Name: Nealian Beth B. Nanquil	**Date Performed:** September 15, 2022
Course/Section: CPE232-CPE31S23	**Date Submitted:** September 15, 2022
Instructor: Engr. Jonathan Taylar	**Semester and SY:** 1st sem 2022- 2023

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**:

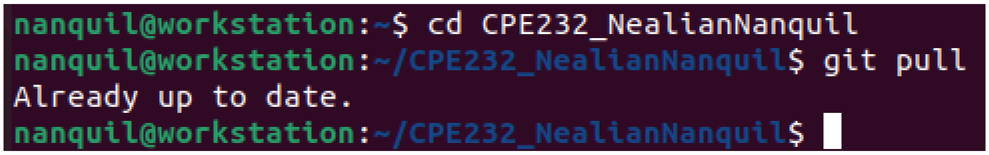
We are going to look at a way that we can differentiate a playbook by a host in terms of which distribution the host is running. It's very common in most Linux shops to run multiple distributions, for example, Ubuntu shop or Debian shop and you need a different distribution for a one off-case or perhaps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

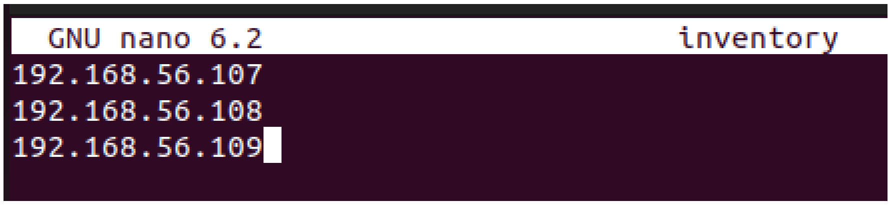
1. In the local machine, make sure you are in the local repository directory (*CPE232_yourname*). Issue the command git pull. When prompted, enter the correct passphrase or password. Describe what happened when you issue this command. Did something happen? Why? Yes, something happened. Because the git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



```
nanquil@workstation:~$ cd CPE232_NealianNanquil
nanquil@workstation:~/CPE232_NealianNanquil$ git pull
Already up to date.
nanquil@workstation:~/CPE232_NealianNanquil$ 
```

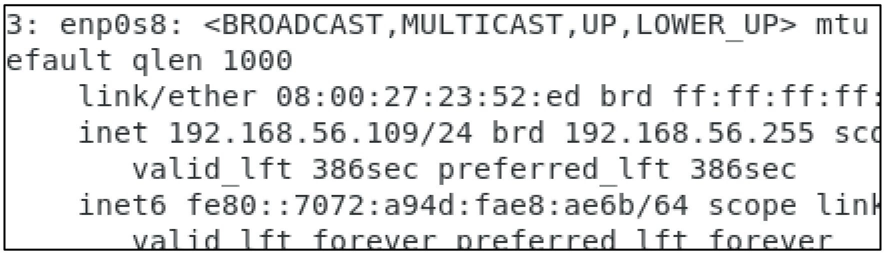
Figure 1.1: Entered the command git pull.

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."



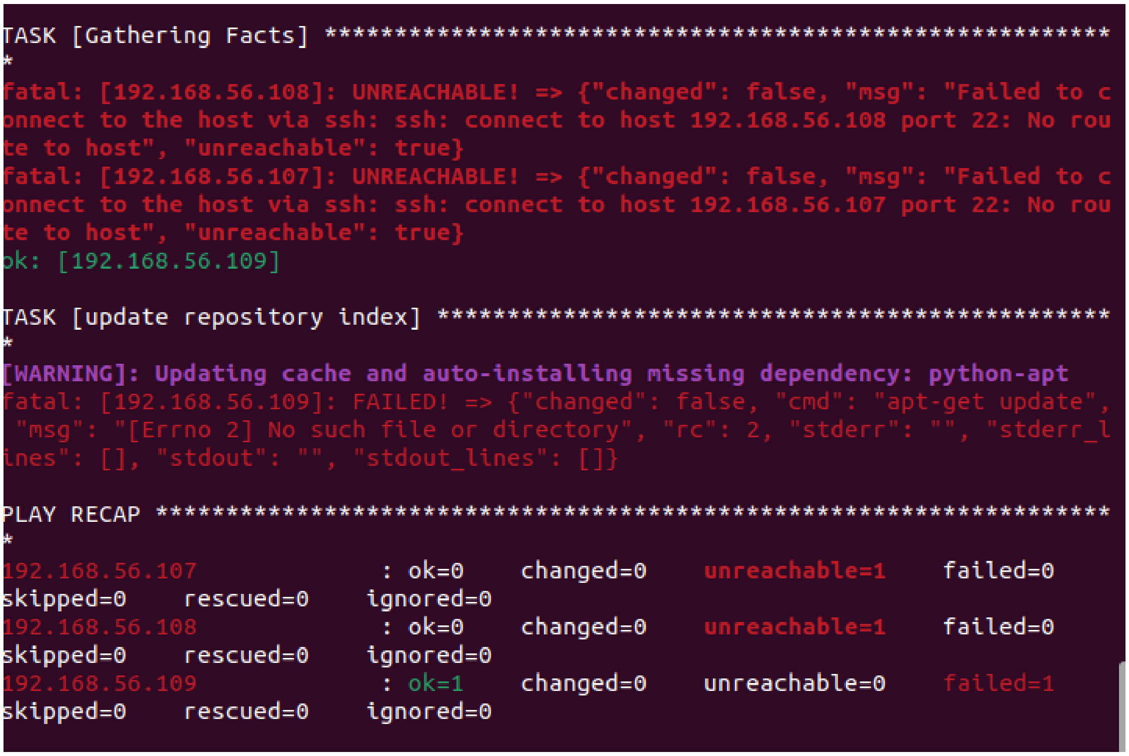
```
  GNU nano 6.2                    inventory
192.168.56.107
192.168.56.108
192.168.56.109
```

Figure 1.2: Edited inventory file.



```
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu
efault qlen 1000
    link/ether 08:00:27:23:52:ed brd ff:ff:ff:ff:
    inet 192.168.56.109/24 brd 192.168.56.255 sco
       valid_lft 386sec preferred_lft 386sec
    inet6 fe80::7072:a94d:fae8:ae6b/64 scope link
       valid_lft forever preferred_lft forever
```

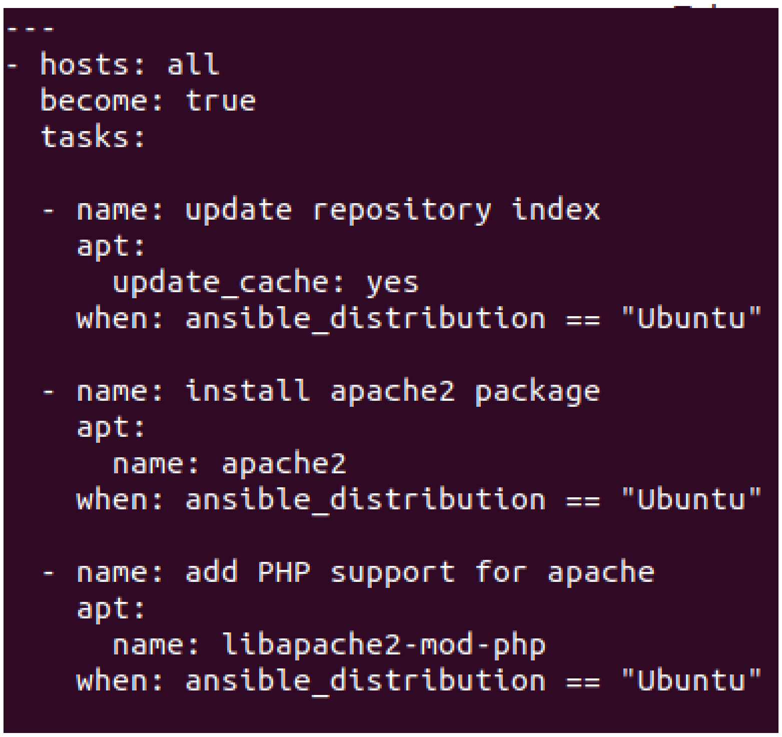
Figure 1.3: IP Address of the CentOS.



```
TASK [Gathering Facts] *****************************************************
*
fatal: [192.168.56.108]: UNREACHABLE! => {"changed": false, "msg": "Failed to c
onnect to the host via ssh: ssh: connect to host 192.168.56.108 port 22: No rou
te to host", "unreachable": true}
fatal: [192.168.56.107]: UNREACHABLE! => {"changed": false, "msg": "Failed to c
onnect to the host via ssh: ssh: connect to host 192.168.56.107 port 22: No rou
te to host", "unreachable": true}
ok: [192.168.56.109]

TASK [update repository index] *********************************************
*
[WARNING]: Updating cache and auto-installing missing dependency: python-apt
fatal: [192.168.56.109]: FAILED! => {"changed": false, "cmd": "apt-get update",
 "msg": "[Errno 2] No such file or directory", "rc": 2, "stderr": "", "stderr_l
ines": [], "stdout": "", "stdout_lines": []}

PLAY RECAP *****************************************************************
*
192.168.56.107             : ok=0    changed=0    unreachable=1    failed=0
skipped=0    rescued=0    ignored=0
192.168.56.108             : ok=0    changed=0    unreachable=1    failed=0
skipped=0    rescued=0    ignored=0
192.168.56.109             : ok=1    changed=0    unreachable=0    failed=1
skipped=0    rescued=0    ignored=0
```

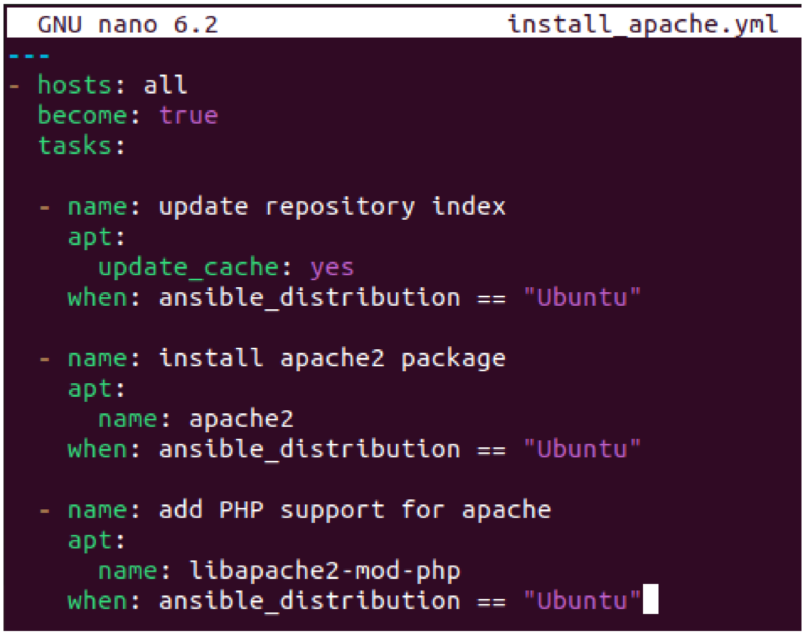
Figure 1.4: Output of the command used to execute the playbook.

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"
```

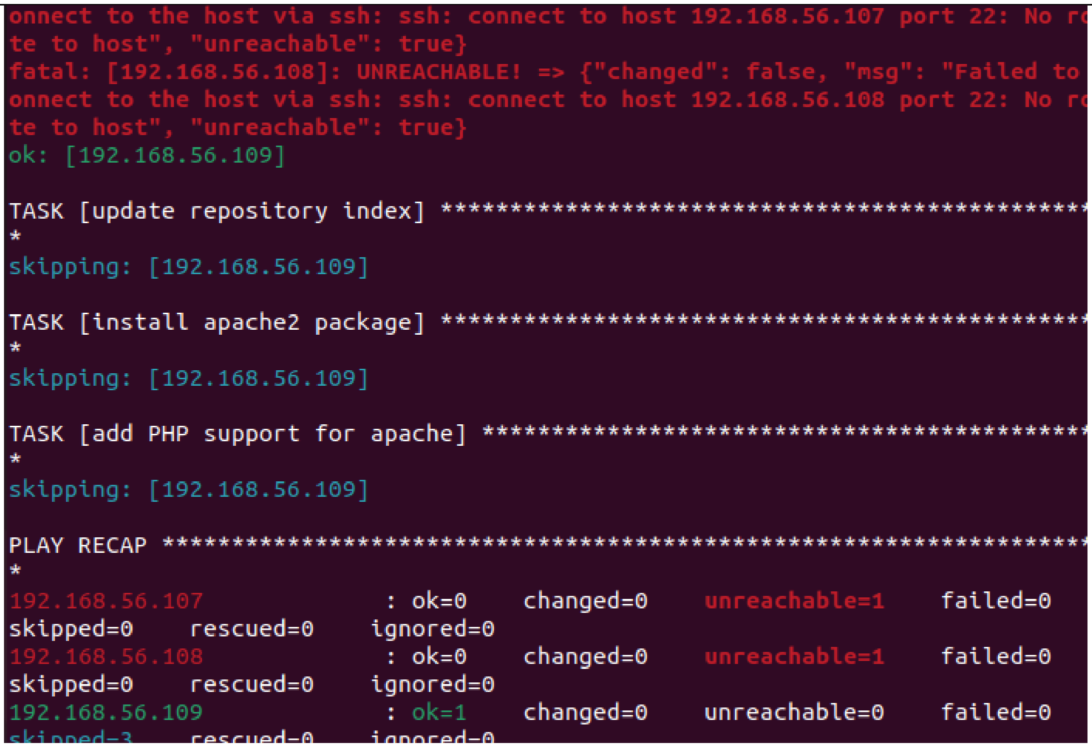
Make sure to save the file and exit.'



```
  GNU nano 6.2                      install_apache.yml
---
- hosts: all
  become: true
  tasks:

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

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

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

Figure 1.5: Edited the install_apache.yml playbook.

Run *ansible-playbook --ask-become-pass install_apache.yml* and describe the result. The output after entering this command, is the CentOS has now failed=0.



```
onnect to the host via ssh: ssh: connect to host 192.168.56.107 port 22: No ro
te to host", "unreachable": true}
fatal: [192.168.56.108]: UNREACHABLE! => {"changed": false, "msg": "Failed to
onnect to the host via ssh: ssh: connect to host 192.168.56.108 port 22: No ro
te to host", "unreachable": true}
ok: [192.168.56.109]

TASK [update repository index] **********************************************
*
skipping: [192.168.56.109]

TASK [install apache2 package] **********************************************
*
skipping: [192.168.56.109]

TASK [add PHP support for apache] *******************************************
*
skipping: [192.168.56.109]

PLAY RECAP ******************************************************************
*
192.168.56.107             : ok=0    changed=0    unreachable=1    failed=0
skipped=0    rescued=0    ignored=0
192.168.56.108             : ok=0    changed=0    unreachable=1    failed=0
skipped=0    rescued=0    ignored=0
192.168.56.109             : ok=1    changed=0    unreachable=0    failed=0
skipped=3    rescued=0    ignored=0
```

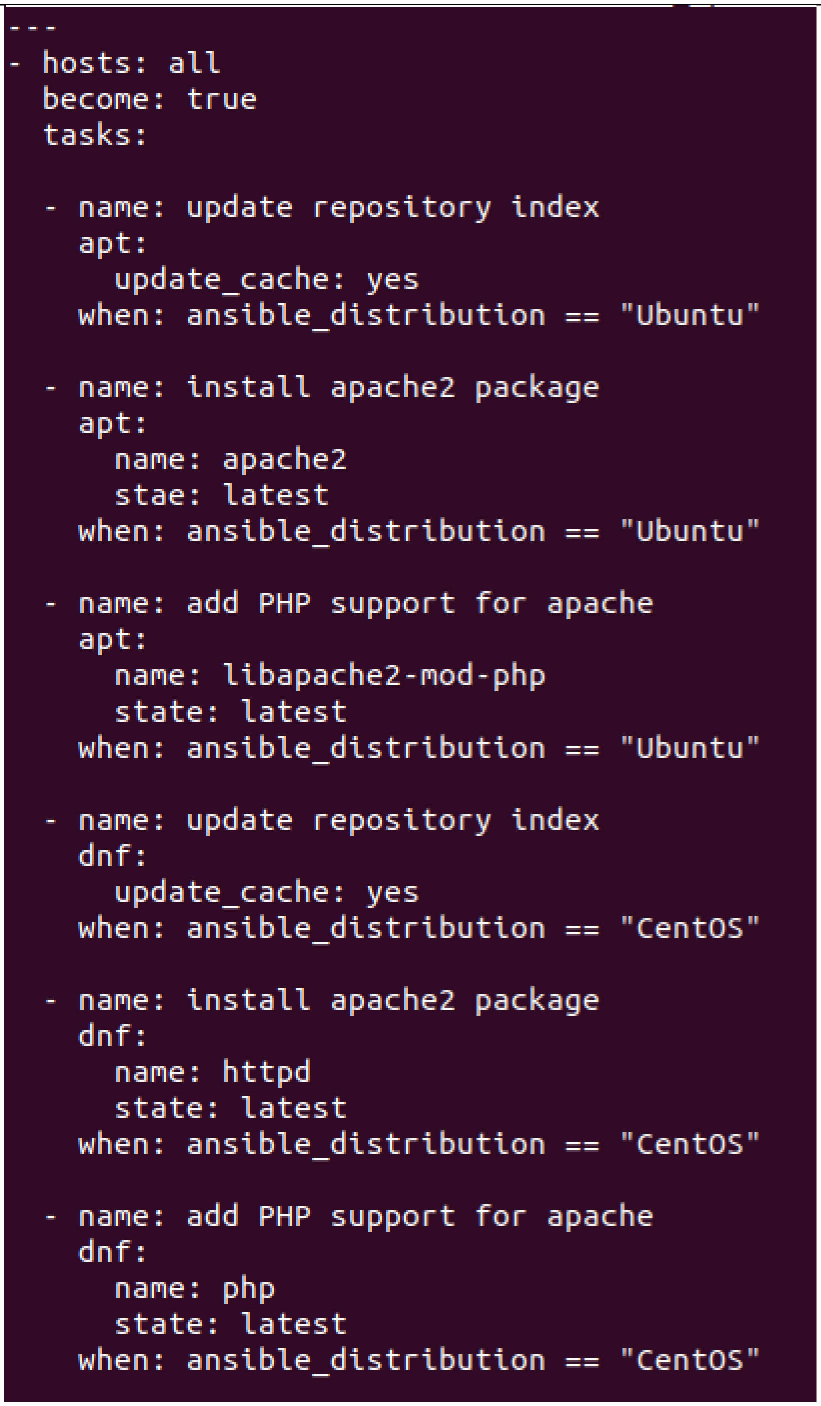
Figure 1.6: Output of the command used to execute the playbook.

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]
```

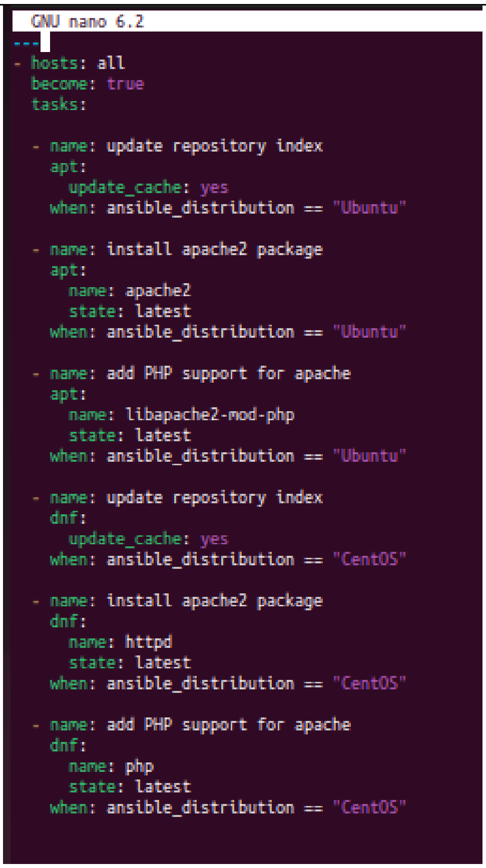
Note: This will work also if you try. Notice the changes are highlighted.

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"
```

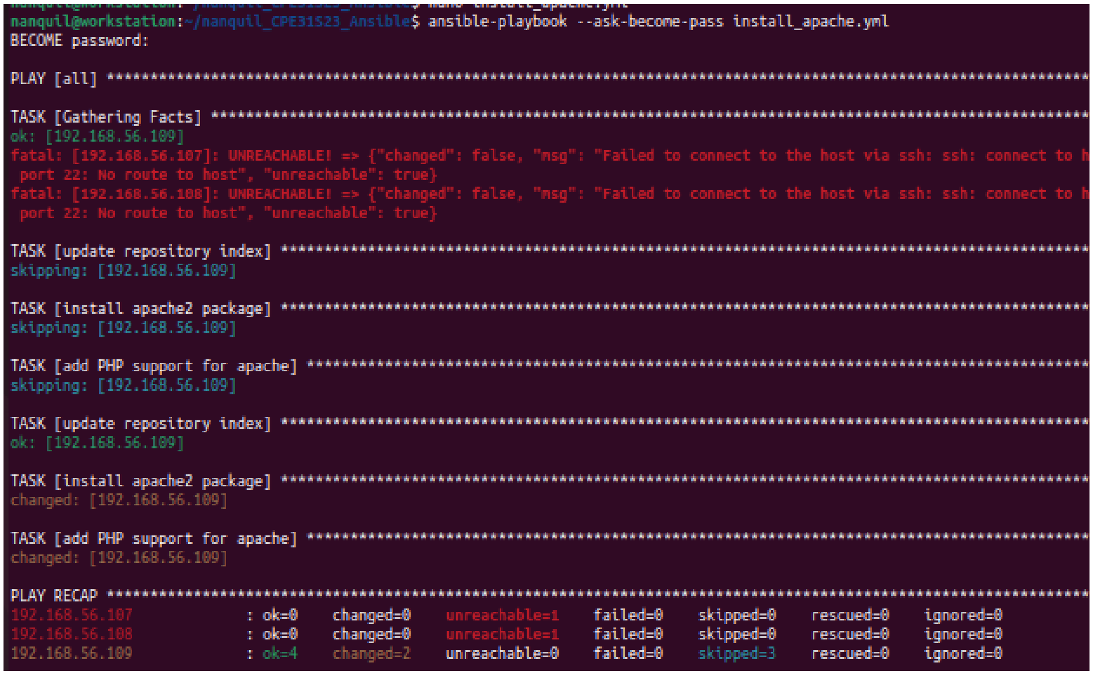
Make sure to save and exit.



```
GNU nano 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
    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"
```

Figure 1.7: Edited the install_apache.yml playbook.

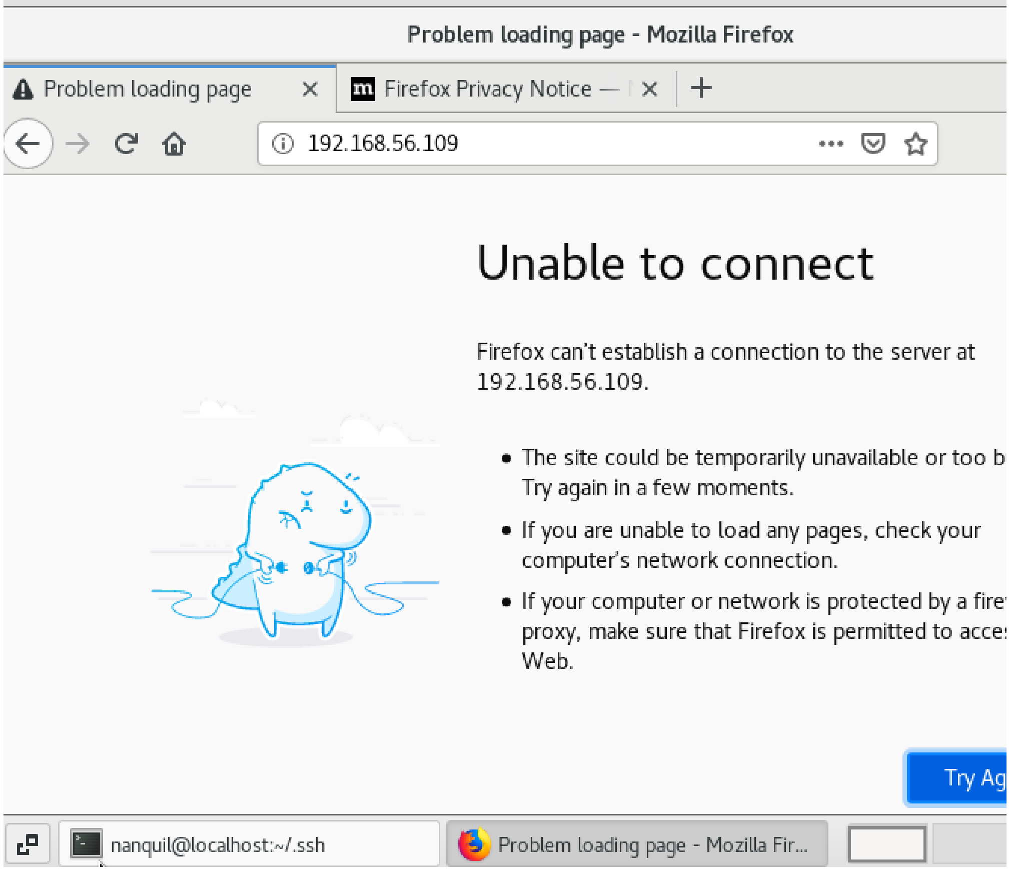
Run *ansible-playbook –ask-become-pass install_apache.yml* and describe the result.



```
nanquil@workstation:~/nanquil_CPE31S23_Ansible$ ansible-playbook --ask-become-pass install_apache.yml
BECOME password:

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

TASK [Gathering Facts] *************************************************************************************
ok: [192.168.56.109]
fatal: [192.168.56.107]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to the host via ssh: ssh: connect to h
 port 22: No route to host", "unreachable": true}
fatal: [192.168.56.108]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to the host via ssh: ssh: connect to h
 port 22: No route to host", "unreachable": true}

TASK [update repository index] *****************************************************************************
skipping: [192.168.56.109]

TASK [install apache2 package] *****************************************************************************
skipping: [192.168.56.109]

TASK [add PHP support for apache] **************************************************************************
skipping: [192.168.56.109]

TASK [update repository index] *****************************************************************************
ok: [192.168.56.109]

TASK [install apache2 package] *****************************************************************************
changed: [192.168.56.109]

TASK [add PHP support for apache] **************************************************************************
changed: [192.168.56.109]

PLAY RECAP *************************************************************************************************
192.168.56.107             : ok=0    changed=0    unreachable=1    failed=0    skipped=0    rescued=0    ignored=0
192.168.56.108             : ok=0    changed=0    unreachable=1    failed=0    skipped=0    rescued=0    ignored=0
192.168.56.109             : ok=4    changed=2    unreachable=0    failed=0    skipped=3    rescued=0    ignored=0
```

Figure 1.8: Output of the command used to execute the playbook.

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.



Unable to connect

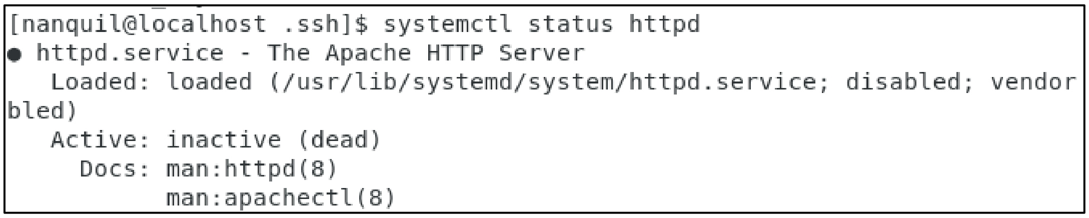
Firefox can’t establish a connection to the server at 192.168.56.109.

- The site could be temporarily unavailable or too b Try again in a few moments.
- If you are unable to load any pages, check your computer’s network connection.
- If your computer or network is protected by a fire proxy, make sure that Firefox is permitted to acce Web.

6.

Figure 1.9: Entering the IP Address on the browser resulting unsuccessful.

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.



```
[nanquil@localhost .ssh]$ systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; vendor
bled)
   Active: inactive (dead)
     Docs: man:httpd(8)
           man:apachectl(8)
```

Figure 1.10: Shows that the service is inactive.

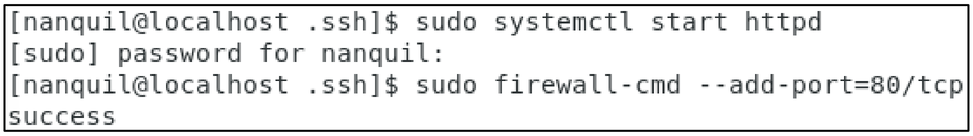
6.2 Issue the following command to start the service:

sudo systemctl start httpd

(When prompted, enter the sudo password)

sudo firewall-cmd –add-port=80/tcp

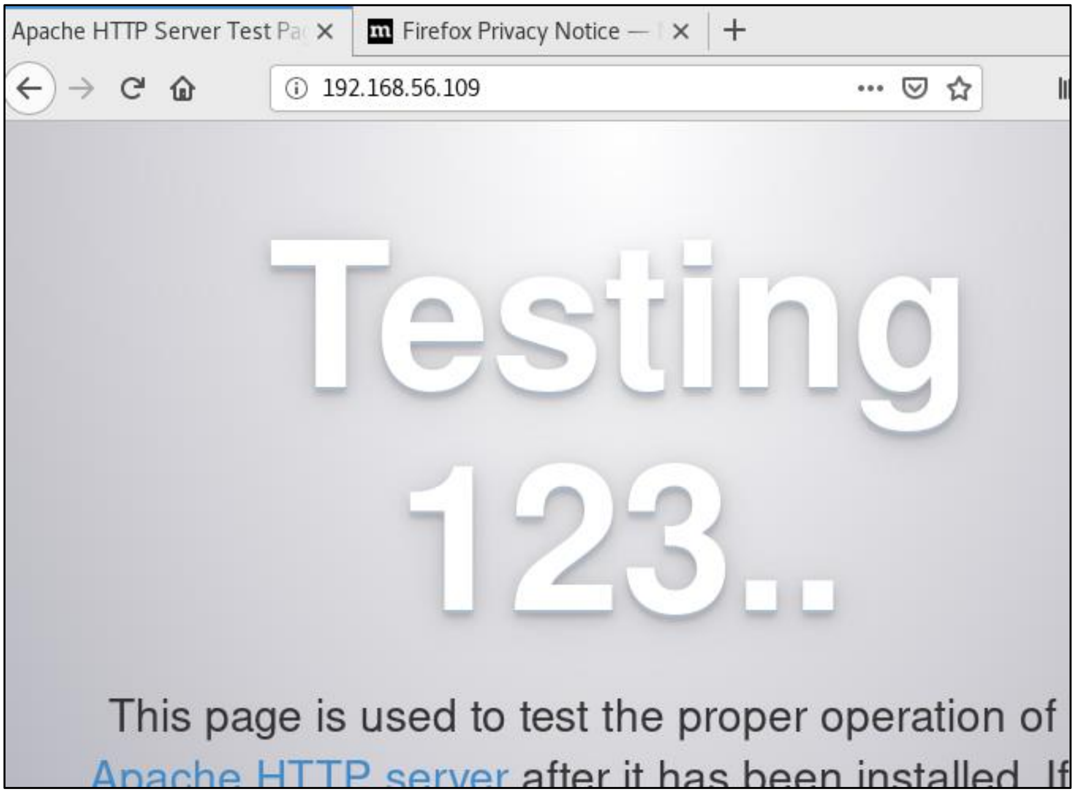
(The result should be a success)



```
[nanquil@localhost .ssh]$ sudo systemctl start httpd
[sudo] password for nanquil:
[nanquil@localhost .ssh]$ sudo firewall-cmd --add-port=80/tcp
success
```

Figure 1.11: Entered the commands to start the service resulting success.

6.3 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) Yes, it is successful.



Testing 123..

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

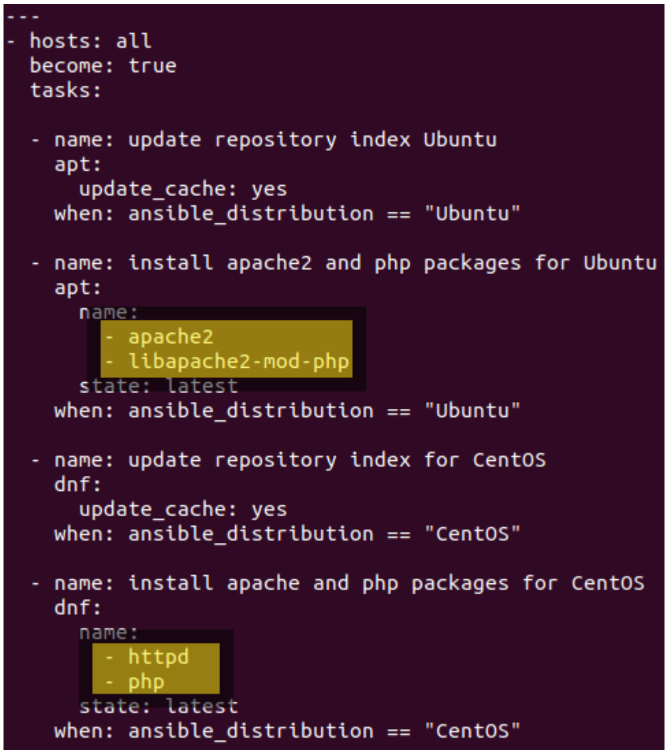
Figure 1.12: Entered again the IP Address on the browser.

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.

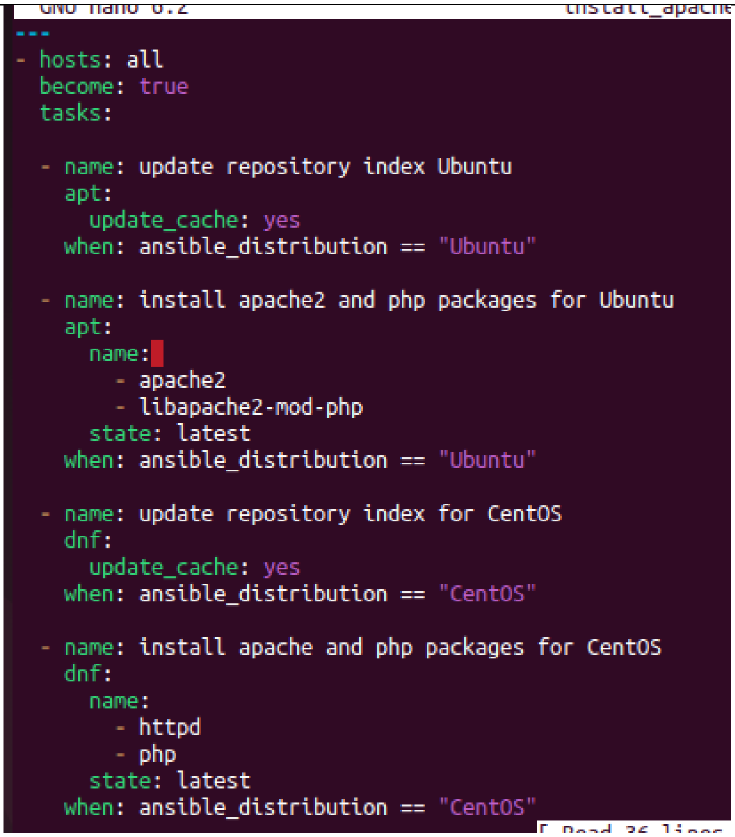
1. Edit the playbook *install_apache.yml*. Currently, we have three tasks targeting our Ubuntu machines and 3 tasks targeting our CentOS machine. Right now, we

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"
```

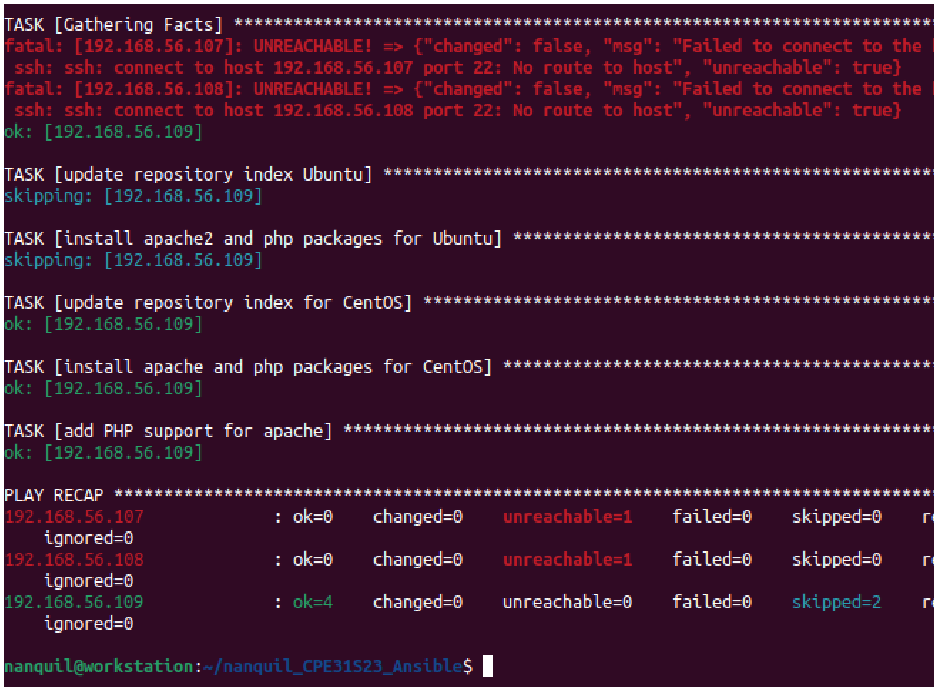
Make sure to save the file and exit.



```
  GNU nano 6.2                                  install_apache
---
- 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"
                                                  [ Read 36 lines
```

Figure 2.1: Edited the install_apache.yml playbook.

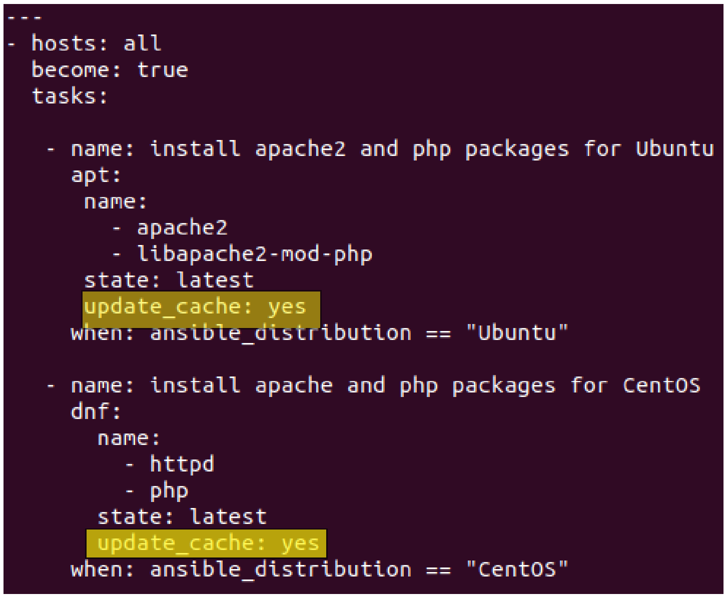
Run *ansible-playbook --ask-become-pass install_apache.yml* and describe the result.



```
TASK [Gathering Facts] *****************************************************************
fatal: [192.168.56.107]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to the
 ssh: ssh: connect to host 192.168.56.107 port 22: No route to host", "unreachable": true}
fatal: [192.168.56.108]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to the
 ssh: ssh: connect to host 192.168.56.108 port 22: No route to host", "unreachable": true}
ok: [192.168.56.109]

TASK [update repository index Ubuntu] **************************************************
skipping: [192.168.56.109]

TASK [install apache2 and php packages for Ubuntu] *************************************
skipping: [192.168.56.109]

TASK [update repository index for CentOS] **********************************************
ok: [192.168.56.109]

TASK [install apache and php packages for CentOS] **************************************
ok: [192.168.56.109]

TASK [add PHP support for apache] ******************************************************
ok: [192.168.56.109]

PLAY RECAP *****************************************************************************
192.168.56.107             : ok=0    changed=0    unreachable=1    failed=0    skipped=0    r
    ignored=0
192.168.56.108             : ok=0    changed=0    unreachable=1    failed=0    skipped=0    r
    ignored=0
192.168.56.109             : ok=4    changed=0    unreachable=0    failed=0    skipped=2    r
    ignored=0

nanquil@workstation:~/nanquil_CPE31S23_Ansible$
```

Figure 2.2: Output of the command used to execute the playbook.

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:

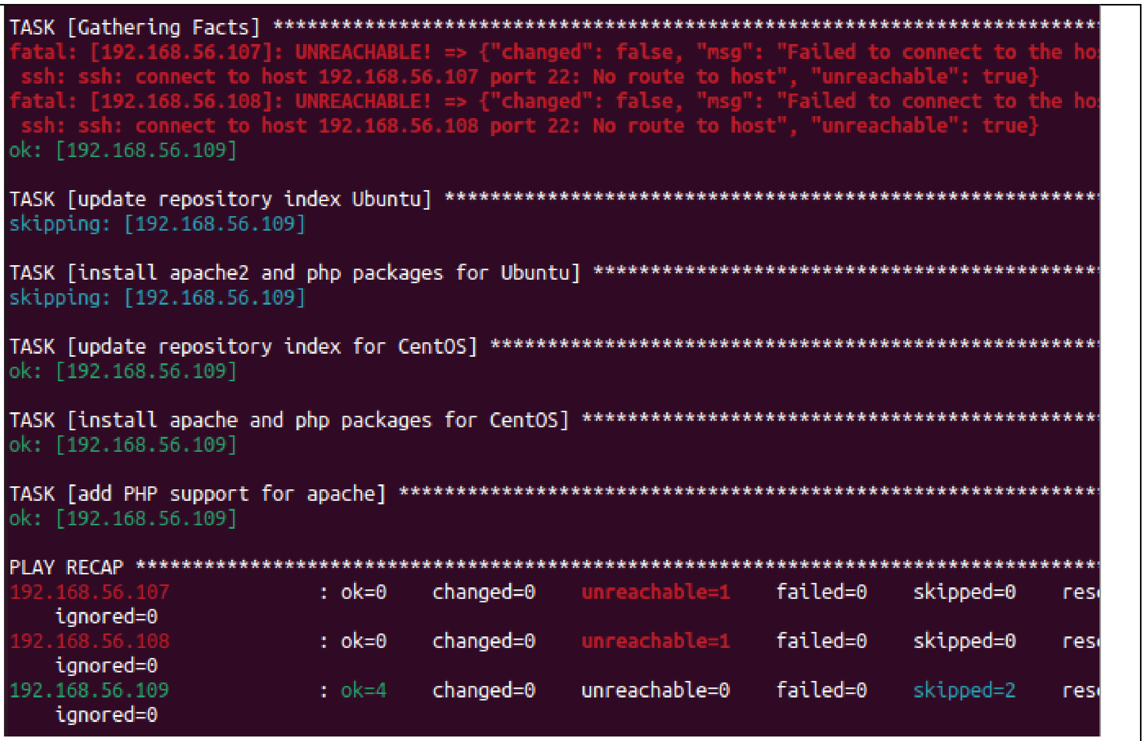


```
---
- hosts: all
  become: true
  tasks:

   - 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"
```

Figure 2.3: Edited the install_apache.yml playbook.

Make sure to save the file and exit.

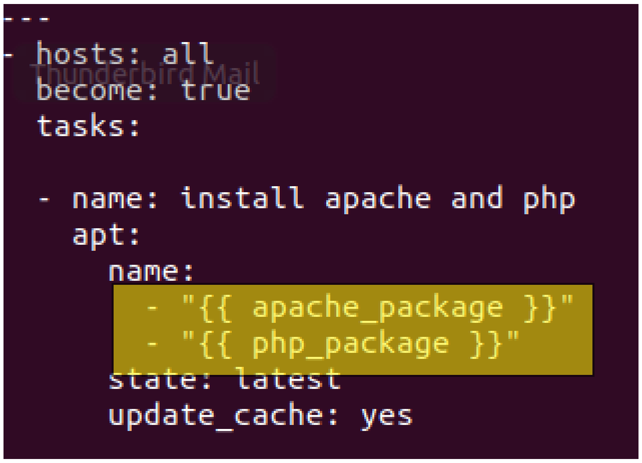
Run *ansible-playbook --ask-become-pass install_apache.yml* and describe the result. The result after entering the command is just the same in the previous step.



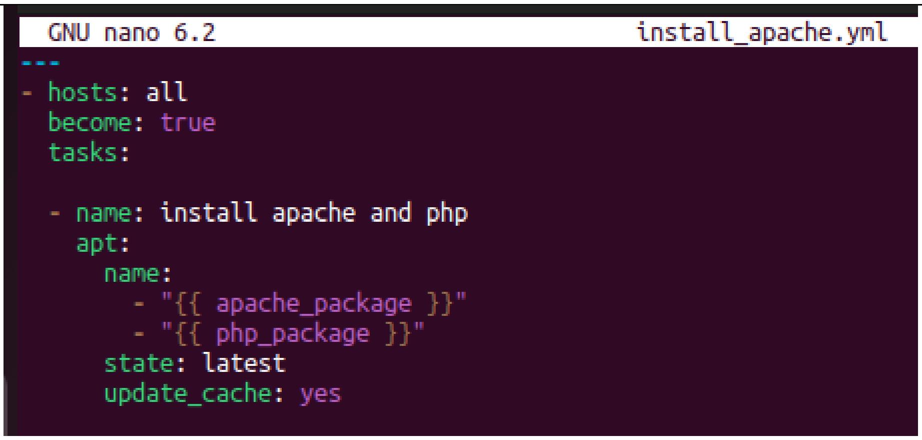
```
TASK [Gathering Facts] ************************************************************
fatal: [192.168.56.107]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to the ho
 ssh: ssh: connect to host 192.168.56.107 port 22: No route to host", "unreachable": true}
fatal: [192.168.56.108]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to the ho
 ssh: ssh: connect to host 192.168.56.108 port 22: No route to host", "unreachable": true}
ok: [192.168.56.109]

TASK [update repository index Ubuntu] *********************************************
skipping: [192.168.56.109]

TASK [install apache2 and php packages for Ubuntu] ********************************
skipping: [192.168.56.109]

TASK [update repository index for CentOS] *****************************************
ok: [192.168.56.109]

TASK [install apache and php packages for CentOS] *********************************
ok: [192.168.56.109]

TASK [add PHP support for apache] *************************************************
ok: [192.168.56.109]

PLAY RECAP ************************************************************************
192.168.56.107             : ok=0    changed=0    unreachable=1    failed=0    skipped=0    res
    ignored=0
192.168.56.108             : ok=0    changed=0    unreachable=1    failed=0    skipped=0    res
    ignored=0
192.168.56.109             : ok=4    changed=0    unreachable=0    failed=0    skipped=2    res
    ignored=0
```

Figure 2.4: Output of the command used to execute the playbook.

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.



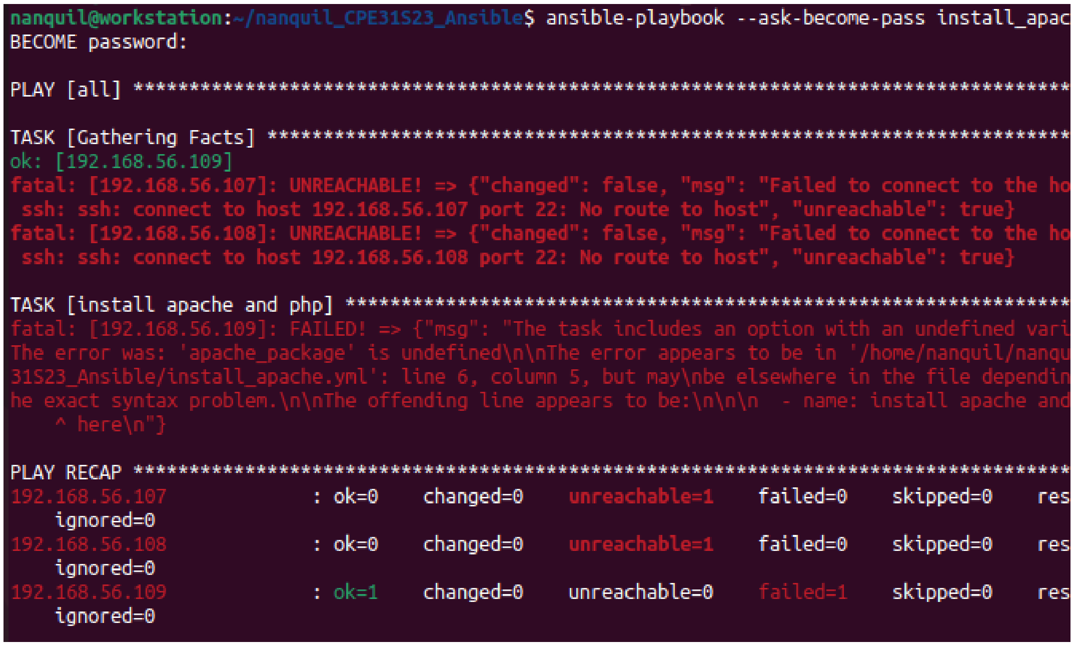
```
---
- hosts: all
  become: true
  tasks:

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



```
GNU nano 6.2                                   install_apache.yml
---
- hosts: all
  become: true
  tasks:

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

Figure 2.5: Edited the playbook.

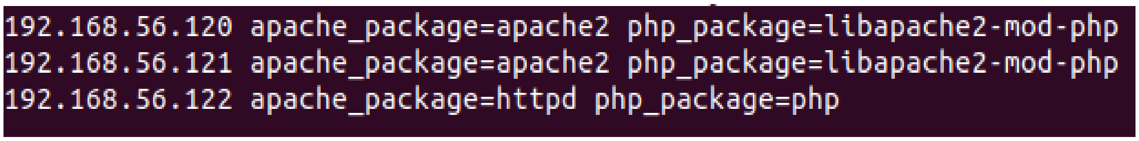
Run *ansible-playbook --ask-become-pass install_apache.yml* and describe the result.



```
nanquil@workstation:~/nanquil_CPE31S23_Ansible$ ansible-playbook --ask-become-pass install_apac
BECOME password:

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

TASK [Gathering Facts] ************************************************************************
ok: [192.168.56.109]
fatal: [192.168.56.107]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to the ho
 ssh: ssh: connect to host 192.168.56.107 port 22: No route to host", "unreachable": true}
fatal: [192.168.56.108]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to the ho
 ssh: ssh: connect to host 192.168.56.108 port 22: No route to host", "unreachable": true}

TASK [install apache and php] *****************************************************************
fatal: [192.168.56.109]: FAILED! => {"msg": "The task includes an option with an undefined vari
The error was: 'apache_package' is undefined\n\nThe error appears to be in '/home/nanquil/nanqu
31S23_Ansible/install_apache.yml': line 6, column 5, but may\nbe elsewhere in the file dependin
he exact syntax problem.\n\nThe offending line appears to be:\n\n\n  - name: install apache and
    ^ here\n"}

PLAY RECAP ************************************************************************************
192.168.56.107             : ok=0    changed=0    unreachable=1    failed=0    skipped=0    res
    ignored=0
192.168.56.108             : ok=0    changed=0    unreachable=1    failed=0    skipped=0    res
    ignored=0
192.168.56.109             : ok=1    changed=0    unreachable=0    failed=1    skipped=0    res
    ignored=0
```

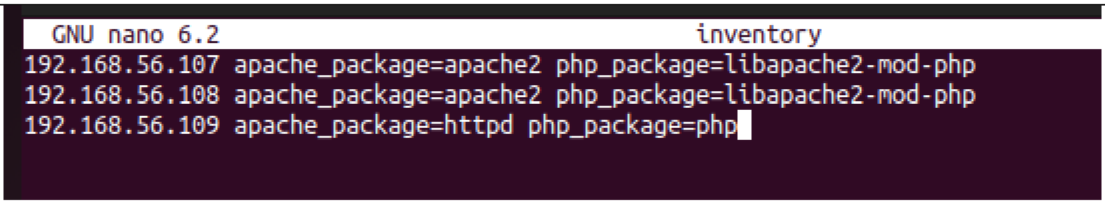
Figure 2.6: After entering the command. It shows that it is not successful.

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
```

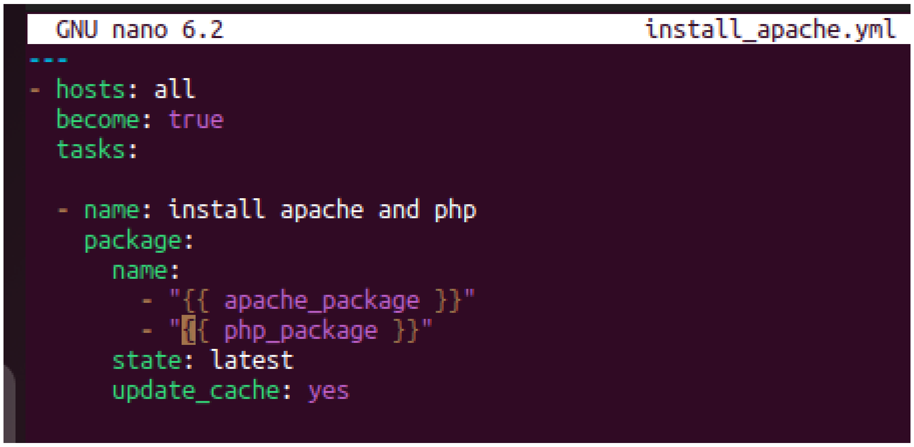
Make sure to save the *inventory* file and exit.



```
  GNU nano 6.2                                   inventory
192.168.56.107 apache_package=apache2 php_package=libapache2-mod-php
192.168.56.108 apache_package=apache2 php_package=libapache2-mod-php
192.168.56.109 apache_package=httpd php_package=php
```

Figure 2.7: Edited the inventory file using the configuration given.

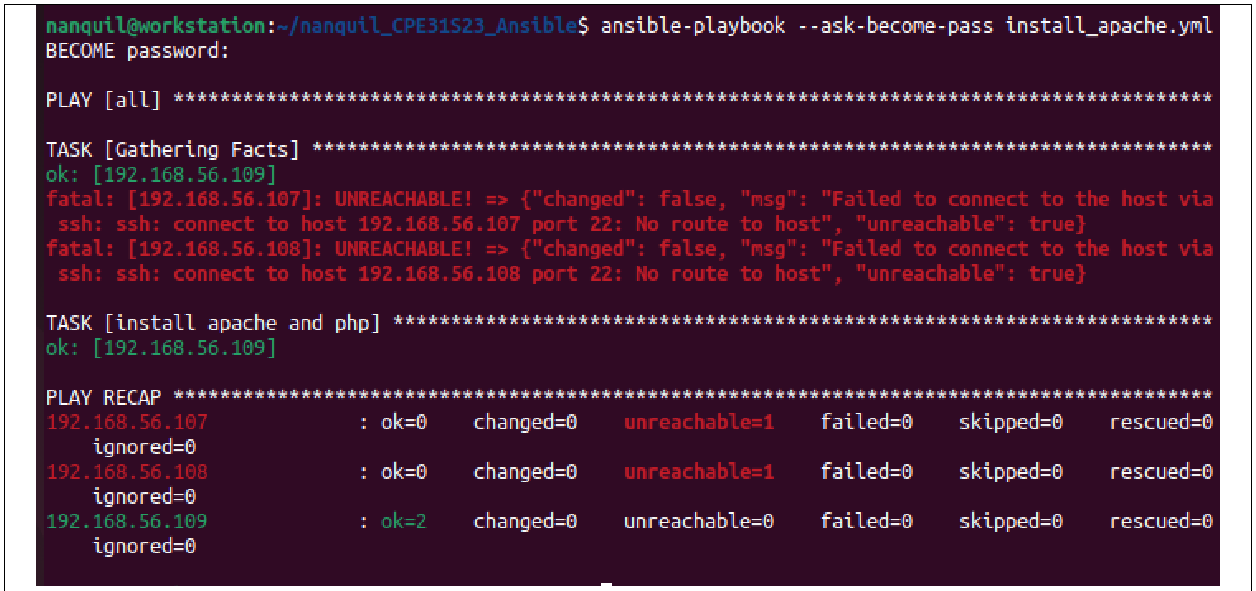
Finally, we still have one more thing to change in our *install_apache.yml* file. In task 2.3, you may notice that the package is assign as apt, which will not run in CentOS. Replace the *apt* with *package*. Package is a module in ansible that is generic, which is going to use whatever package manager 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
```

Figure 2.8: Edited the install_apache.yml playbook by changing the apt to package.

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



```
nanquil@workstation:~/nanquil_CPE31S23_Ansible$ ansible-playbook --ask-become-pass install_apache.yml
BECOME password:

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

TASK [Gathering Facts] ***********************************************************************
ok: [192.168.56.109]
fatal: [192.168.56.107]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to the host via
 ssh: ssh: connect to host 192.168.56.107 port 22: No route to host", "unreachable": true}
fatal: [192.168.56.108]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to the host via
 ssh: ssh: connect to host 192.168.56.108 port 22: No route to host", "unreachable": true}

TASK [install apache and php] ****************************************************************
ok: [192.168.56.109]

PLAY RECAP ***********************************************************************************
192.168.56.107             : ok=0    changed=0    unreachable=1    failed=0    skipped=0    rescued=0
    ignored=0
192.168.56.108             : ok=0    changed=0    unreachable=1    failed=0    skipped=0    rescued=0
    ignored=0
192.168.56.109             : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0
    ignored=0
```

Figure 2.9: Executed the playbook on it is now successful.

Supplementary Activity:

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

Reflections:

Answer the following:

1. Why do you think refactoring of playbook codes is important? Refactoring of playbook codes is important because it is easier to understand and to add new features. Refactoring is important because it simplifies the additional code and it does not change the output or behavior of the program.
2. When do we use the “when” command in playbook? We use the “when” command in playbook if we will add multiple conditions in the process.