Name: Torrecampo, Juan Piolo S.	**Date Performed:** Sep 13, 2022
Course/Section: CPE 232 / CPE31S22	**Date Submitted:** Sep 13, 2022
Instructor: Dr. 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.

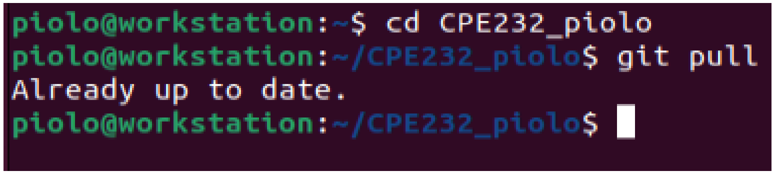
Requirement:

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?

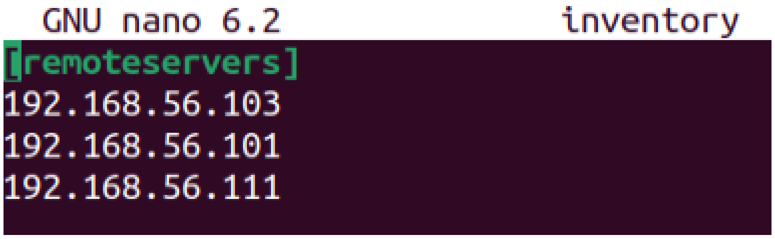
    - **It pulls the git repository and gets the latest update. It does not change anything because we did not add anything into our previous repository. That is why the command displays it is “Already up to date”.**



```
piolo@workstation:~$ cd CPE232_piolo
piolo@workstation:~/CPE232_piolo$ git pull
Already up to date.
piolo@workstation:~/CPE232_piolo$ 
```

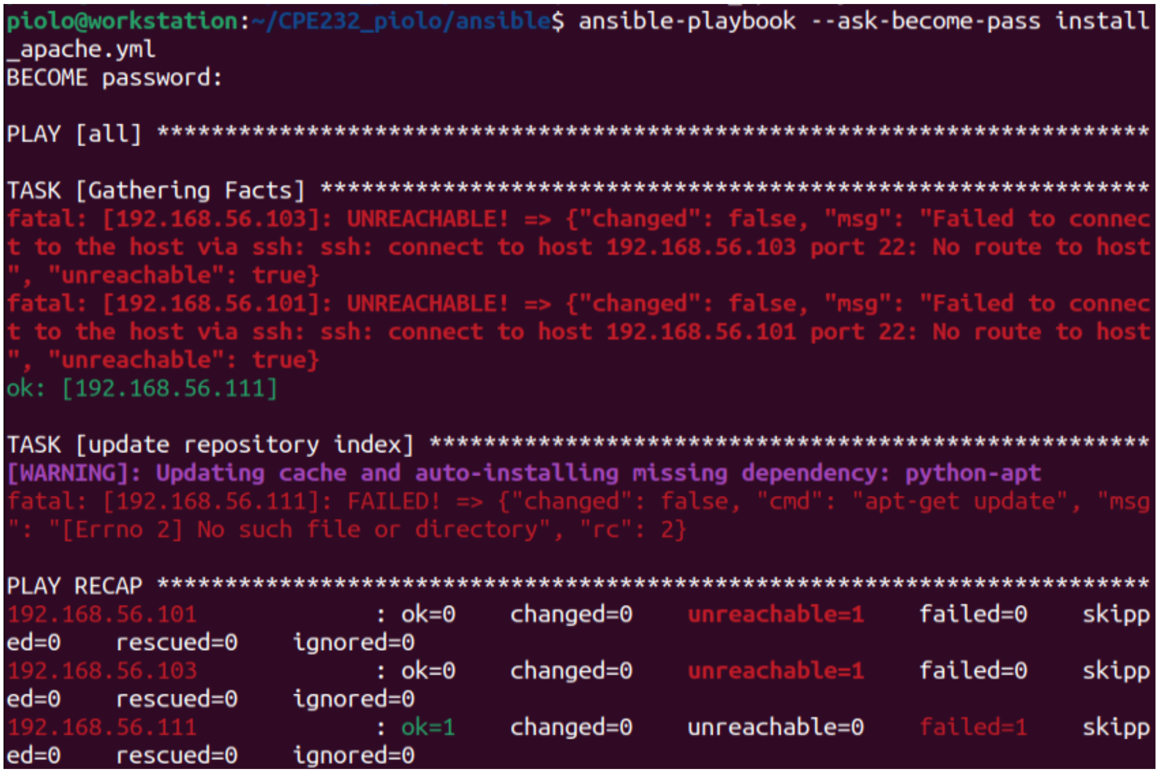
2. Edit the inventory file and add the IP address of the Centos VM. Issue the command we used to execute the playbook (the one we used in the last activity): *ansible-playbook --ask-become-pass install_apache.yml*. After executing this command, you may notice that it did not become successful in the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

Configuring Inventory



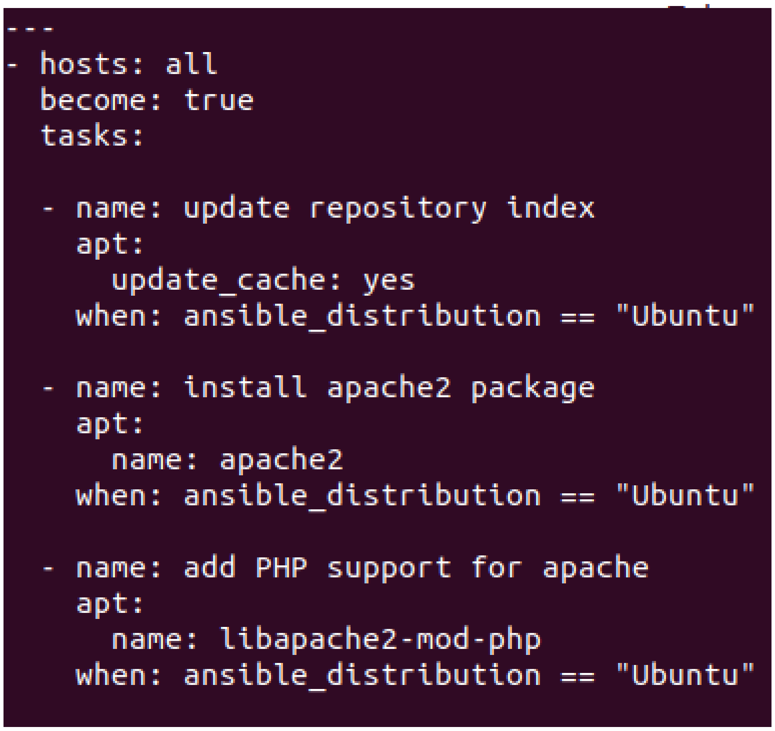
```
  GNU nano 6.2                    inventory
[remoteservers]
192.168.56.103
192.168.56.101
192.168.56.111
```

Executing "ansible-playbook –ask-become-pass install_apache.yml"



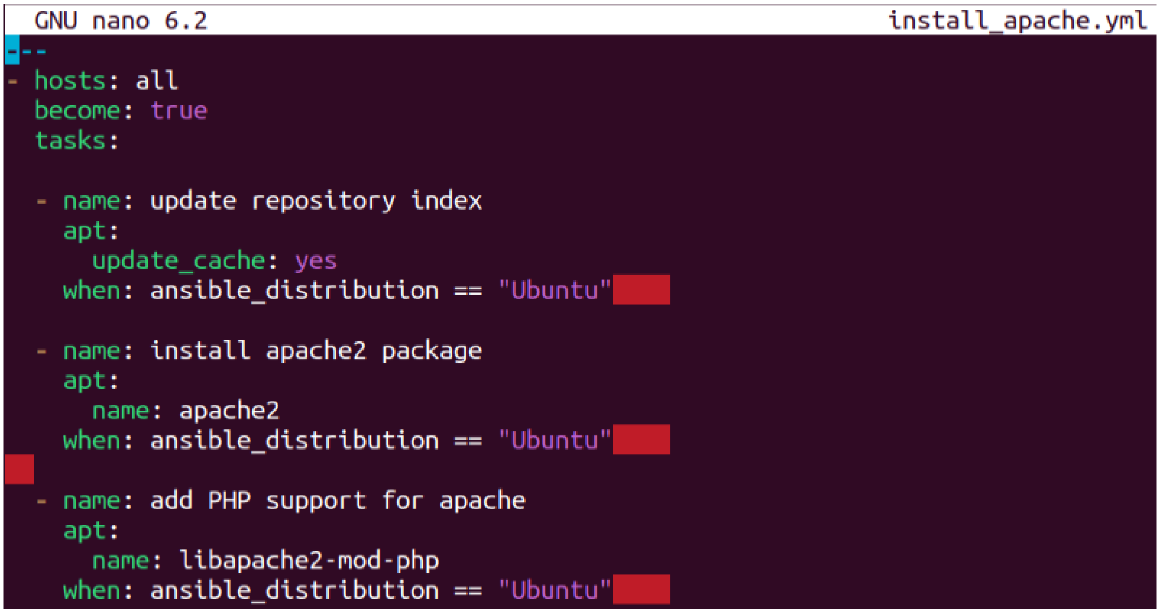
```
piolo@workstation:~/CPE232_piolo/ansible$ ansible-playbook --ask-become-pass install
_apache.yml
BECOME password:

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

TASK [Gathering Facts] ***********************************************************
fatal: [192.168.56.103]: UNREACHABLE! => {"changed": false, "msg": "Failed to connec
t to the host via ssh: ssh: connect to host 192.168.56.103 port 22: No route to host
", "unreachable": true}
fatal: [192.168.56.101]: UNREACHABLE! => {"changed": false, "msg": "Failed to connec
t to the host via ssh: ssh: connect to host 192.168.56.101 port 22: No route to host
", "unreachable": true}
ok: [192.168.56.111]

TASK [update repository index] ***************************************************
[WARNING]: Updating cache and auto-installing missing dependency: python-apt
fatal: [192.168.56.111]: FAILED! => {"changed": false, "cmd": "apt-get update", "msg
": "[Errno 2] No such file or directory", "rc": 2}

PLAY RECAP ***********************************************************************
192.168.56.101             : ok=0    changed=0    unreachable=1    failed=0    skipp
ed=0    rescued=0    ignored=0
192.168.56.103             : ok=0    changed=0    unreachable=1    failed=0    skipp
ed=0    rescued=0    ignored=0
192.168.56.111             : ok=1    changed=0    unreachable=0    failed=1    skipp
ed=0    rescued=0    ignored=0
```

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



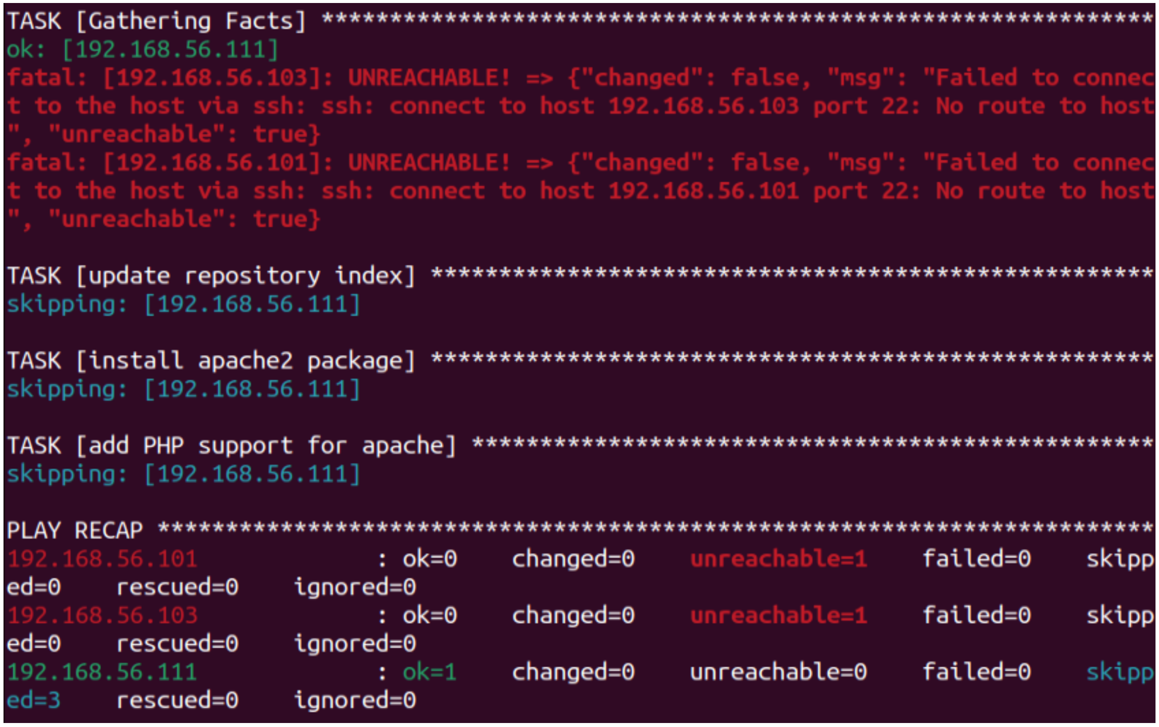
```
---
- hosts: all
  become: true
  tasks:

  - 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.



```
  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"
```

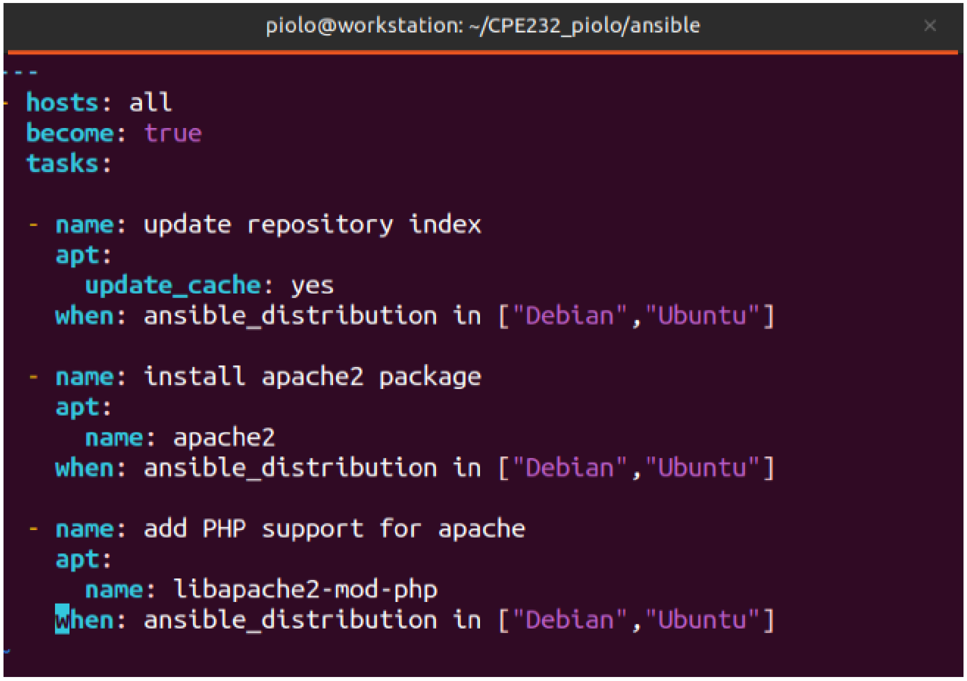
Run *ansible-playbook --ask-become-pass install_apache.yml* and describe the result.



```
TASK [Gathering Facts] *************************************************************
ok: [192.168.56.111]
fatal: [192.168.56.103]: UNREACHABLE! => {"changed": false, "msg": "Failed to connec
t to the host via ssh: ssh: connect to host 192.168.56.103 port 22: No route to host
", "unreachable": true}
fatal: [192.168.56.101]: UNREACHABLE! => {"changed": false, "msg": "Failed to connec
t to the host via ssh: ssh: connect to host 192.168.56.101 port 22: No route to host
", "unreachable": true}

TASK [update repository index] *****************************************************
skipping: [192.168.56.111]

TASK [install apache2 package] *****************************************************
skipping: [192.168.56.111]

TASK [add PHP support for apache] **************************************************
skipping: [192.168.56.111]

PLAY RECAP *************************************************************************
192.168.56.101             : ok=0    changed=0    unreachable=1    failed=0    skipp
ed=0    rescued=0    ignored=0
192.168.56.103             : ok=0    changed=0    unreachable=1    failed=0    skipp
ed=0    rescued=0    ignored=0
192.168.56.111             : ok=1    changed=0    unreachable=0    failed=0    skipp
ed=3    rescued=0    ignored=0
```

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

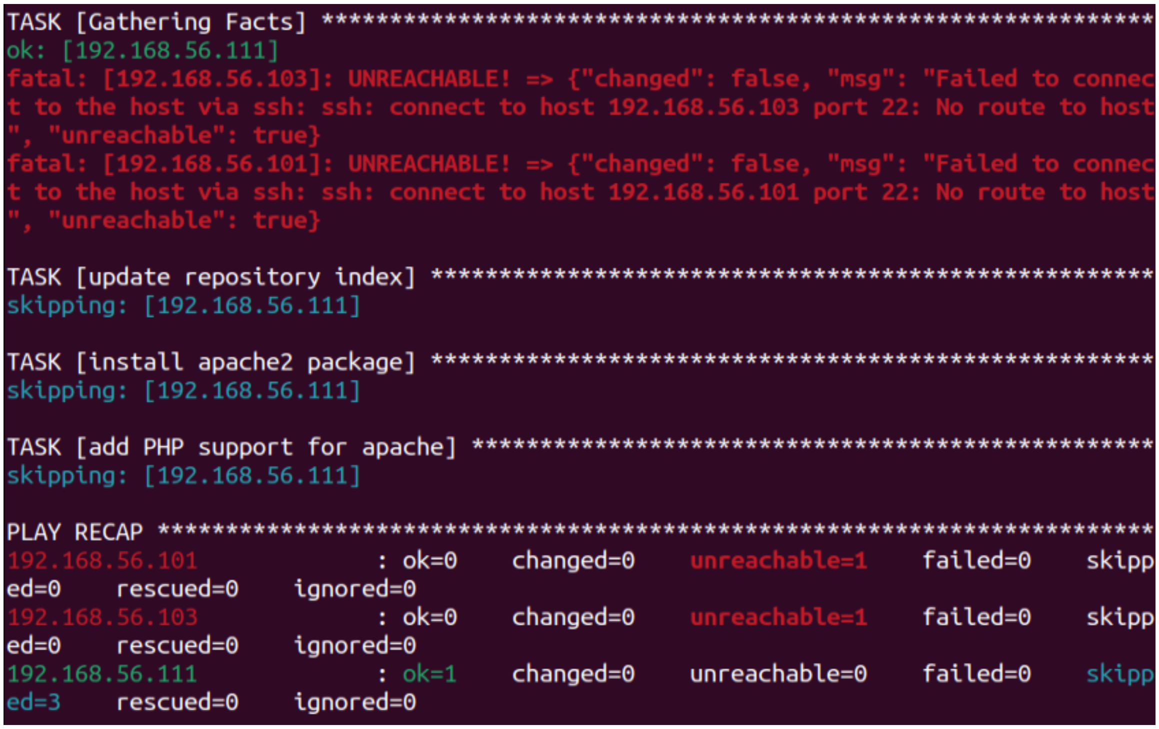
```
-   name: update repository index
    apt:
      update_cache: yes
    when: ansible_distribution in [“Debian”, “Ubuntu]
```



piolo@workstation: ~/CPE232_piolo/ansible

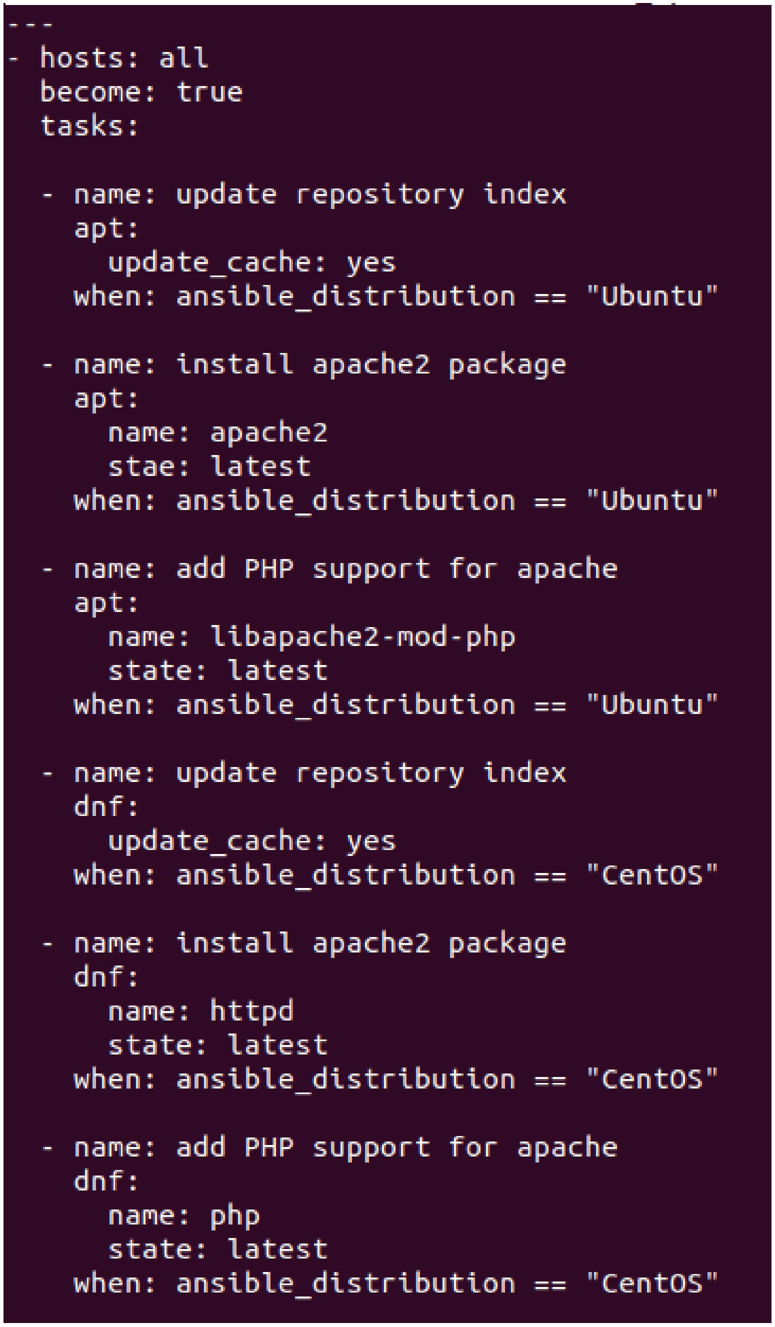
```
---
- hosts: all
  become: true
  tasks:

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

  - name: install apache2 package
    apt:
      name: apache2
    when: ansible_distribution in ["Debian","Ubuntu"]

  - name: add PHP support for apache
    apt:
      name: libapache2-mod-php
    when: ansible_distribution in ["Debian","Ubuntu"]
```

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



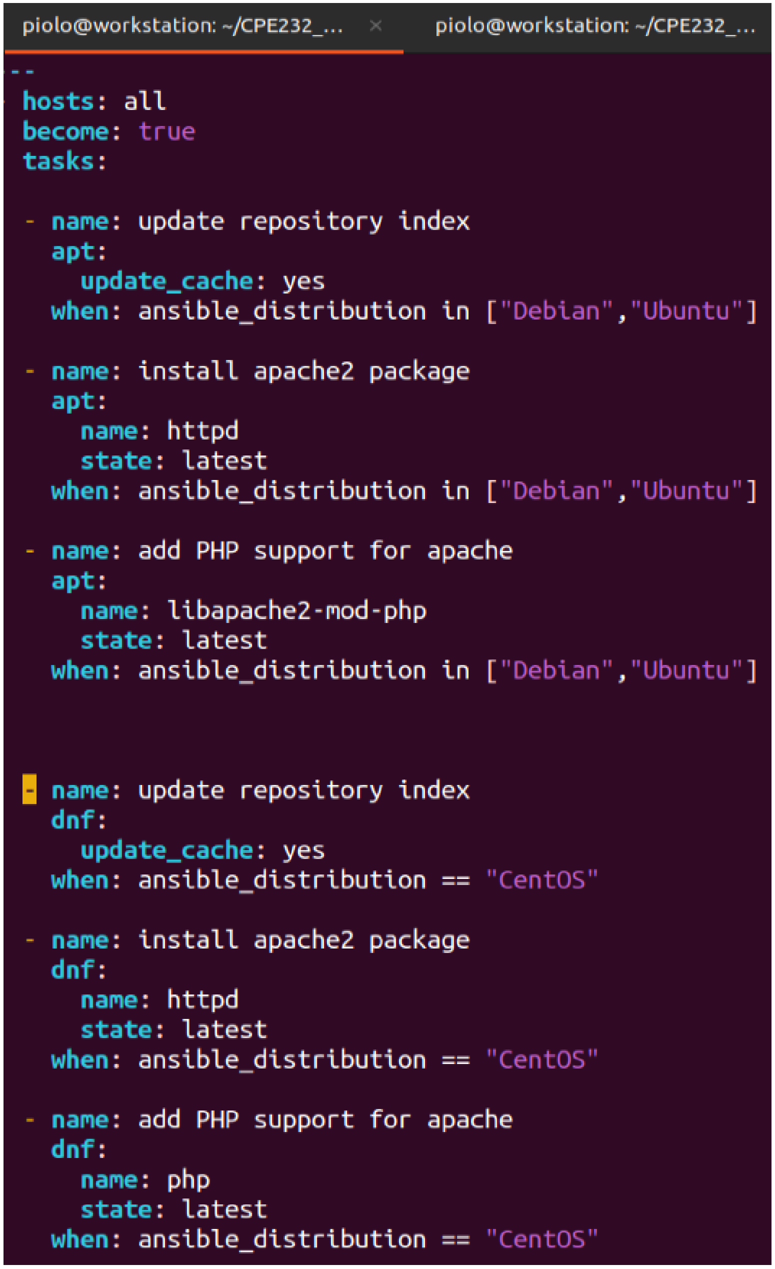
```
TASK [Gathering Facts] *************************************************************
ok: [192.168.56.111]
fatal: [192.168.56.103]: UNREACHABLE! => {"changed": false, "msg": "Failed to connec
t to the host via ssh: ssh: connect to host 192.168.56.103 port 22: No route to host
", "unreachable": true}
fatal: [192.168.56.101]: UNREACHABLE! => {"changed": false, "msg": "Failed to connec
t to the host via ssh: ssh: connect to host 192.168.56.101 port 22: No route to host
", "unreachable": true}

TASK [update repository index] *****************************************************
skipping: [192.168.56.111]

TASK [install apache2 package] *****************************************************
skipping: [192.168.56.111]

TASK [add PHP support for apache] **************************************************
skipping: [192.168.56.111]

PLAY RECAP *************************************************************************
192.168.56.101             : ok=0    changed=0    unreachable=1    failed=0    skipp
ed=0    rescued=0    ignored=0
192.168.56.103             : ok=0    changed=0    unreachable=1    failed=0    skipp
ed=0    rescued=0    ignored=0
192.168.56.111             : ok=1    changed=0    unreachable=0    failed=0    skipp
ed=3    rescued=0    ignored=0
```

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



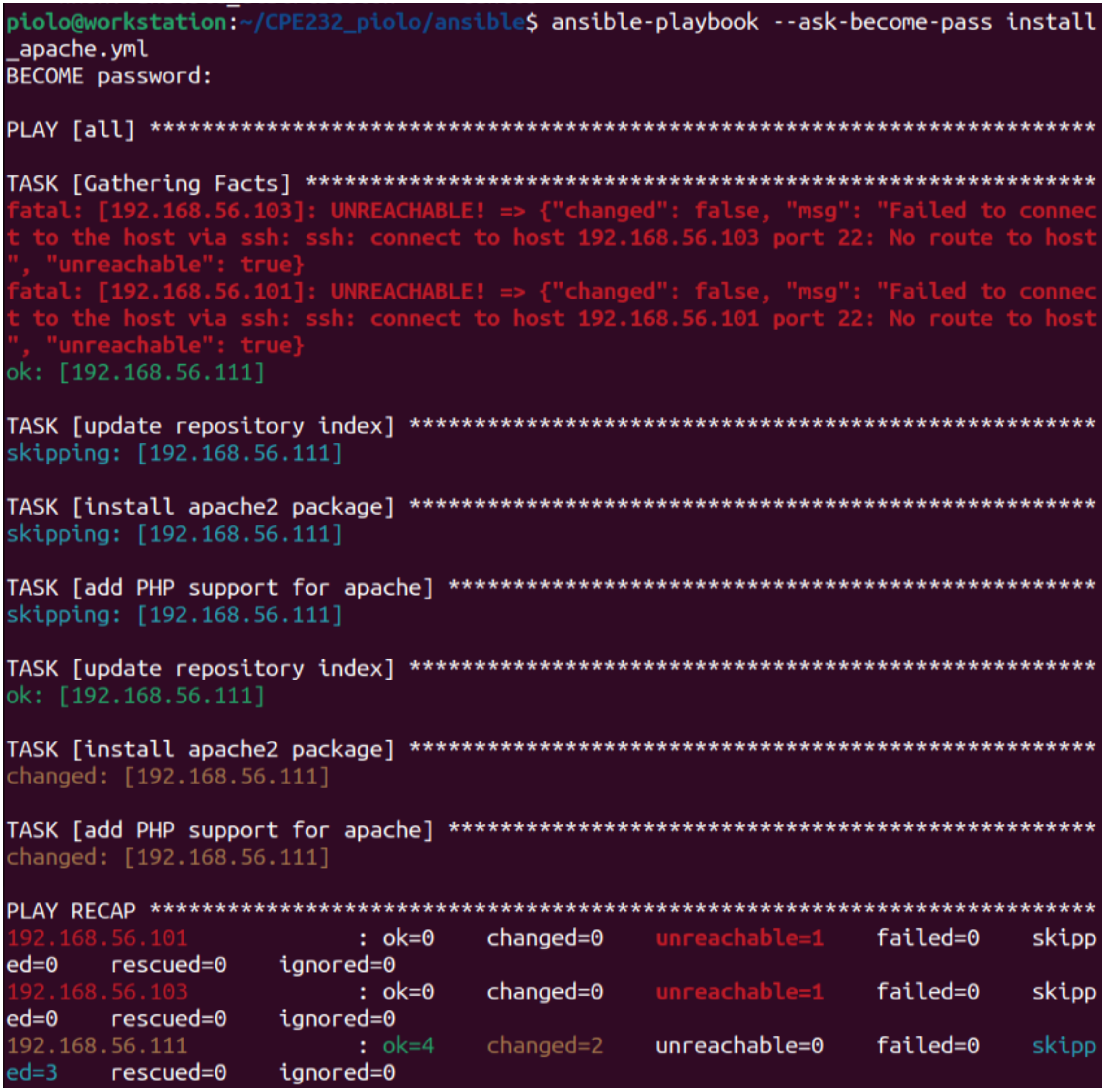
```
---
- hosts: all
  become: true
  tasks:

  - 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.


piolo@workstation: ~/CPE232_...
piolo@workstation: ~/CPE232_...

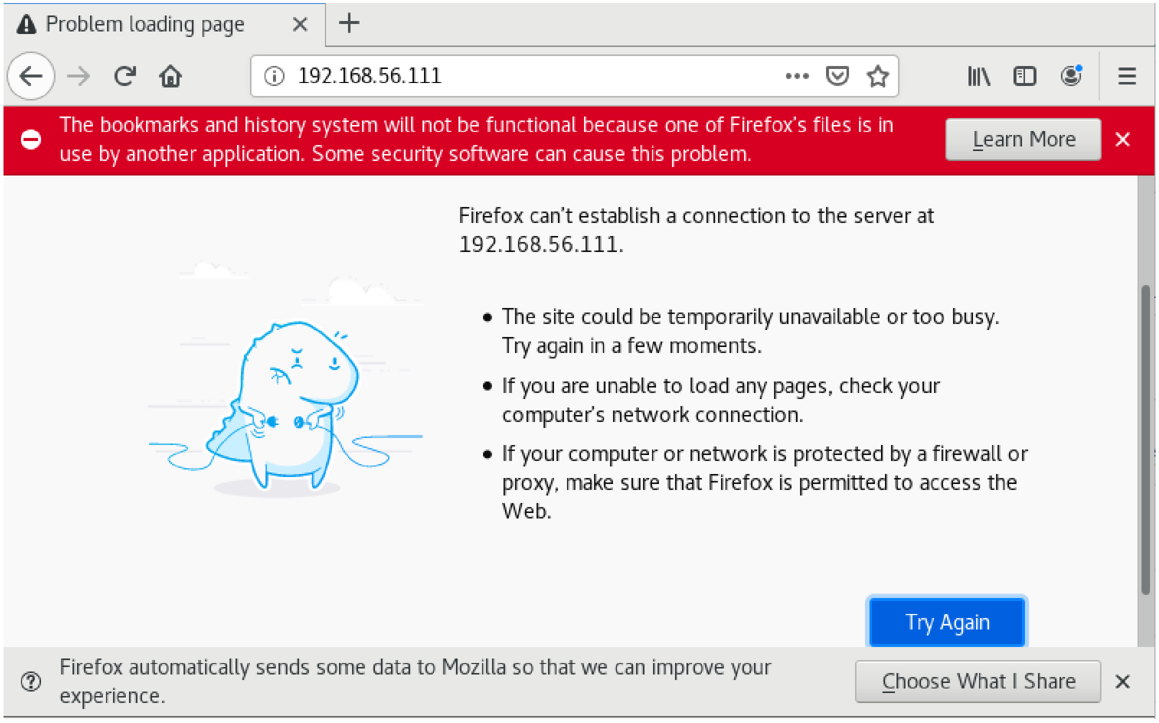
```
--
- hosts: all
  become: true
  tasks:

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

  - name: install apache2 package
    apt:
      name: httpd
      state: latest
    when: ansible_distribution in ["Debian","Ubuntu"]

  - name: add PHP support for apache
    apt:
      name: libapache2-mod-php
      state: latest
    when: ansible_distribution in ["Debian","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"
```

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



```
piolo@workstation:~/CPE232_piolo/ansible$ ansible-playbook --ask-become-pass install
_apache.yml
BECOME password:

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

TASK [Gathering Facts] *********************************************************
fatal: [192.168.56.103]: UNREACHABLE! => {"changed": false, "msg": "Failed to connec
t to the host via ssh: ssh: connect to host 192.168.56.103 port 22: No route to host
", "unreachable": true}
fatal: [192.168.56.101]: UNREACHABLE! => {"changed": false, "msg": "Failed to connec
t to the host via ssh: ssh: connect to host 192.168.56.101 port 22: No route to host
", "unreachable": true}
ok: [192.168.56.111]

TASK [update repository index] *************************************************
skipping: [192.168.56.111]

TASK [install apache2 package] *************************************************
skipping: [192.168.56.111]

TASK [add PHP support for apache] **********************************************
skipping: [192.168.56.111]

TASK [update repository index] *************************************************
ok: [192.168.56.111]

TASK [install apache2 package] *************************************************
changed: [192.168.56.111]

TASK [add PHP support for apache] **********************************************
changed: [192.168.56.111]

PLAY RECAP *********************************************************************
192.168.56.101             : ok=0    changed=0    unreachable=1    failed=0    skipp
ed=0    rescued=0    ignored=0
192.168.56.103             : ok=0    changed=0    unreachable=1    failed=0    skipp
ed=0    rescued=0    ignored=0
192.168.56.111             : ok=4    changed=2    unreachable=0    failed=0    skipp
ed=3    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.



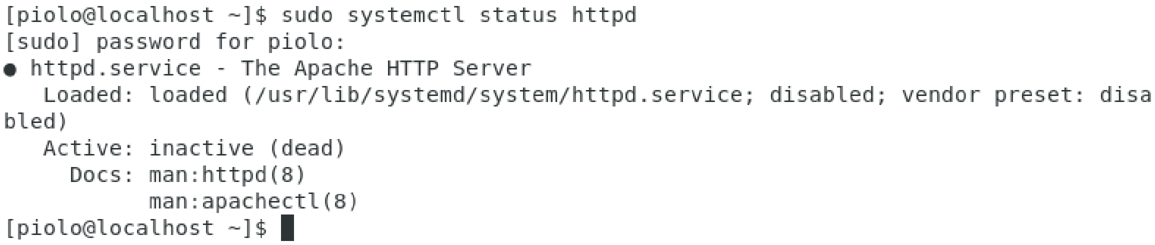
The bookmarks and history system will not be functional because one of Firefox's files is in use by another application. Some security software can cause this problem.

Firefox can't establish a connection to the server at 192.168.56.111.

- The site could be temporarily unavailable or too busy. 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 firewall or proxy, make sure that Firefox is permitted to access the Web.

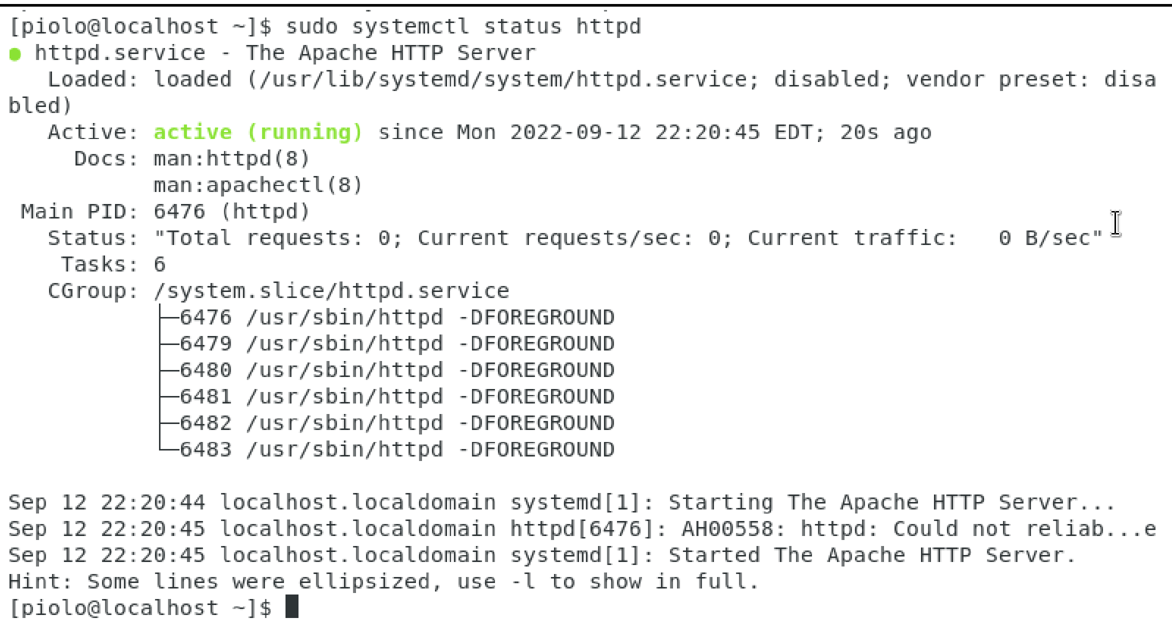
Firefox automatically sends some data to Mozilla so that we can improve your experience.

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.



```
[piolo@localhost ~]$ sudo systemctl status httpd
[sudo] password for piolo:
● 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)
[piolo@localhost ~]$ █
```

5.2 Issue the following command to start the service:
sudo systemctl start httpd

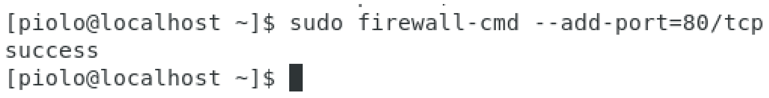


```
[piolo@localhost ~]$ sudo systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; vendor preset: disa
bled)
   Active: active (running) since Mon 2022-09-12 22:20:45 EDT; 20s ago
     Docs: man:httpd(8)
           man:apachectl(8)
 Main PID: 6476 (httpd)
   Status: "Total requests: 0; Current requests/sec: 0; Current traffic:   0 B/sec"
    Tasks: 6
   CGroup: /system.slice/httpd.service
           ├─6476 /usr/sbin/httpd -DFOREGROUND
           ├─6479 /usr/sbin/httpd -DFOREGROUND
           ├─6480 /usr/sbin/httpd -DFOREGROUND
           ├─6481 /usr/sbin/httpd -DFOREGROUND
           ├─6482 /usr/sbin/httpd -DFOREGROUND
           └─6483 /usr/sbin/httpd -DFOREGROUND

Sep 12 22:20:44 localhost.localdomain systemd[1]: Starting The Apache HTTP Server...
Sep 12 22:20:45 localhost.localdomain httpd[6476]: AH00558: httpd: Could not reliab...e
Sep 12 22:20:45 localhost.localdomain systemd[1]: Started The Apache HTTP Server.
Hint: Some lines were ellipsized, use -l to show in full.
[piolo@localhost ~]$ 
```

(When prompted, enter the sudo password)

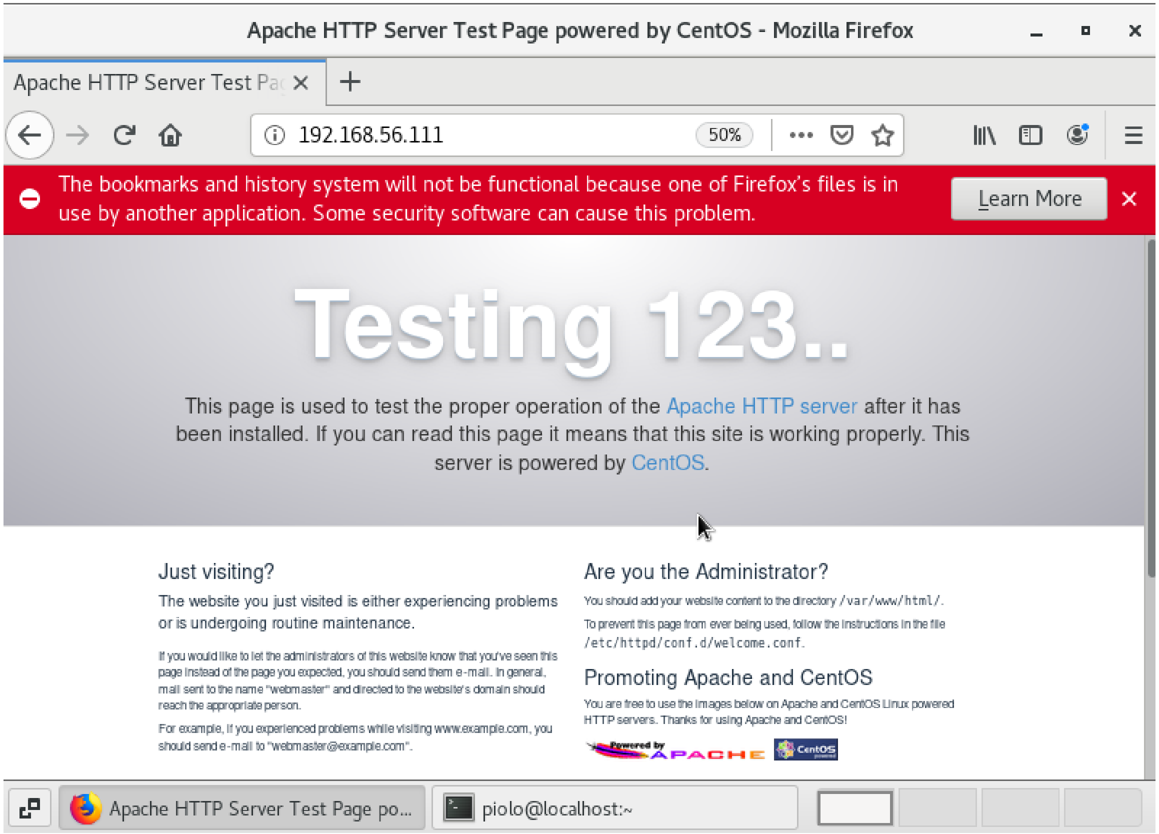
sudo firewall-cmd --add-port=80/tcp

(The result should be a success)



```
[piolo@localhost ~]$ sudo firewall-cmd --add-port=80/tcp
success
[piolo@localhost ~]$ 
```

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)



Apache HTTP Server Test Page powered by CentOS - Mozilla Firefox

The bookmarks and history system will not be functional because one of Firefox's files is in use by another application. Some security software can cause this problem.

Testing 123..

This page is used to test the proper operation of the Apache HTTP server after it has been installed. If you can read this page it means that this site is working properly. This server is powered by CentOS.

Just visiting?

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.

For example, if you experienced problems while visiting www.example.com, you should send e-mail to "webmaster@example.com".

Are you the Administrator?

You should add your website content to the directory `/var/www/html/`.

To prevent this page from ever being used, follow the instructions in the file `/etc/httpd/conf.d/welcome.conf`.

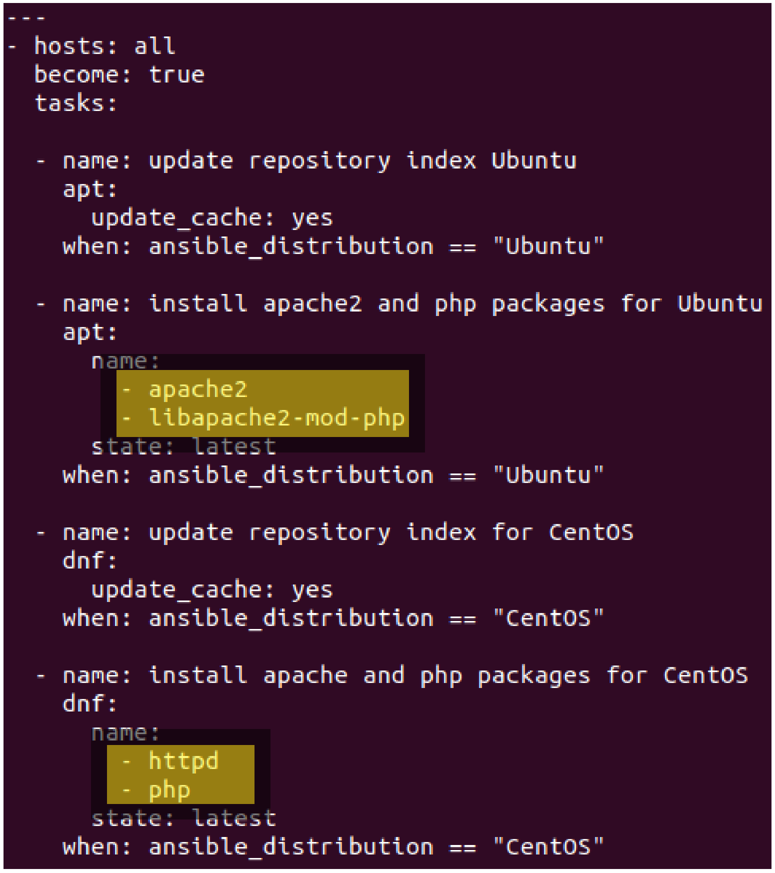
Promoting Apache and CentOS

You are free to use the images below on Apache and CentOS Linux powered HTTP servers. Thanks for using Apache and CentOS!

Task 2: Refactoring playbook

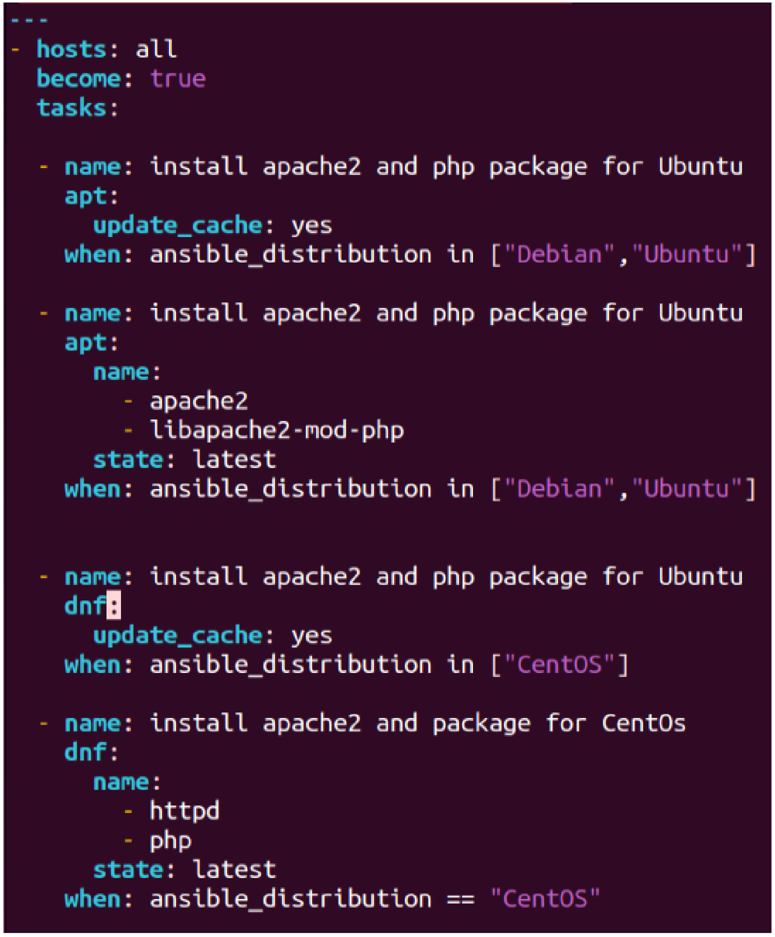
This time, we want to make sure that our playbook is efficient and that the codes are easier to read. This will also makes run ansible more quickly if it has to execute fewer tasks to do the same thing.

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:



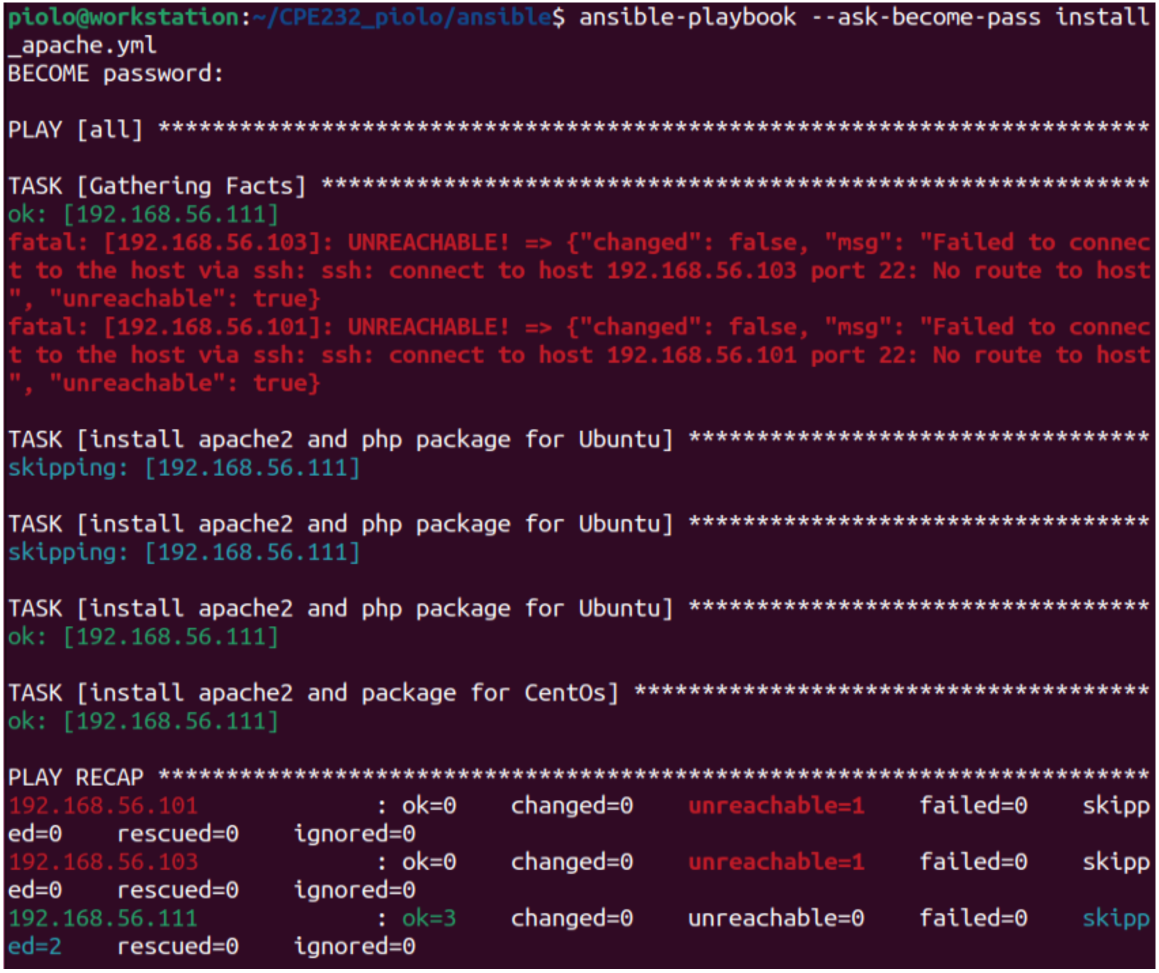
```
---
- hosts: all
  become: true
  tasks:

  - 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.



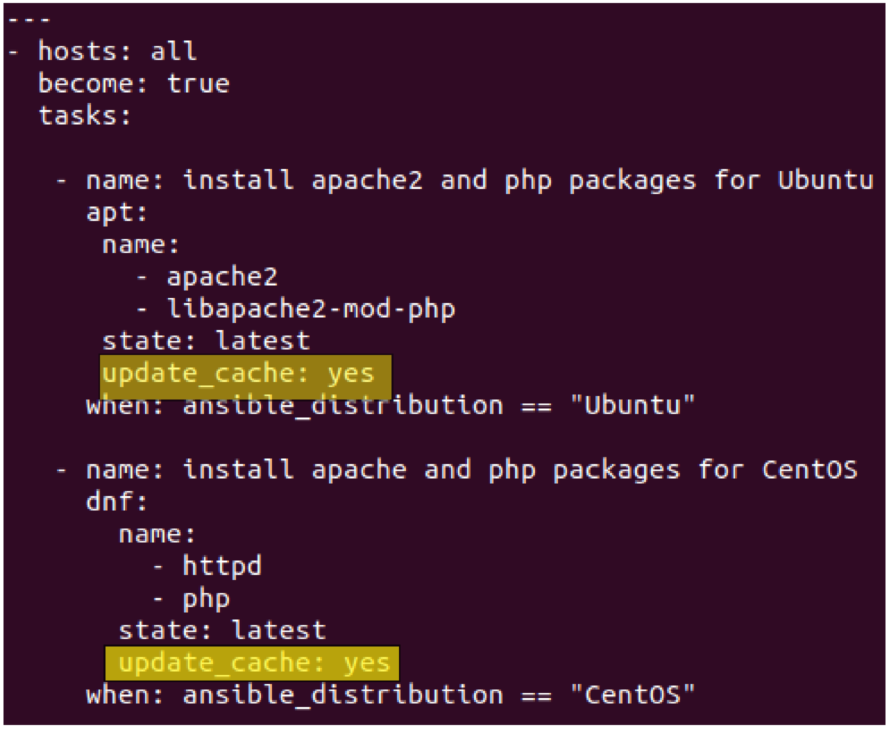
```yaml
---
- hosts: all
  become: true
  tasks:

  - name: install apache2 and php package for Ubuntu
    apt:
      update_cache: yes
    when: ansible_distribution in ["Debian","Ubuntu"]

  - name: install apache2 and php package for Ubuntu
    apt:
      name:
        - apache2
        - libapache2-mod-php
      state: latest
    when: ansible_distribution in ["Debian","Ubuntu"]


  - name: install apache2 and php package for Ubuntu
    dnf:
      update_cache: yes
    when: ansible_distribution in ["CentOS"]

  - name: install apache2 and package 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.



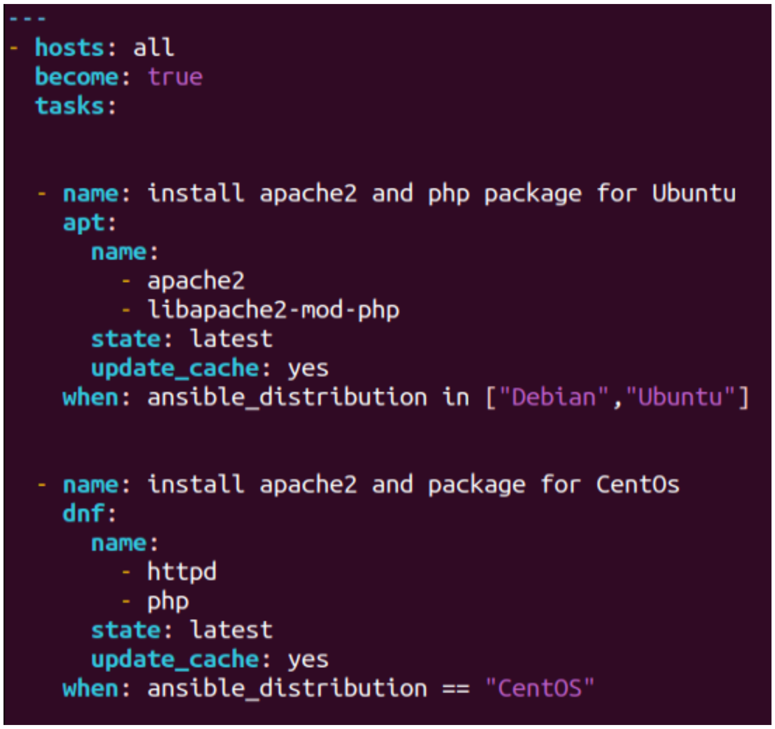
```
piolo@workstation:~/CPE232_piolo/ansible$ ansible-playbook --ask-become-pass install
_apache.yml
BECOME password:

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

TASK [Gathering Facts] *********************************************************
ok: [192.168.56.111]
fatal: [192.168.56.103]: UNREACHABLE! => {"changed": false, "msg": "Failed to connec
t to the host via ssh: ssh: connect to host 192.168.56.103 port 22: No route to host
", "unreachable": true}
fatal: [192.168.56.101]: UNREACHABLE! => {"changed": false, "msg": "Failed to connec
t to the host via ssh: ssh: connect to host 192.168.56.101 port 22: No route to host
", "unreachable": true}

TASK [install apache2 and php package for Ubuntu] ******************************
skipping: [192.168.56.111]

TASK [install apache2 and php package for Ubuntu] ******************************
skipping: [192.168.56.111]

TASK [install apache2 and php package for Ubuntu] ******************************
ok: [192.168.56.111]

TASK [install apache2 and package for CentOs] **********************************
ok: [192.168.56.111]

PLAY RECAP *********************************************************************
192.168.56.101             : ok=0    changed=0    unreachable=1    failed=0    skipp
ed=0    rescued=0    ignored=0
192.168.56.103             : ok=0    changed=0    unreachable=1    failed=0    skipp
ed=0    rescued=0    ignored=0
192.168.56.111             : ok=3    changed=0    unreachable=0    failed=0    skipp
ed=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:



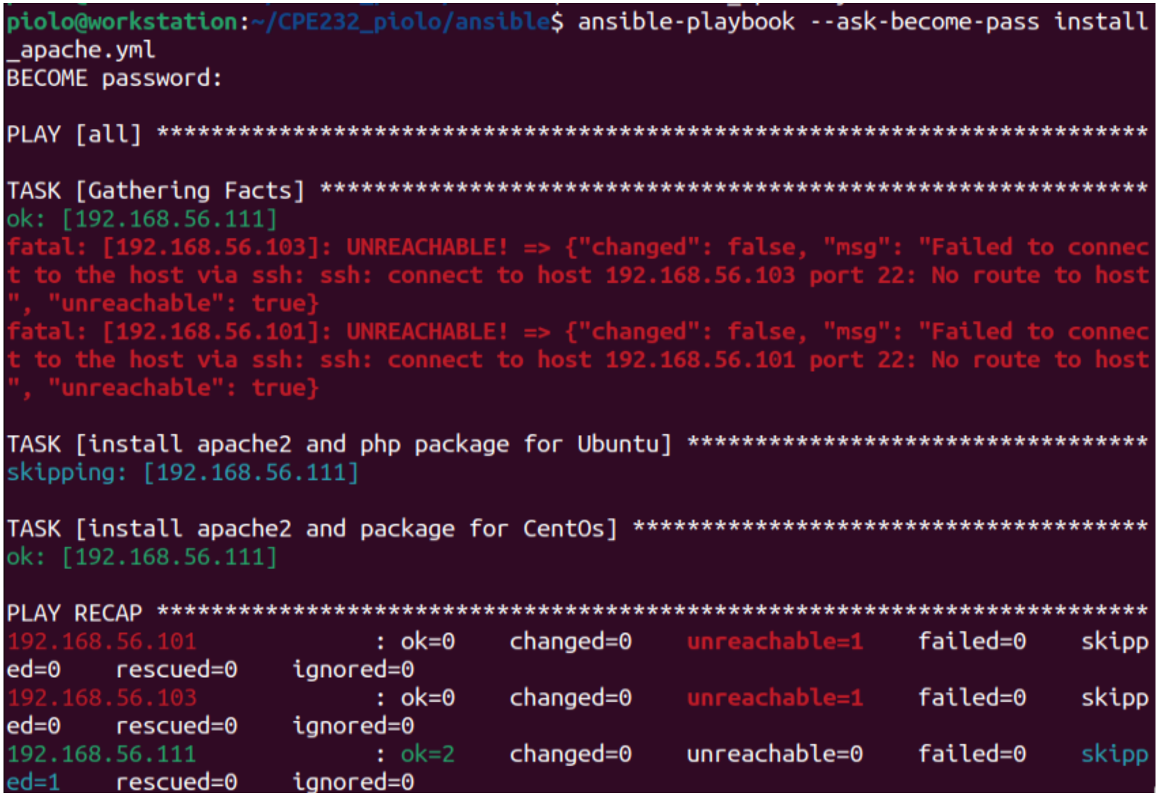
```
---
- hosts: all
  become: true
  tasks:

   - 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.



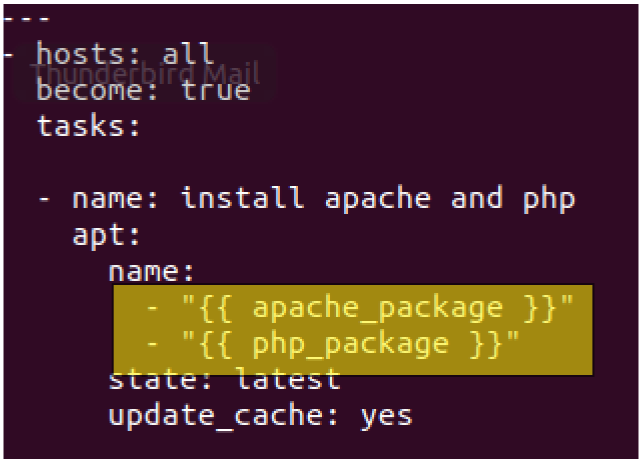
```
---
- hosts: all
  become: true
  tasks:


  - name: install apache2 and php package for Ubuntu
    apt:
      name:
        - apache2
        - libapache2-mod-php
      state: latest
      update_cache: yes
    when: ansible_distribution in ["Debian","Ubuntu"]


  - name: install apache2 and package for CentOs
    dnf:
      name:
        - httpd
        - php
      state: latest
      update_cache: yes
    when: ansible_distribution == "CentOS"
```

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

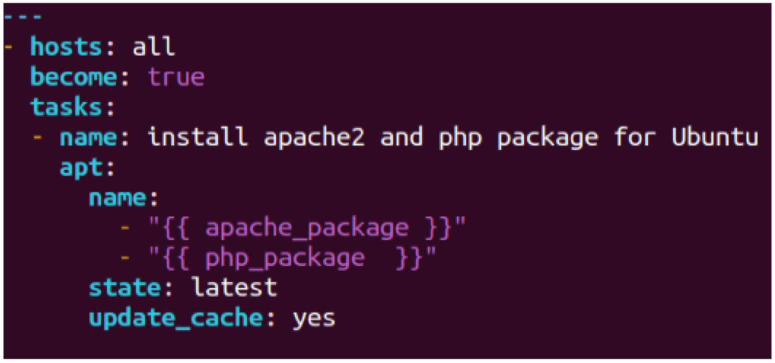


```
piolo@workstation:~/CPE232_piolo/ansible$ ansible-playbook --ask-become-pass install
_apache.yml
BECOME password:

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

TASK [Gathering Facts] *************************************************************
ok: [192.168.56.111]
fatal: [192.168.56.103]: UNREACHABLE! => {"changed": false, "msg": "Failed to connec
t to the host via ssh: ssh: connect to host 192.168.56.103 port 22: No route to host
", "unreachable": true}
fatal: [192.168.56.101]: UNREACHABLE! => {"changed": false, "msg": "Failed to connec
t to the host via ssh: ssh: connect to host 192.168.56.101 port 22: No route to host
", "unreachable": true}

TASK [install apache2 and php package for Ubuntu] **********************************
skipping: [192.168.56.111]

TASK [install apache2 and package for CentOs] **************************************
ok: [192.168.56.111]

PLAY RECAP *************************************************************************
192.168.56.101             : ok=0    changed=0    unreachable=1    failed=0    skipp
ed=0    rescued=0    ignored=0
192.168.56.103             : ok=0    changed=0    unreachable=1    failed=0    skipp
ed=0    rescued=0    ignored=0
192.168.56.111             : ok=2    changed=0    unreachable=0    failed=0    skipp
ed=1    rescued=0    ignored=0
```

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.

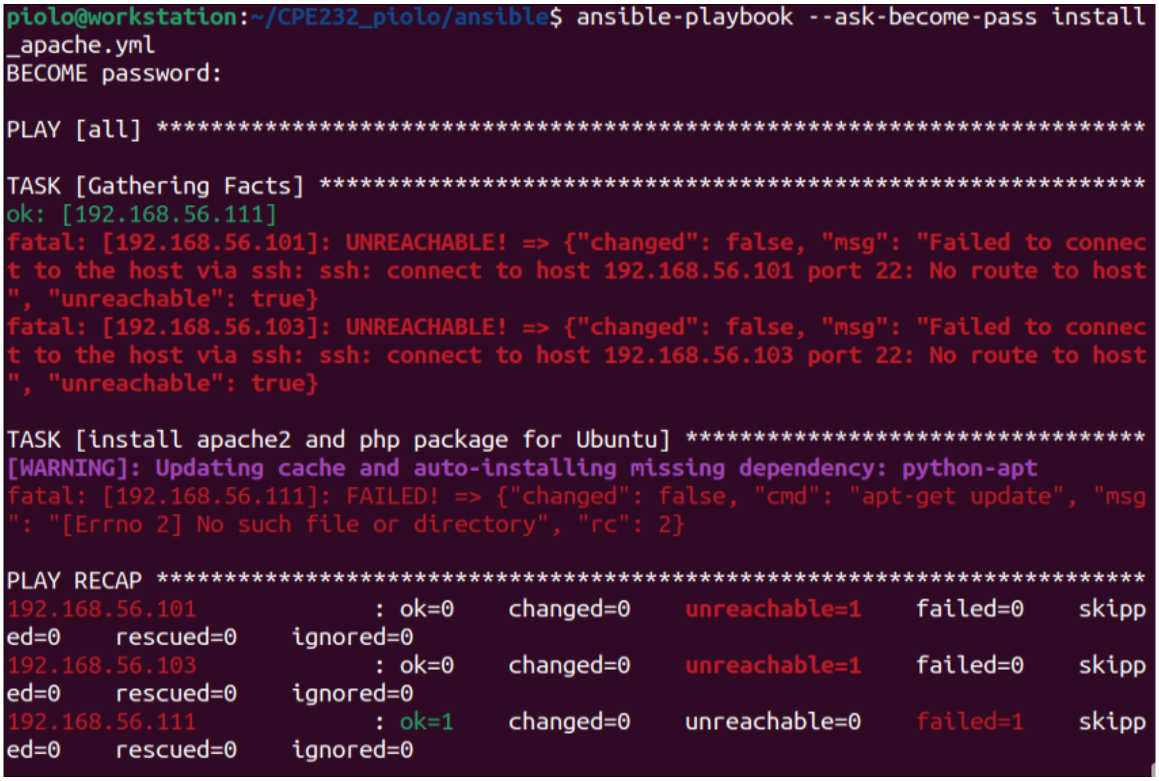


```
---
- hosts: all
  become: true
  tasks:

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



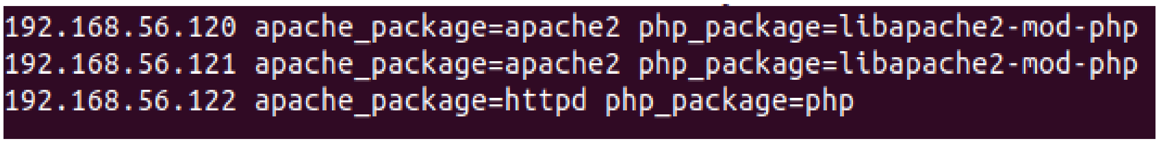
```
---
- hosts: all
  become: true
  tasks:
  - name: install apache2 and php package for Ubuntu
    apt:
      name:
        - "{{ apache_package }}"
        - "{{ php_package  }}"
      state: latest
      update_cache: yes
```

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



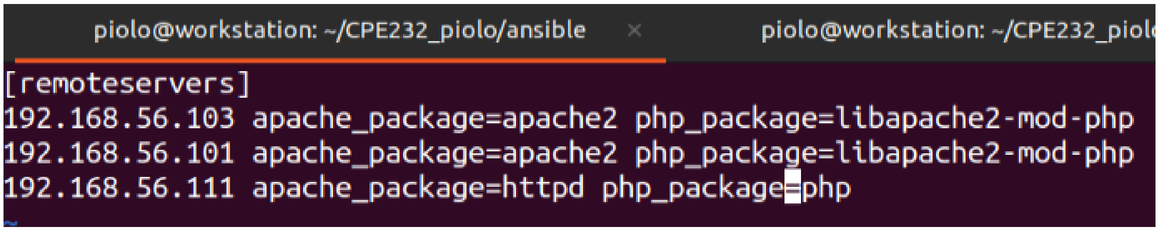
```
piolo@workstation:~/CPE232_piolo/ansible$ ansible-playbook --ask-become-pass install
_apache.yml
BECOME password:

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

TASK [Gathering Facts] ***********************************************************
ok: [192.168.56.111]
fatal: [192.168.56.101]: UNREACHABLE! => {"changed": false, "msg": "Failed to connec
t to the host via ssh: ssh: connect to host 192.168.56.101 port 22: No route to host
", "unreachable": true}
fatal: [192.168.56.103]: UNREACHABLE! => {"changed": false, "msg": "Failed to connec
t to the host via ssh: ssh: connect to host 192.168.56.103 port 22: No route to host
", "unreachable": true}

TASK [install apache2 and php package for Ubuntu] ********************************
[WARNING]: Updating cache and auto-installing missing dependency: python-apt
fatal: [192.168.56.111]: FAILED! => {"changed": false, "cmd": "apt-get update", "msg
": "[Errno 2] No such file or directory", "rc": 2}

PLAY RECAP ***********************************************************************
192.168.56.101             : ok=0    changed=0    unreachable=1    failed=0    skipp
ed=0    rescued=0    ignored=0
192.168.56.103             : ok=0    changed=0    unreachable=1    failed=0    skipp
ed=0    rescued=0    ignored=0
192.168.56.111             : ok=1    changed=0    unreachable=0    failed=1    skipp
ed=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:



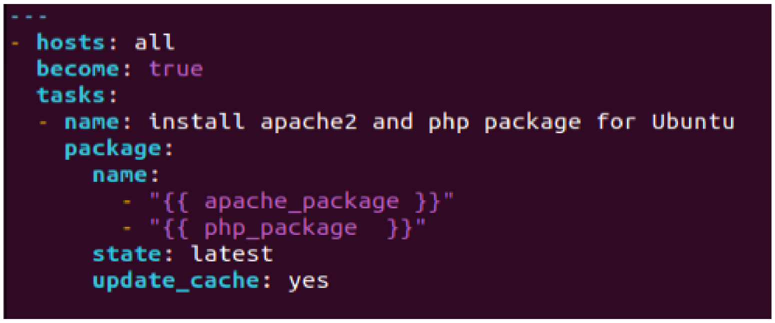
```
192.168.56.120 apache_package=apache2 php_package=libapache2-mod-php
192.168.56.121 apache_package=apache2 php_package=libapache2-mod-php
192.168.56.122 apache_package=httpd php_package=php
```

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


piolo@workstation: ~/CPE232_piolo/ansible
piolo@workstation: ~/CPE232_piol

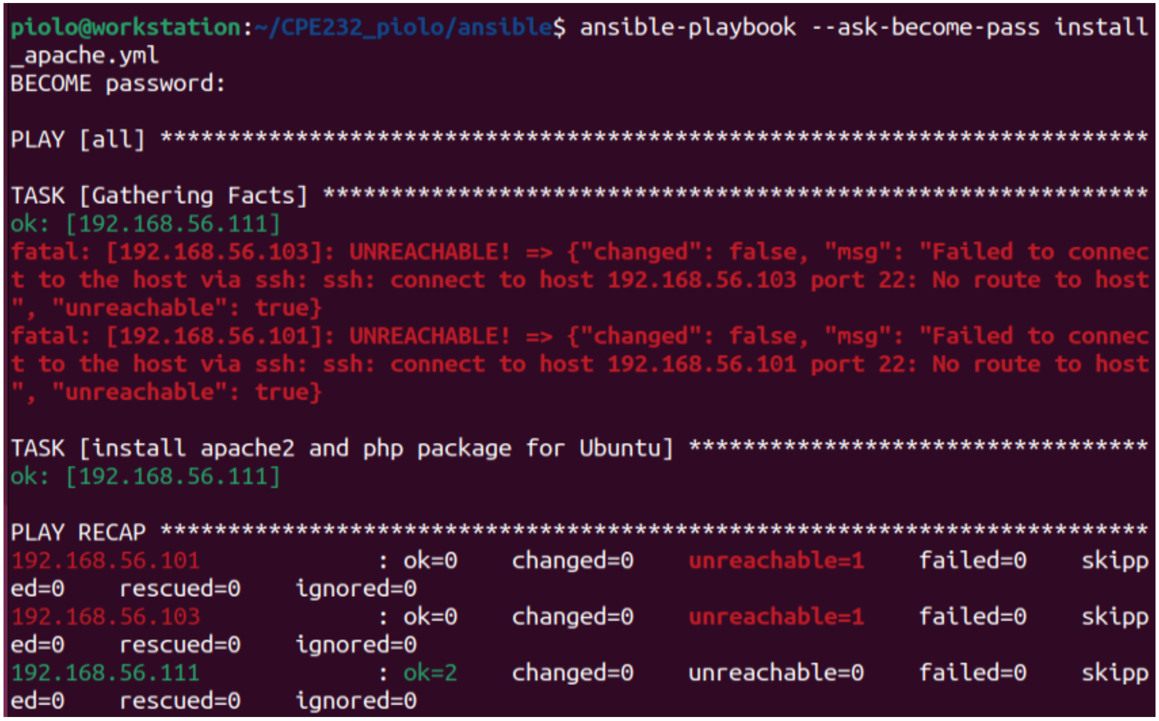
```
[remoteservers]
192.168.56.103 apache_package=apache2 php_package=libapache2-mod-php
192.168.56.101 apache_package=apache2 php_package=libapache2-mod-php
192.168.56.111 apache_package=httpd php_package=php
~
```

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

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



```
piolo@workstation:~/CPE232_piolo/ansible$ ansible-playbook --ask-become-pass install
_apache.yml
BECOME password:

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

TASK [Gathering Facts] *********************************************************
ok: [192.168.56.111]
fatal: [192.168.56.103]: UNREACHABLE! => {"changed": false, "msg": "Failed to connec
t to the host via ssh: ssh: connect to host 192.168.56.103 port 22: No route to host
", "unreachable": true}
fatal: [192.168.56.101]: UNREACHABLE! => {"changed": false, "msg": "Failed to connec
t to the host via ssh: ssh: connect to host 192.168.56.101 port 22: No route to host
", "unreachable": true}

TASK [install apache2 and php package for Ubuntu] ******************************
ok: [192.168.56.111]

PLAY RECAP *********************************************************************
192.168.56.101             : ok=0    changed=0    unreachable=1    failed=0    skipp
ed=0    rescued=0    ignored=0
192.168.56.103             : ok=0    changed=0    unreachable=1    failed=0    skipp
ed=0    rescued=0    ignored=0
192.168.56.111             : ok=2    changed=0    unreachable=0    failed=0    skipp
ed=0    rescued=0    ignored=0
```

Reflections:

Answer the following:

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

   - In doing refactoring, we are simplifying the massive yml configuration file into smaller and flexible code that can handle a bunch of commands that are suited for different operating systems without specifying it. As I observe in the task it performs the refactoring in a phase which shows on how we can simplify our yml ansible code into smaller version of it plus a feature of flexibility by defining a variable. Where the variable is stated in the yml file while the values are stated in the inventory file under ansible directory.

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

   - The "when" command is a basic conditional command for ansible yml configuration files. This is used when we are integrating a condition to a process like the "when: ansible_distributtion = osperating_system". If this command is inside of a block of process then it will first check whether the operating system matches the server before running the command. Therefore, this command is used in a wide variety of different yml configurations but mostly used in defining an operating system.

Honor Pledge

"I affirm that I will not give or receive unauthorized help on this activity and that all will be my own."