

<b>Name: Agpaoa, Ma.Diane J.</b>	<b>Date Performed: 09/13/2022</b>
<b>Course/Section: CPE 232-CPE31S22</b>	<b>Date Submitted: 09/15/2022</b>
<b>Instructor: Dr. Jonathan Taylar</b>	<b>Semester and SY: 1st Sem 2022-2023</b>
<b>Activity 5: Consolidating Playbook plays</b>	
<b>1. Objectives:</b> 1.1 Use <b>when</b> command in playbook for different OS distributions 1.2 Apply refactoring techniques in cleaning up the playbook codes	
<b>2. Discussion:</b>  <p>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.</p> <p>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.</p> <p><b>Requirement:</b>  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 <b>ssh-copy-id</b> to copy the public key to CentOS. Verify if you can successfully SSH to CentOS VM.</p>	
<b>Task 1: Use when command for different distributions</b>	
1. In the local machine, make sure you are in the local repository directory ( <b>CPE232_yourname</b> ). 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?	

```
madiane@workstation:~$ cd CPE232_Agpaoa-Ma.Diane
madiane@workstation:~/CPE232_Agpaoa-Ma.Diane$ ls
cpe_ansible  README.md
madiane@workstation:~/CPE232_Agpaoa-Ma.Diane$ git pull
Already up to date.
madiane@workstation:~/CPE232_Agpaoa-Ma.Diane$
```

**Figure 1.1.** Changing directory to repository directory and executing the command “*git pull*”

After issuing the command `git pull`, the message “Already up to date.” was displayed. This happened because there was no new content added from the local machine.

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
[remoteservers]
192.168.56.109
192.168.56.115
192.168.56.116
```

**Figure 1.2.** Adding the IP address of the Centos VM

```

nadiane@workstation:~/CPE232_Agpaoa-Ma.Diane/agpaoa_ansible$ ansible-playbook --ask-become-pass install_apache.yml
BECOME password:

PLAY [all] *****

TASK [Gathering Facts] *****
ok: [192.168.56.115]
ok: [192.168.56.109]
ok: [192.168.56.116]

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}
changed: [192.168.56.115]
changed: [192.168.56.116]

TASK [install apache2 package] *****
ok: [192.168.56.115]
ok: [192.168.56.116]

TASK [add PHP support for apache] *****
ok: [192.168.56.115]
ok: [192.168.56.116]

PLAY RECAP *****
192.168.56.109      : ok=1    changed=0    unreachable=0    failed=1    skipped=0    rescued=0
   ignored=0
192.168.56.115      : ok=4    changed=1    unreachable=0    failed=0    skipped=0    rescued=0
   ignored=0
192.168.56.116      : ok=4    changed=1    unreachable=0    failed=0    skipped=0    rescued=0
   ignored=0

```

**Figure 1.3.** Executing the playbook `install_apache` from the previous activity  
Based on the result, the server with CentOS failed in the task “update repository index”.

3. Edit the *install\_apache.yml* file and insert the lines shown below.

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

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

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

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

Make sure to save the file and exit.

Run *ansible-playbook --ask-become-pass install\_apache.yml* and describe the result.

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.

```
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 1.4.** Edited the install\_apache.yml based on the image shown in Task 1 Step 3

```
madiane@workstation:~/CPE232_Agpaoa-Ma.Diane/agpaoa_ansible$ ansible-playbook --ask-become-pass install_apache.yml
BECOME password:

PLAY [all] *****

TASK [Gathering Facts] *****
ok: [192.168.56.115]
ok: [192.168.56.116]
ok: [192.168.56.109]

TASK [update repository index] *****
skipping: [192.168.56.109]
changed: [192.168.56.115]
changed: [192.168.56.116]

TASK [install apache2 package] *****
skipping: [192.168.56.109]
ok: [192.168.56.115]
ok: [192.168.56.116]

TASK [add PHP support for apache] *****
skipping: [192.168.56.109]
ok: [192.168.56.115]
ok: [192.168.56.116]

PLAY RECAP *****
192.168.56.109      : ok=1    changed=0    unreachable=0    failed=0    skipped=3    rescued=0
                  ignored=0
192.168.56.115      : ok=4    changed=1    unreachable=0    failed=0    skipped=0    rescued=0
                  ignored=0
192.168.56.116      : ok=4    changed=1    unreachable=0    failed=0    skipped=0    rescued=0
                  ignored=0
```

**Figure 1.5.** Executing the new edited install\_apache.yml  
Based on the result, the remote server with CentOS was skipped on executing all of the tasks.

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
        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"
```

Make sure to save and exit.

Run *ansible-playbook --ask-become-pass install\_apache.yml* and describe the result.

```
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
    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 1.6.** Edited the install\_apache.yml based on the image shown in Task 1 Step 4.

```

nadiane@workstation:~/CPE232_Agpa0a-Ma.0lane/agpa0a_ansible$ ansible-playbook --ask-become-pass install_apache.yml
BECOME password:

PLAY [all] *****

TASK [Gathering Facts] *****
ok: [192.168.56.115]
ok: [192.168.56.116]
ok: [192.168.56.109]

TASK [update repository index] *****
skipping: [192.168.56.109]
changed: [192.168.56.115]
changed: [192.168.56.116]

TASK [install apache2 package] *****
skipping: [192.168.56.109]
ok: [192.168.56.115]
ok: [192.168.56.116]

TASK [add PHP support for apache] *****
skipping: [192.168.56.109]
ok: [192.168.56.115]
ok: [192.168.56.116]

TASK [update repository index] *****
skipping: [192.168.56.115]
skipping: [192.168.56.116]
ok: [192.168.56.109]

TASK [install apache2 package] *****
skipping: [192.168.56.115]
skipping: [192.168.56.116]
changed: [192.168.56.109]

TASK [add PHP support for apache] *****
skipping: [192.168.56.115]
skipping: [192.168.56.116]
changed: [192.168.56.109]

PLAY RECAP *****
192.168.56.109      : ok=4    changed=2    unreachable=0    failed=0    skipped=3    rescued=0    ignored=0
192.168.56.115      : ok=4    changed=1    unreachable=0    failed=0    skipped=3    rescued=0    ignored=0
192.168.56.116      : ok=4    changed=1    unreachable=0    failed=0    skipped=3    rescued=0    ignored=0

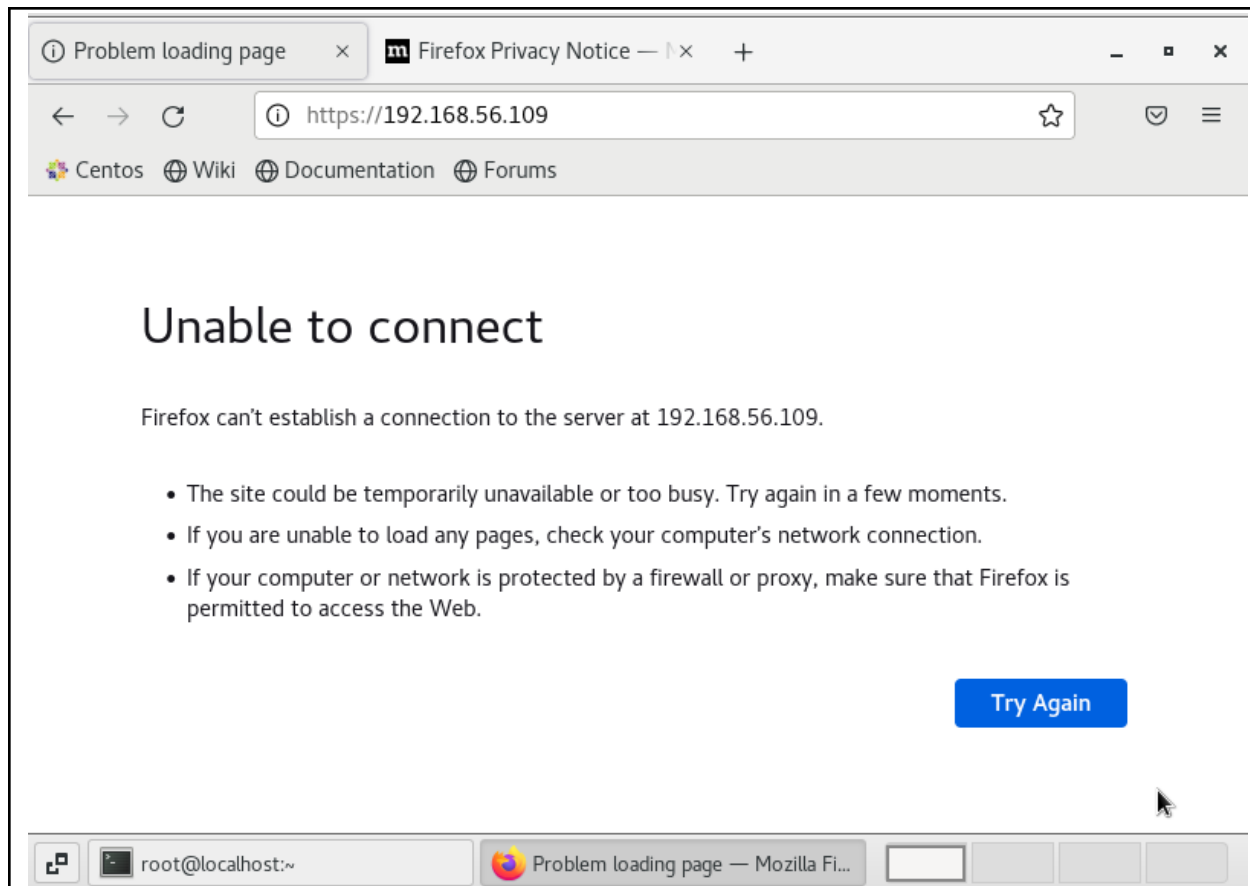
```

**Figure 1.7.** Executing the new edited install\_apache.yml

Based on the result, the remote server with CentOS was skipped in the 3 tasks which are the “update repository index”, “install apache2 package”, and add PHP support for apache. While in the last 3 tasks the CentOS was not skipped, the state was ok and changed in these last 3 tasks. If we observe the contents of install\_apache.yml, we could observe that the first 3 plays were designed for Ubuntu desktops, and the last 3 plays were designed for CentOS.

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.





**Figure 1.8.** Attempting to connect to the server at 192.168.56.109

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.

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

**Figure 1.9.** Executing the command “*systemctl status httpd*”

5.2 Issue the following command to start the service:

*sudo systemctl start httpd*

(When prompted, enter the sudo password)

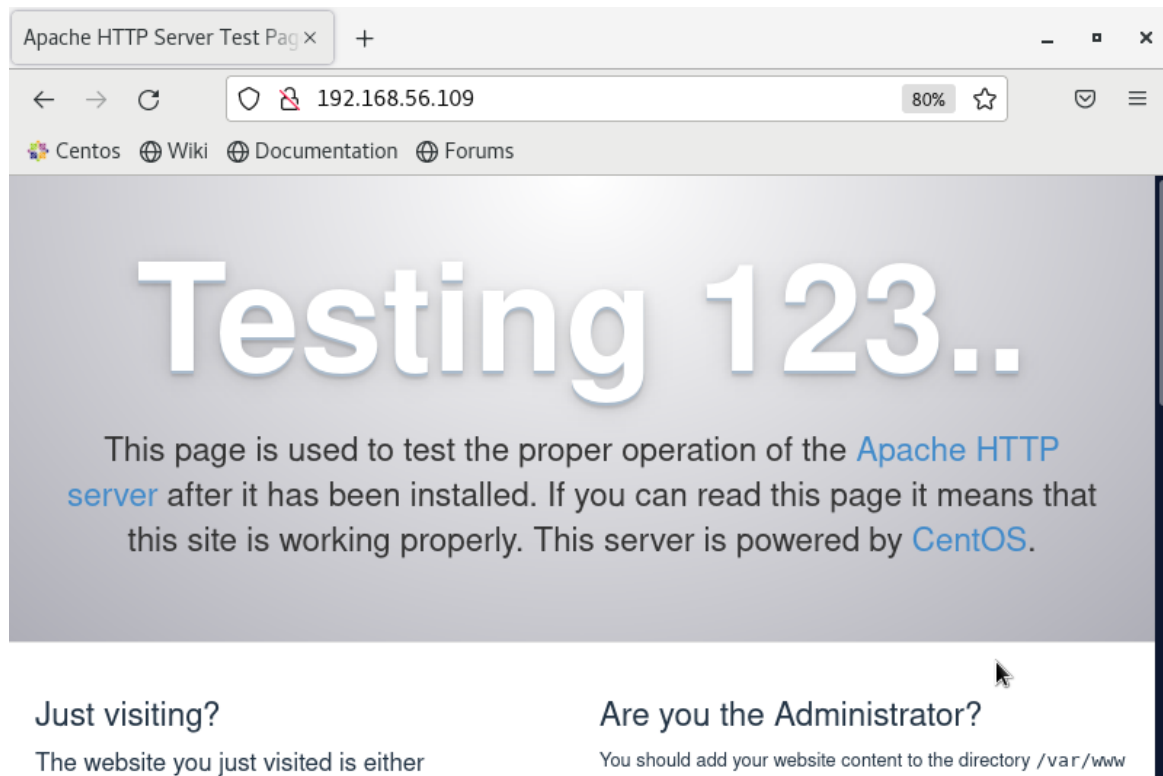
*sudo firewall-cmd --add-port=80/tcp*

(The result should be a success)

```
[root@localhost ~]# sudo systemctl start httpd
[root@localhost ~]# sudo firewall-cmd --add-port=80/tcp
success
```

**Figure 1.10.** Executing the command “*sudo systemctl start httpd*” and “*sudo firewall-cmd --add-port=80/tcp*”

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)



**Figure 1.11.** Successful verification of the service running

## Task 2: Refactoring playbook

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

1. Edit the playbook *install\_apache.yml*. Currently, we have three tasks targeting our Ubuntu machines and 3 tasks targeting our CentOS machine. Right now, we try to consolidate some tasks that are typically the same. For example, we can consolidate two plays that install packages. We can do that by creating a list of installation packages as shown below:

```

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

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

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

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

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

```

Make sure to save the file and exit.

Run *ansible-playbook --ask-become-pass install\_apache.yml* and describe the result.

```

GNU nano 6.2                                install_apache.yml
---
- 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"

```

**Figure 2.1.** Edited the install\_apache.yml based on the image shown in Task 2 step 1

```

madiane@workstation: ~/CPE232_Agpaoa-Ma.Diane/agpaoa_ansible$ ansible-playbook --ask-become-pass install_apache.yml
BECOME password:

PLAY [all] *****

TASK [Gathering Facts] *****
ok: [192.168.56.115]
ok: [192.168.56.116]
ok: [192.168.56.109]

TASK [update repository index Ubuntu] *****
skipping: [192.168.56.109]
changed: [192.168.56.115]
changed: [192.168.56.116]

TASK [install apache2 and php packages for Ubuntu] *****
skipping: [192.168.56.109]
ok: [192.168.56.115]
ok: [192.168.56.116]

TASK [update repository index for CentOS] *****
skipping: [192.168.56.115]
skipping: [192.168.56.116]
ok: [192.168.56.109]

TASK [install apache and php packages for CentOS] *****
skipping: [192.168.56.115]
skipping: [192.168.56.116]
ok: [192.168.56.109]

PLAY RECAP *****
192.168.56.109      : ok=3    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
192.168.56.115      : ok=3    changed=1    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
192.168.56.116      : ok=3    changed=1    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0

```

**Figure 2.2.** Executing the new edited install\_apache.yml

Based on the result, the remote server with CentOS was skipped in the tasks that are for Ubuntu and the server with Ubuntu was skipped in the tasks that are for CentOS.

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

    - name: install apache2 and php packages for Ubuntu
      apt:
        name:
          - apache2
          - libapache2-mod-php
        state: latest
        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 based on the image shown in Task 2 Step 2

```

madiane@workstation:~/CPE232_Agpaoa-Ma.Diane/agpaoa_ansible$ ansible-playbook --ask-become-pass install_apache.yml
BECOME password:

PLAY [all] *****

TASK [Gathering Facts] *****
ok: [192.168.56.115]
ok: [192.168.56.109]
ok: [192.168.56.116]

TASK [install apache2 and php packages for Ubuntu] *****
skipping: [192.168.56.109]
ok: [192.168.56.116]
ok: [192.168.56.115]

TASK [install apache and php packages for CentOS] *****
skipping: [192.168.56.115]
skipping: [192.168.56.116]
ok: [192.168.56.109]

PLAY RECAP *****
192.168.56.109      : ok=2    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
192.168.56.115      : ok=2    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
192.168.56.116      : ok=2    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0

```

**Figure 2.4.** Executing the new edited install\_apache.yml

Based on the result, the remote server with CentOS was skipped in the tasks that are for Ubuntu and the server with Ubuntu was skipped in the tasks that are for CentOS. In addition, the tasks executed on this playbook were only 2; it was fewer than the previous playbook.

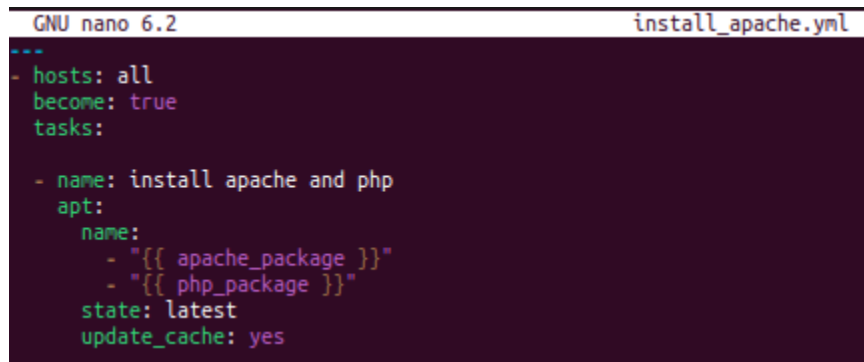
- Finally, we can consolidate these 2 plays in just 1 play. This can be done by declaring variables that will represent the packages that we want to install. Basically, the apache\_package and php\_package are variables. The names

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 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 *install\_apache.yml* based on the image shown in Task 2 Step 3.

```

madiane@workstation:~/CPE232_Agpaoa-Ma.Diane/agpaoa_ansible$ ansible-playbook --ask-become-pass install_apache.yml
BECOME password:

PLAY [all] *****

TASK [Gathering Facts] *****
ok: [192.168.56.116]
ok: [192.168.56.115]
ok: [192.168.56.109]

TASK [install apache and php] *****
fatal: [192.168.56.109]: FAILED! => {"msg": "The task includes an option with an undefined variable. The error was: 'a
pache_package' is undefined\n\nThe error appears to be in '/home/madiane/CPE232_Agpaoa-Ma.Diane/agpaoa_ansible/install
_apache.yml': line 6, column 5, but may\nbe elsewhere in the file depending on the exact syntax problem.\n\nThe offend
ing line appears to be:\n\n\n - name: install apache and php\n    ^ here\n"}
fatal: [192.168.56.115]: FAILED! => {"msg": "The task includes an option with an undefined variable. The error was: 'a
pache_package' is undefined\n\nThe error appears to be in '/home/madiane/CPE232_Agpaoa-Ma.Diane/agpaoa_ansible/install
_apache.yml': line 6, column 5, but may\nbe elsewhere in the file depending on the exact syntax problem.\n\nThe offend
ing line appears to be:\n\n\n - name: install apache and php\n    ^ here\n"}
fatal: [192.168.56.116]: FAILED! => {"msg": "The task includes an option with an undefined variable. The error was: 'a
pache_package' is undefined\n\nThe error appears to be in '/home/madiane/CPE232_Agpaoa-Ma.Diane/agpaoa_ansible/install
_apache.yml': line 6, column 5, but may\nbe elsewhere in the file depending on the exact syntax problem.\n\nThe offend
ing line appears to be:\n\n\n - name: install apache and php\n    ^ here\n"}

PLAY RECAP *****
192.168.56.109      : ok=1    changed=0    unreachable=0    failed=1    skipped=0    rescued=0    ignored=0
192.168.56.115      : ok=1    changed=0    unreachable=0    failed=1    skipped=0    rescued=0    ignored=0
192.168.56.116      : ok=1    changed=0    unreachable=0    failed=1    skipped=0    rescued=0    ignored=0

```

**Figure 2.6.** Executing the new edited install\_apache.yml

Based on the result, our attempt to consolidate 2 plays in just 1 play failed.

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
[remoteservers]
192.168.56.109 apache_package=httpd php_package=php
192.168.56.116 apache_package=apache2 php_package=libapache2-mod-php
192.168.56.115 apache_package=apache2 php_package=libapache2-mod-php

```

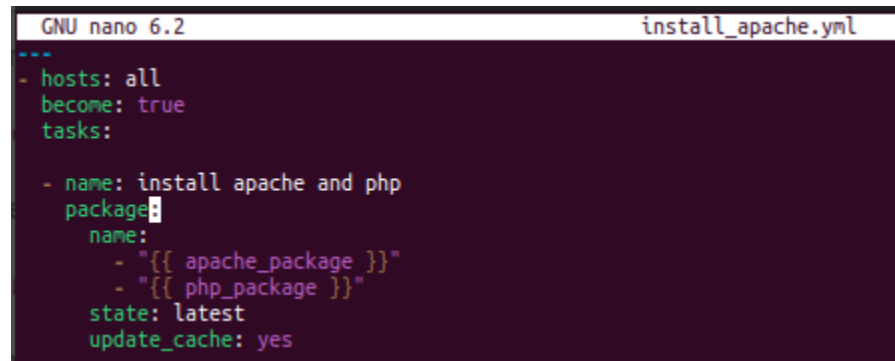
**Figure 2.7.** Edited the inventory based on the instruction

**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:



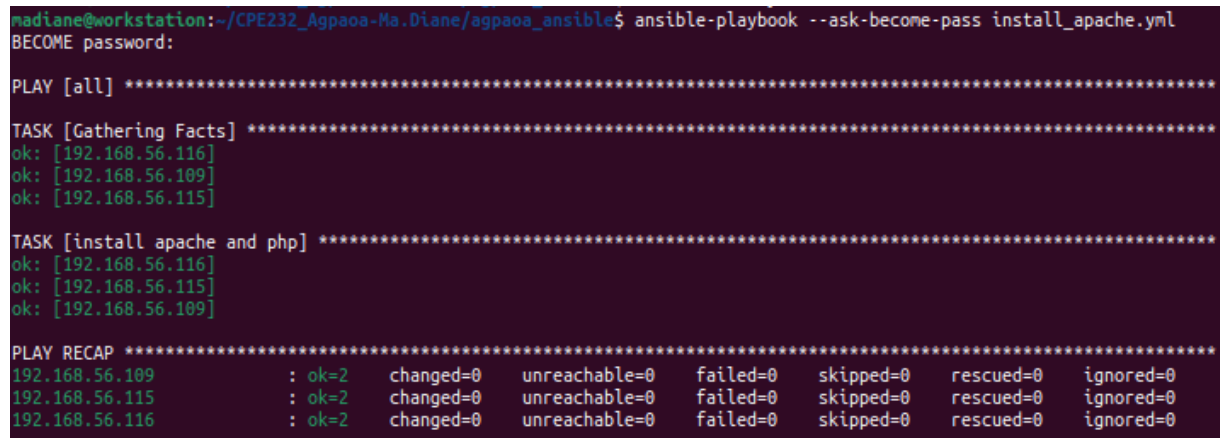
[ansible.builtin.package – Generic OS package manager — Ansible Documentation](#)

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



```
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. Replacing “apt” with “package”



```
madiane@workstation:~/CPE232_Agpaoa-Ma.Diane/agpaoa_ansible$ ansible-playbook --ask-become-pass install_apache.yml
BECOME password:

PLAY [all] *****

TASK [Gathering Facts] *****
ok: [192.168.56.116]
ok: [192.168.56.109]
ok: [192.168.56.115]

TASK [install apache and php] *****
ok: [192.168.56.116]
ok: [192.168.56.115]
ok: [192.168.56.109]

PLAY RECAP *****
192.168.56.109      : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
192.168.56.115      : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
192.168.56.116      : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

Figure 2.9. Executing the new edited install\_apache.yml

Based on the result, consolidating 2 plays in just 1 play succeeded. By declaring the variables in the inventory and changing the designation of the package, specifically we changed it from “apt” to “package”, this will allow it to run in CentOS.

## Reflections:

Answer the following:

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

Refactoring of playbook code is important to speed up the execution of plays in the playbook code. Using a playbook code with many plays will only result in higher possibilities of bugs and a much difficult process in debugging.

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

The when command is used as a conditional statement in playbooks it is usually used when a playbook is executed to remote servers with different operating systems that use different package managers and/or different apache packages. For example, in this activity we use the command “*when: ansible\_distribution == operating system of the remote server*”. When we run this playbook, the remote servers that have an operating system that does not match with the condition of a certain task/play will be skipped and the remote servers that match the condition of a certain task/play will have a state of “*changed*” or “*ok*”.

**Honor Pledge:**

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