachectl[1544]: AH00558: apache2: Could not reliably determine the server's fully qualified domain n>
Aug 28 14:32:41 Server2 systemd[1]: Started apache2.service - The Apache HTTP Server.
lines 1-20/20 (END)
```

I do not have CentOS set up, so I used 'systemctl status apache2', which is the Ubuntu equivalent of the given command

5.2 Issue the following command to start the service:

sudo systemctl start httpd

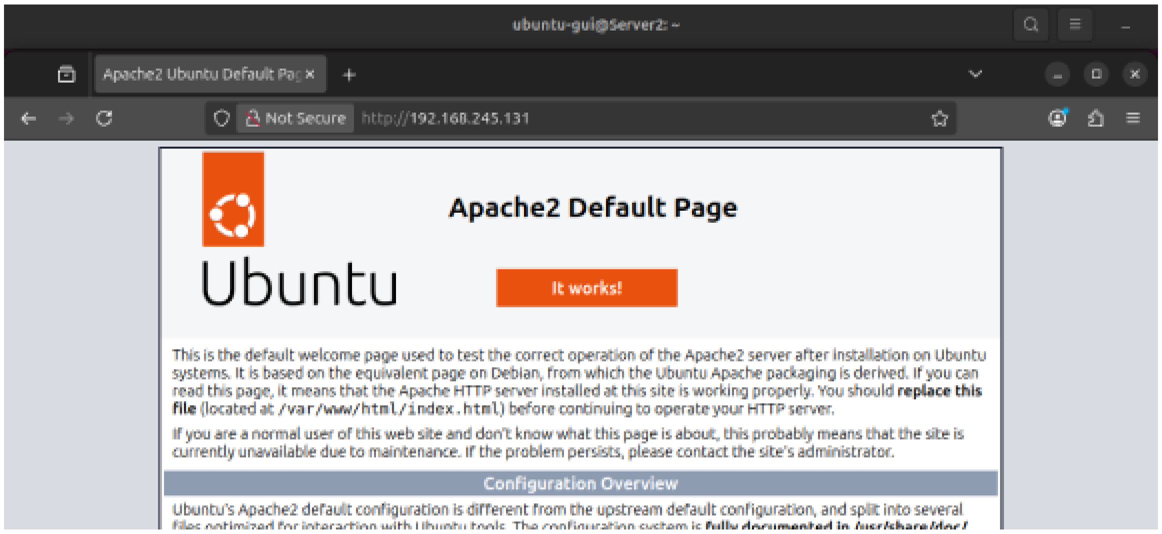
(When prompted, enter the sudo password)

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

(The result should be a success)

To start a process in Ubuntu, enter 'systemctl start [process_name]'

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)



Apache2 Default Page

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

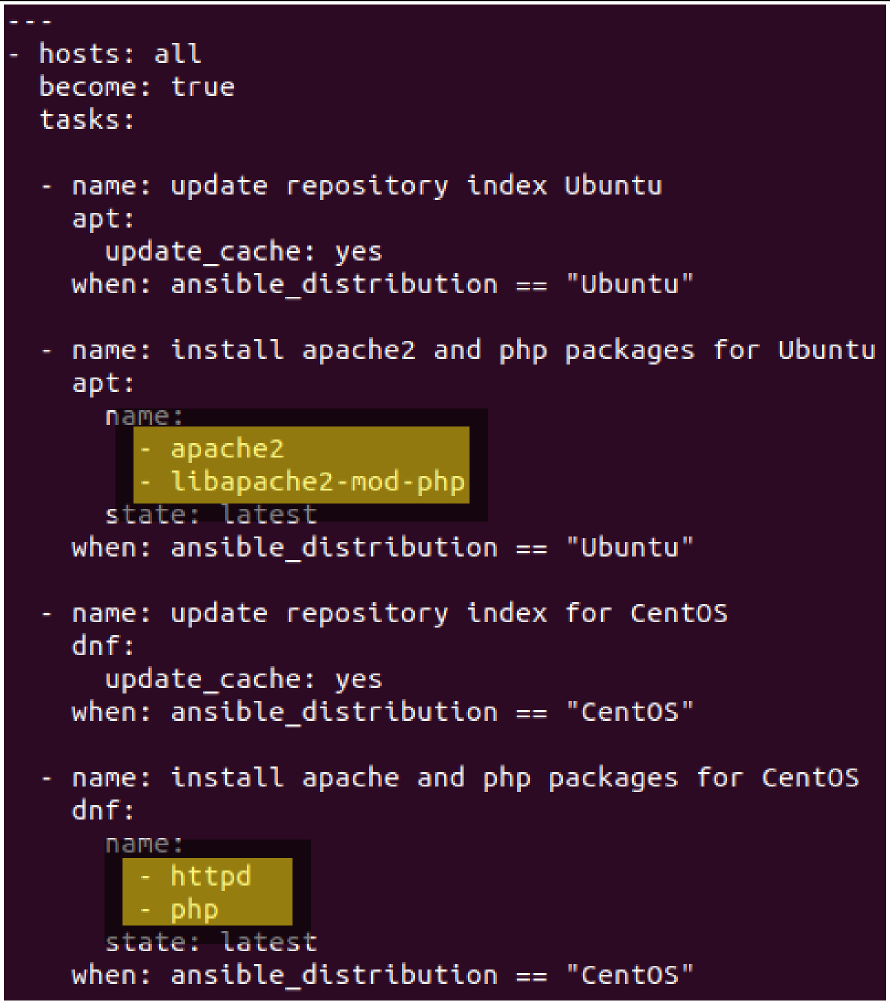
Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several

Task 2: Refactoring playbook

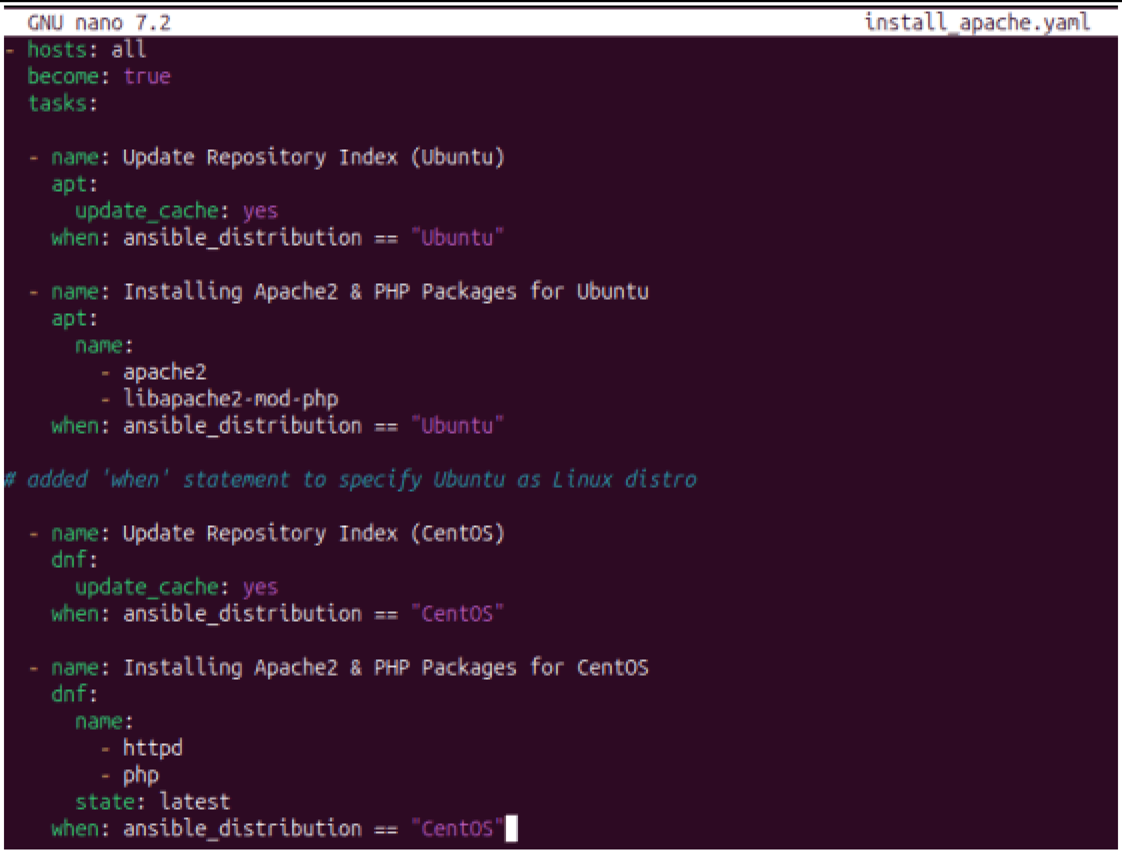
This time, we want to make sure that our playbook is efficient and that the codes are easier to read. This will also makes run ansible more quickly if it has to execute fewer tasks to do the same thing.

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:



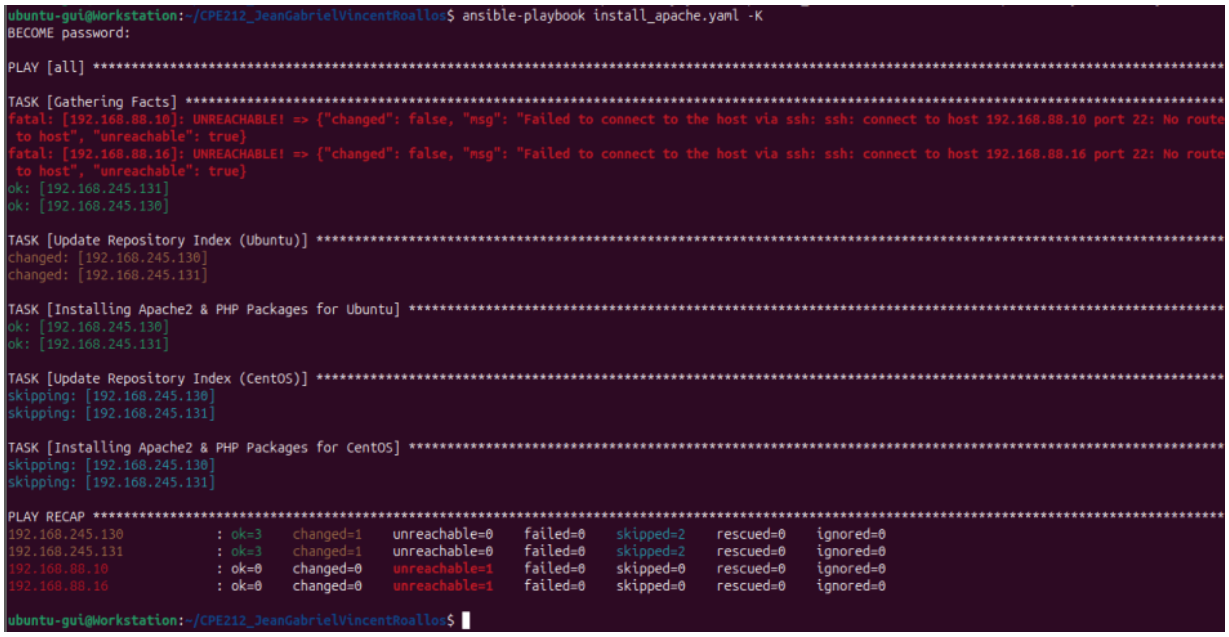
```
---
- hosts: all
  become: true
  tasks:

  - 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.



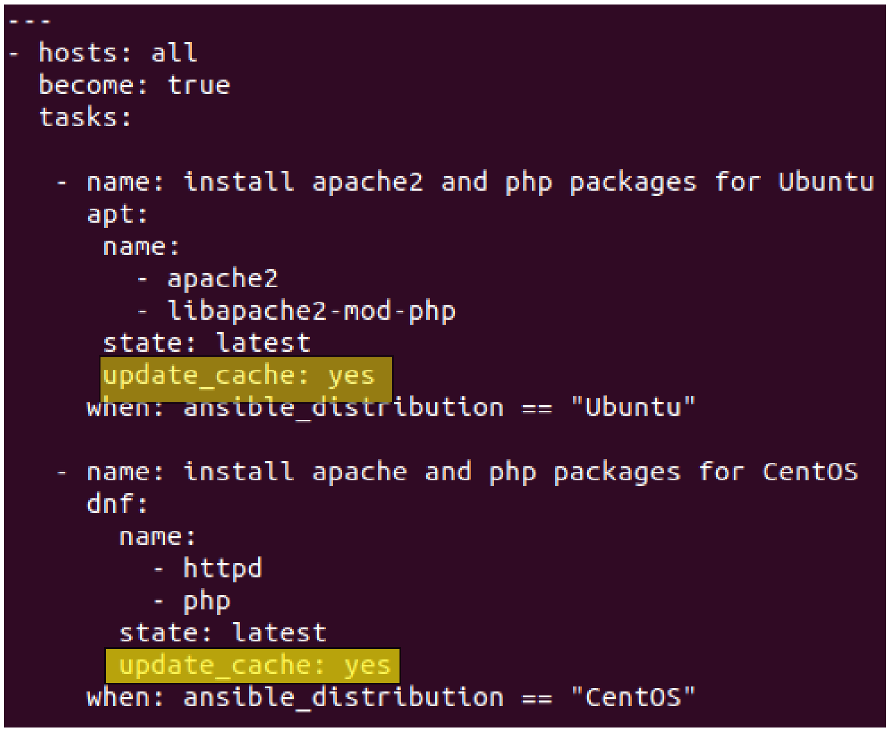
```
GNU nano 7.2                                                    install_apache.yaml
- hosts: all
  become: true
  tasks:

  - name: Update Repository Index (Ubuntu)
    apt:
      update_cache: yes
    when: ansible_distribution == "Ubuntu"

  - name: Installing Apache2 & PHP Packages for Ubuntu
    apt:
      name:
        - apache2
        - libapache2-mod-php
    when: ansible_distribution == "Ubuntu"

# added 'when' statement to specify Ubuntu as Linux distro

  - name: Update Repository Index (CentOS)
    dnf:
      update_cache: yes
    when: ansible_distribution == "CentOS"

  - name: Installing Apache2 & PHP Packages for CentOS
    dnf:
      name:
        - httpd
        - php
      state: latest
    when: ansible_distribution == "CentOS"
```

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



```
ubuntu-gui@Workstation:~/CPE212_JeanGabrielVincentRoallos$ ansible-playbook install_apache.yaml -K
BECOME password:

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

TASK [Gathering Facts] ***********************************************************************************
fatal: [192.168.88.10]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to the host via ssh: ssh: connect to host 192.168.88.10 port 22: No route
 to host", "unreachable": true}
fatal: [192.168.88.16]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to the host via ssh: ssh: connect to host 192.168.88.16 port 22: No route
 to host", "unreachable": true}
ok: [192.168.245.131]
ok: [192.168.245.130]

TASK [Update Repository Index (Ubuntu)] ******************************************************************
changed: [192.168.245.130]
changed: [192.168.245.131]

TASK [Installing Apache2 & PHP Packages for Ubuntu] ******************************************************
ok: [192.168.245.130]
ok: [192.168.245.131]

TASK [Update Repository Index (CentOS)] ******************************************************************
skipping: [192.168.245.130]
skipping: [192.168.245.131]

TASK [Installing Apache2 & PHP Packages for CentOS] ******************************************************
skipping: [192.168.245.130]
skipping: [192.168.245.131]

PLAY RECAP ***********************************************************************************************
192.168.245.130            : ok=3    changed=1    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
192.168.245.131            : ok=3    changed=1    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
192.168.88.10              : ok=0    changed=0    unreachable=1    failed=0    skipped=0    rescued=0    ignored=0
192.168.88.16              : ok=0    changed=0    unreachable=1    failed=0    skipped=0    rescued=0    ignored=0

ubuntu-gui@Workstation:~/CPE212_JeanGabrielVincentRoallos$
```

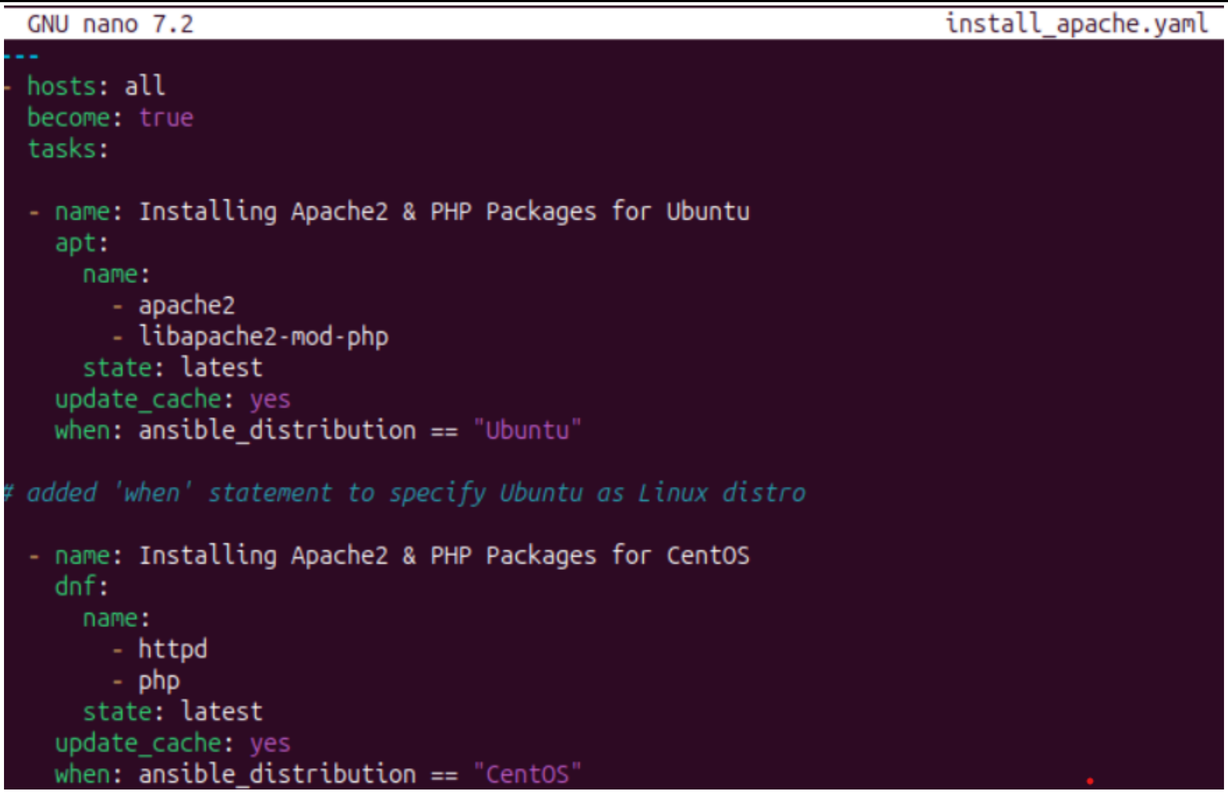
2. Edit the playbook *install_apache.yml* again. In task 2.1, we consolidated the plays into one play. This time we can actually consolidated everything in just 2 plays. This can be done by removing the update repository play and putting the command *update_cache: yes* below the command *state: latest*.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

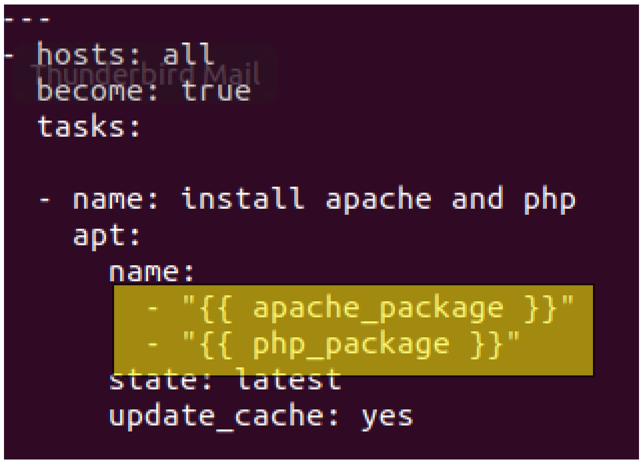
Make sure to save the file and exit.

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



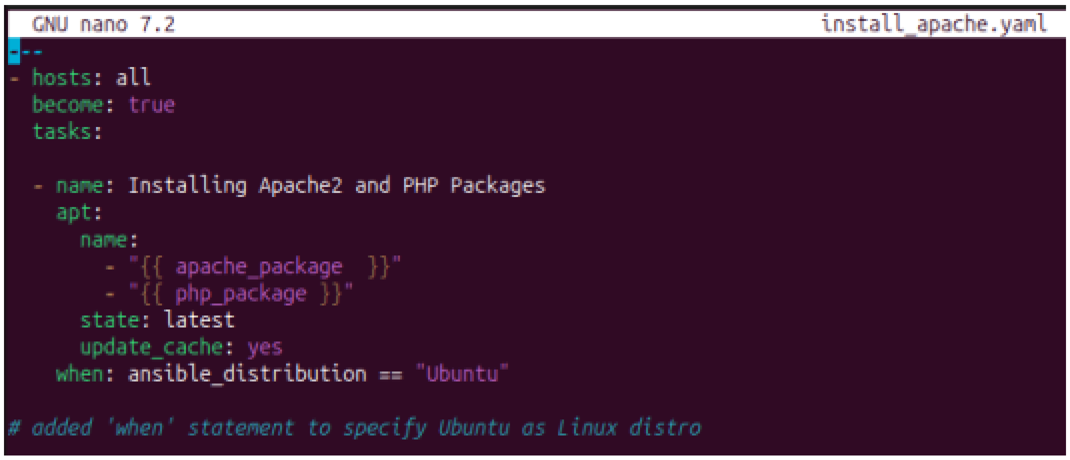
```
GNU nano 7.2                                                    install_apache.yaml
---
- hosts: all
  become: true
  tasks:

  - name: Installing Apache2 & PHP Packages for Ubuntu
    apt:
      name:
        - apache2
        - libapache2-mod-php
      state: latest
    update_cache: yes
    when: ansible_distribution == "Ubuntu"

# added 'when' statement to specify Ubuntu as Linux distro

  - name: Installing Apache2 & PHP Packages for CentOS
    dnf:
      name:
        - httpd
        - php
      state: latest
    update_cache: yes
    when: ansible_distribution == "CentOS"
```

3. Finally, we can consolidate these 2 plays in just 1 play. This can be done by declaring variables that will represent the packages that we want to install. Basically, the apache_package and php_package are variables. The names are arbitrary, which means we can choose different names. We also take out the line when: ansible_distribution. Edit the playbook *install_apache.yml* again and make sure to follow the below image. Make sure to save the file and exit.

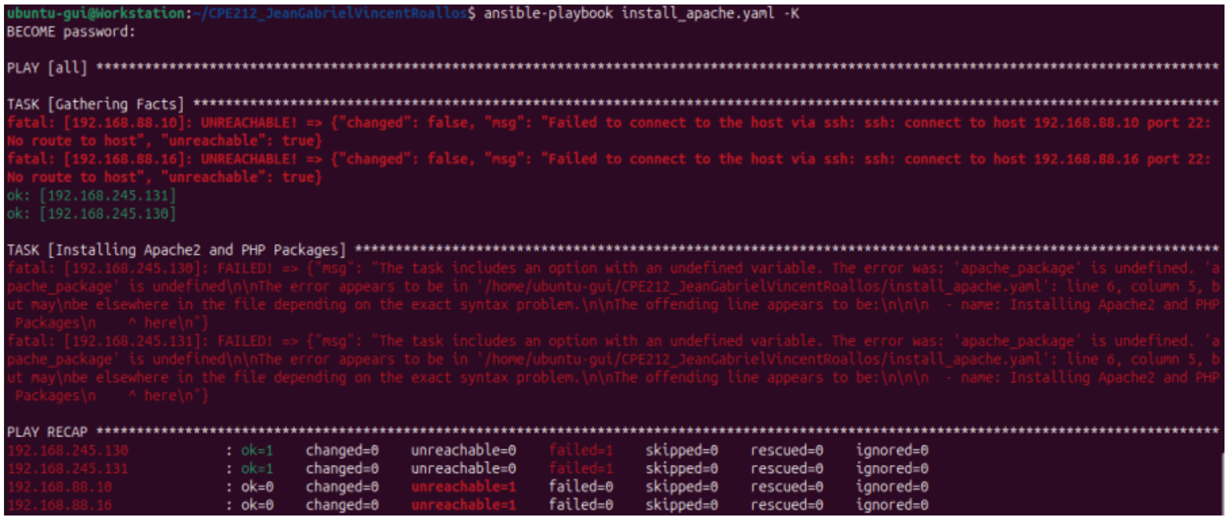


```
---
- hosts: all
  become: true
  tasks:

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

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



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

  - name: Installing Apache2 and PHP Packages
    apt:
      name:
        - "{{ apache_package  }}"
        - "{{ php_package }}"
      state: latest
      update_cache: yes
    when: ansible_distribution == "Ubuntu"

# added 'when' statement to specify Ubuntu as Linux distro
```

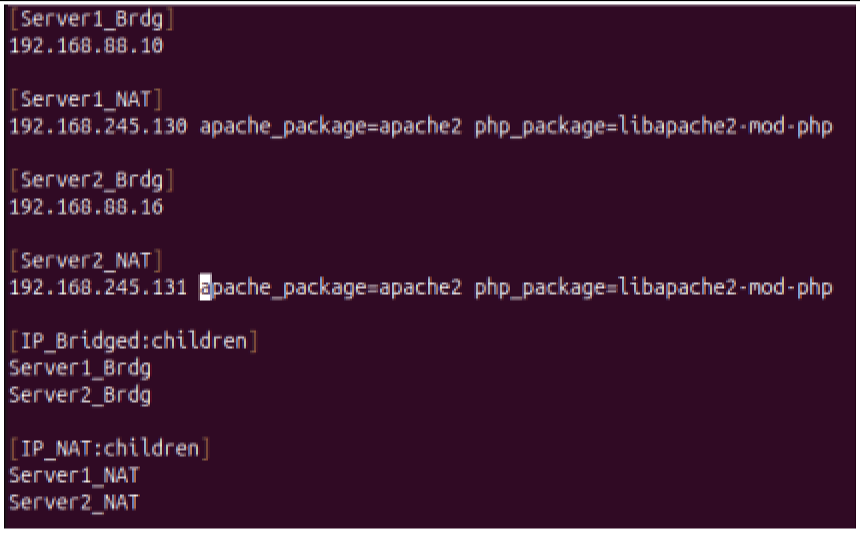


```
ubuntu-gui@Workstation:~/CPE212_JeanGabrielVincentRoallos$ ansible-playbook install_apache.yaml -K
BECOME password:

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

TASK [Gathering Facts] *****************************************************************************************************
fatal: [192.168.88.10]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to the host via ssh: ssh: connect to host 192.168.88.10 port 22:
No route to host", "unreachable": true}
fatal: [192.168.88.16]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to the host via ssh: ssh: connect to host 192.168.88.16 port 22:
No route to host", "unreachable": true}
ok: [192.168.245.131]
ok: [192.168.245.130]

TASK [Installing Apache2 and PHP Packages] *********************************************************************************
fatal: [192.168.245.130]: FAILED! => {"msg": "The task includes an option with an undefined variable. The error was: 'apache_package' is undefined. 'a
pache_package' is undefined\n\nThe error appears to be in '/home/ubuntu-gui/CPE212_JeanGabrielVincentRoallos/install_apache.yaml': line 6, column 5, b
ut may\nbe elsewhere in the file depending on the exact syntax problem.\n\nThe offending line appears to be:\n\n\n  - name: Installing Apache2 and PHP
 Packages\n    ^ here\n"}
fatal: [192.168.245.131]: FAILED! => {"msg": "The task includes an option with an undefined variable. The error was: 'apache_package' is undefined. 'a
pache_package' is undefined\n\nThe error appears to be in '/home/ubuntu-gui/CPE212_JeanGabrielVincentRoallos/install_apache.yaml': line 6, column 5, b
ut may\nbe elsewhere in the file depending on the exact syntax problem.\n\nThe offending line appears to be:\n\n\n  - name: Installing Apache2 and PHP
 Packages\n    ^ here\n"}

PLAY RECAP *****************************************************************************************************************
192.168.245.130            : ok=1    changed=0    unreachable=0    failed=1    skipped=0    rescued=0    ignored=0
192.168.245.131            : ok=1    changed=0    unreachable=0    failed=1    skipped=0    rescued=0    ignored=0
192.168.88.10              : ok=0    changed=0    unreachable=1    failed=0    skipped=0    rescued=0    ignored=0
192.168.88.16              : ok=0    changed=0    unreachable=1    failed=0    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:

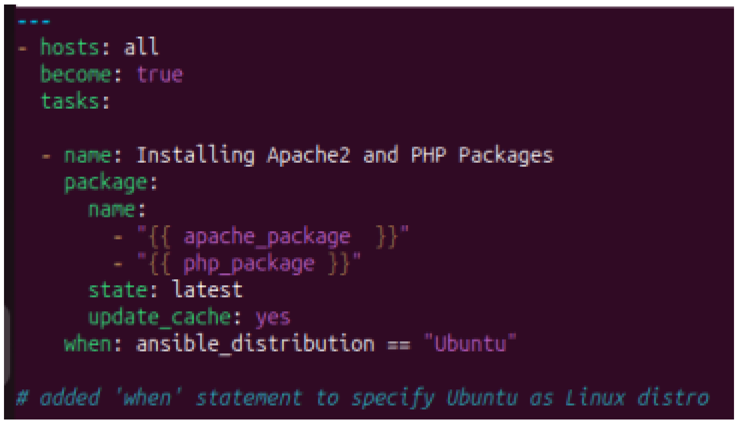
```
192.168.56.120 apache_package=apache2 php_package=libapache2-mod-php
192.168.56.121 apache_package=apache2 php_package=libapache2-mod-php
192.168.56.122 apache_package=httpd php_package=php
```

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



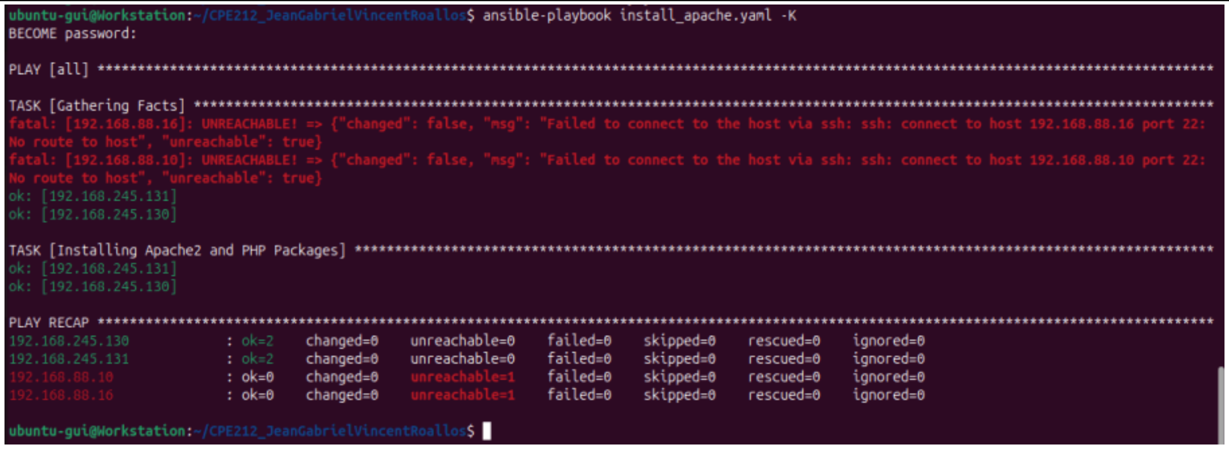
```
[Server1_Brdg]
192.168.88.10

[Server1_NAT]
192.168.245.130 apache_package=apache2 php_package=libapache2-mod-php

[Server2_Brdg]
192.168.88.16

[Server2_NAT]
192.168.245.131 apache_package=apache2 php_package=libapache2-mod-php

[IP_Bridged:children]
Server1_Brdg
Server2_Brdg

[IP_NAT:children]
Server1_NAT
Server2_NAT
```

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



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

  - name: Installing Apache2 and PHP Packages
    package:
      name:
        - "{{ apache_package  }}"
        - "{{ php_package }}"
      state: latest
      update_cache: yes
    when: ansible_distribution == "Ubuntu"

# added 'when' statement to specify Ubuntu as Linux distro
```

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



```
ubuntu-gui@Workstation:~/CPE212_JeanGabrielVincentRoallos$ ansible-playbook install_apache.yaml -K
BECOME password:

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

TASK [Gathering Facts] *******************************************************************************************
fatal: [192.168.88.16]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to the host via ssh: ssh: connect to host 192.168.88.16 port 22:
No route to host", "unreachable": true}
fatal: [192.168.88.10]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to the host via ssh: ssh: connect to host 192.168.88.10 port 22:
No route to host", "unreachable": true}
ok: [192.168.245.131]
ok: [192.168.245.130]

TASK [Installing Apache2 and PHP Packages] ***********************************************************************
ok: [192.168.245.131]
ok: [192.168.245.130]

PLAY RECAP *******************************************************************************************************
192.168.245.130            : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
192.168.245.131            : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
192.168.88.10              : ok=0    changed=0    unreachable=1    failed=0    skipped=0    rescued=0    ignored=0
192.168.88.16              : ok=0    changed=0    unreachable=1    failed=0    skipped=0    rescued=0    ignored=0

ubuntu-gui@Workstation:~/CPE212_JeanGabrielVincentRoallos$
```

Supplementary Activity:

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

Shell

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

    - name: Installing Apache2 and PHP Packages
      package:
        name:
         - httpd
         - php
        state: latest
        update_cache: yes
      when: ansible_distribution == "RedHat"
```

Reflections:

Answer the following:

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

Refactoring playbooks “generalizes” the syntax used in an Ansible playbook, making the playbook able to cater to multiple types of systems, compressing the same tasks into one play instead of dedicating separate plays per system distribution. This improves the efficiency of the playbook itself, having to run less plays from a playbook.

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

The ‘when’ in Ansible is a conditional statement that can be used to allow or block tasks to be run when the criteria results are true or false. In the activity, we have used the ‘when’ statement to verify if the Ansible distribution of the managed node is appropriate to the play segment it specified above. This prevents simple errors from wrong commands that might be unsupported to other distributions.