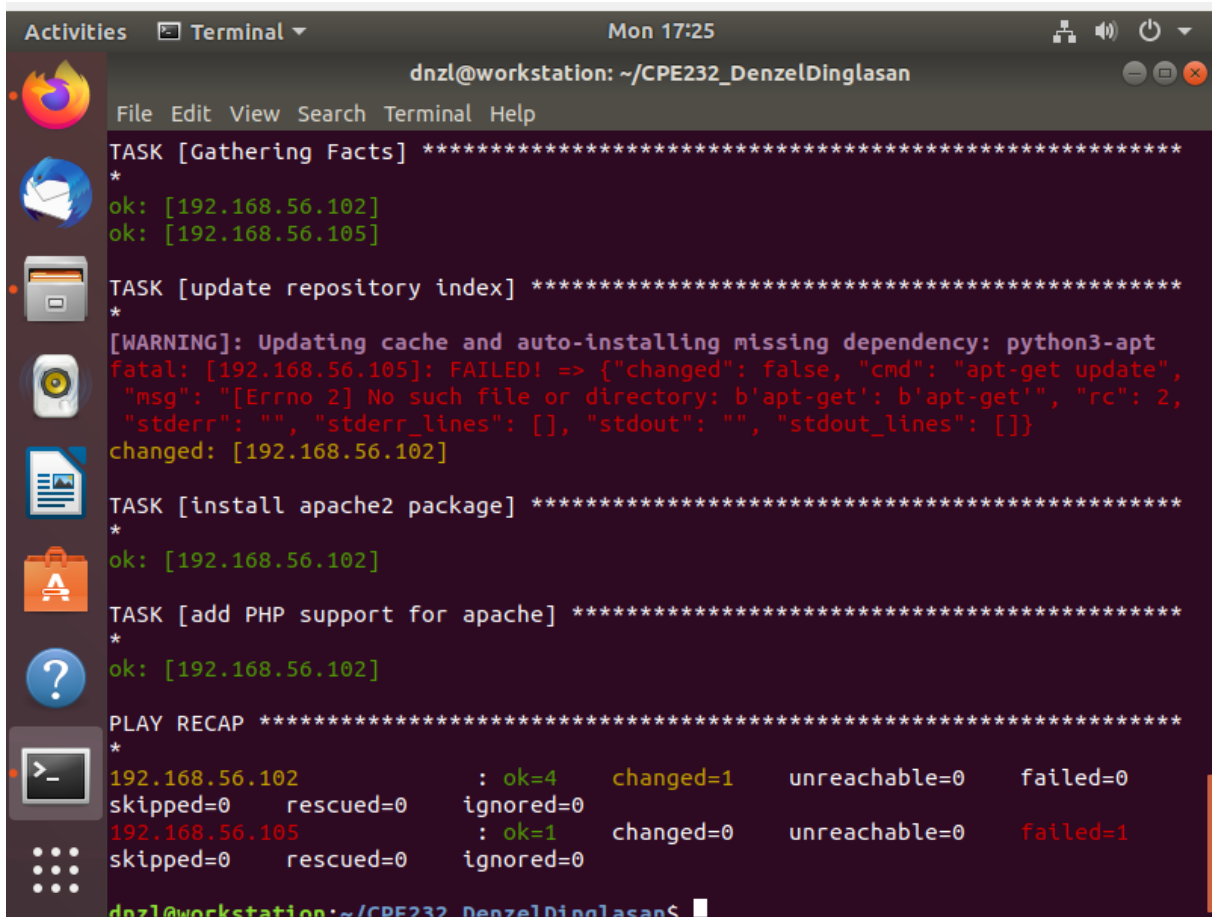


Name: Denzel A. Dinglasan	Date Performed: 18/09/2023
Course/Section: CPE232-CPE31S6	Date Submitted: 18/09/2023
Instructor: Dr. Jonathan Vidal Taylar	Semester and SY: 1st Sem 2023-2024
Activity 5: Consolidating Playbook plays	
1. Objectives: 1.1 Use when command in playbook for different OS distributions 1.2 Apply refactoring techniques in cleaning up the playbook codes	
2. Discussion: <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>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.</p>	
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?	
<pre>dnzl@workstation:~/CPE232_DenzelDinglasan\$ git pull Already up to date.</pre>	

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."



```
Mon 17:25
dnzl@workstation: ~/CPE232_DenzelDinglasan

TASK [Gathering Facts] *****
*
ok: [192.168.56.102]
ok: [192.168.56.105]

TASK [update repository index] *****
*
[WARNING]: Updating cache and auto-installing missing dependency: python3-apt
fatal: [192.168.56.105]: FAILED! => {"changed": false, "cmd": "apt-get update",
"msg": "[Errno 2] No such file or directory: b'apt-get': b'apt-get'", "rc": 2,
"stderr": "", "stderr_lines": [], "stdout": "", "stdout_lines": []}
changed: [192.168.56.102]

TASK [install apache2 package] *****
*
ok: [192.168.56.102]

TASK [add PHP support for apache] *****
*
ok: [192.168.56.102]

PLAY RECAP *****
*
192.168.56.102      : ok=4    changed=1    unreachable=0    failed=0
skipped=0    rescued=0    ignored=0
192.168.56.105    : ok=1    changed=0    unreachable=0    failed=1
skipped=0    rescued=0    ignored=0

dnzl@workstation:~/CPE232_DenzelDinglasan$
```

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.

```

TASK [Gathering Facts] *****
*
ok: [192.168.56.102]
ok: [192.168.56.105]

TASK [update repository index] *****
*
skipping: [192.168.56.105]
changed: [192.168.56.102]

TASK [install apache2 package] *****
*
skipping: [192.168.56.105]
ok: [192.168.56.102]

TASK [add PHP support for apache] *****
*
skipping: [192.168.56.105]
ok: [192.168.56.102]

PLAY RECAP *****
*
192.168.56.102      : ok=4    changed=1    unreachable=0    failed=0
skipped=0    rescued=0    ignored=0
192.168.56.105      : ok=1    changed=0    unreachable=0    failed=0
skipped=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"]

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
        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.

Activities Terminal Mon 17:42 dnl@workstation: ~/CPE232_DenzelDinglasan

File Edit View Search Terminal Help

GNU nano 2.9.3 install apache.yml

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

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

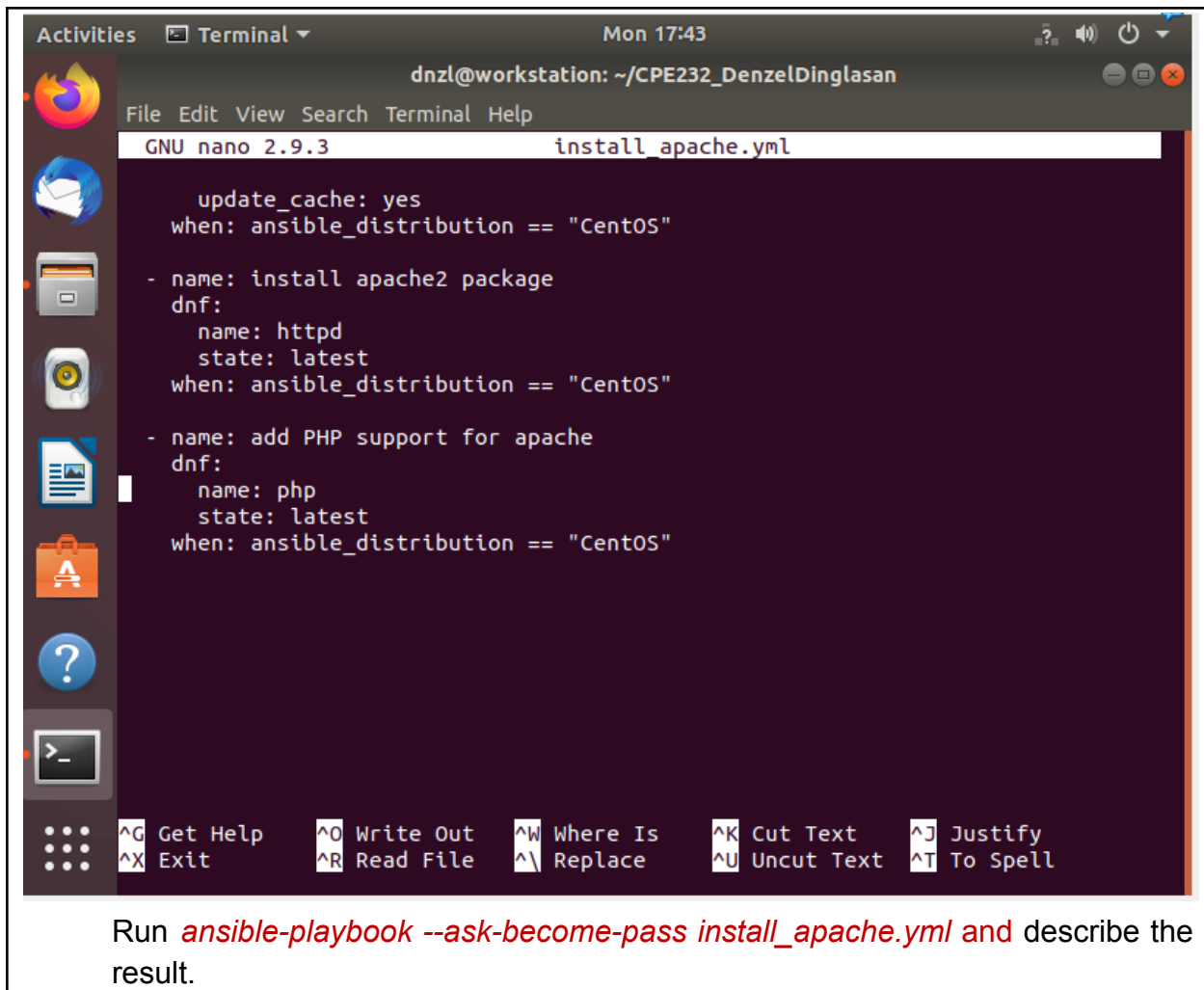
    - name: install apache2 package
      apt:
        name: apache2
        state: latest
        when: ansible_distribution == "Ubuntu"

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

    - name: update repository index
```

[Read 39 lines]

^G Get Help	^O Write Out	^W Where Is	^K Cut Text	^J Justify
^X Exit	^R Read File	^_ Replace	^U Uncut Text	^T To Spell



The screenshot shows a Linux desktop environment with a terminal window open. The terminal title bar indicates the user is 'dnzl@workstation' and the current directory is '~/CPE232_DenzelDinglasan'. The terminal is running the GNU nano 2.9.3 text editor, editing a file named 'install_apache.yml'. The playbook content is as follows:

```
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"
```

At the bottom of the terminal window, there is a list of keyboard shortcuts for the nano editor:

^G	Get Help	^O	Write Out	^W	Where Is	^K	Cut Text	^J	Justify
^X	Exit	^R	Read File	^_	Replace	^U	Uncut Text	^T	To Spell

Below the terminal window, there is a text instruction:

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

```
dnzl@workstation:~/CPE232_DenzelDinglasan$ ansible-playbook --ask-become-pass i
nsta11_apache.yml
BECOME password:
```

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

```
TASK [Gathering Facts] *****
*
```

```
ok: [192.168.56.102]
```

```
ok: [192.168.56.105]
```

```
TASK [update repository index] *****
*
```

```
skipping: [192.168.56.105]
```

```
changed: [192.168.56.102]
```

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

```
skipping: [192.168.56.105]
```

```
ok: [192.168.56.102]
```

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

```
skipping: [192.168.56.105]
```

```
ok: [192.168.56.102]
```

```
TASK [update repository index] *****
*
```

```
skipping: [192.168.56.102]
```

```
ok: [192.168.56.105]
```

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

```
skipping: [192.168.56.102]
```

```
changed: [192.168.56.105]
```

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

```
skipping: [192.168.56.102]
```

```
changed: [192.168.56.105]
```

```
PLAY RECAP *****
*
```

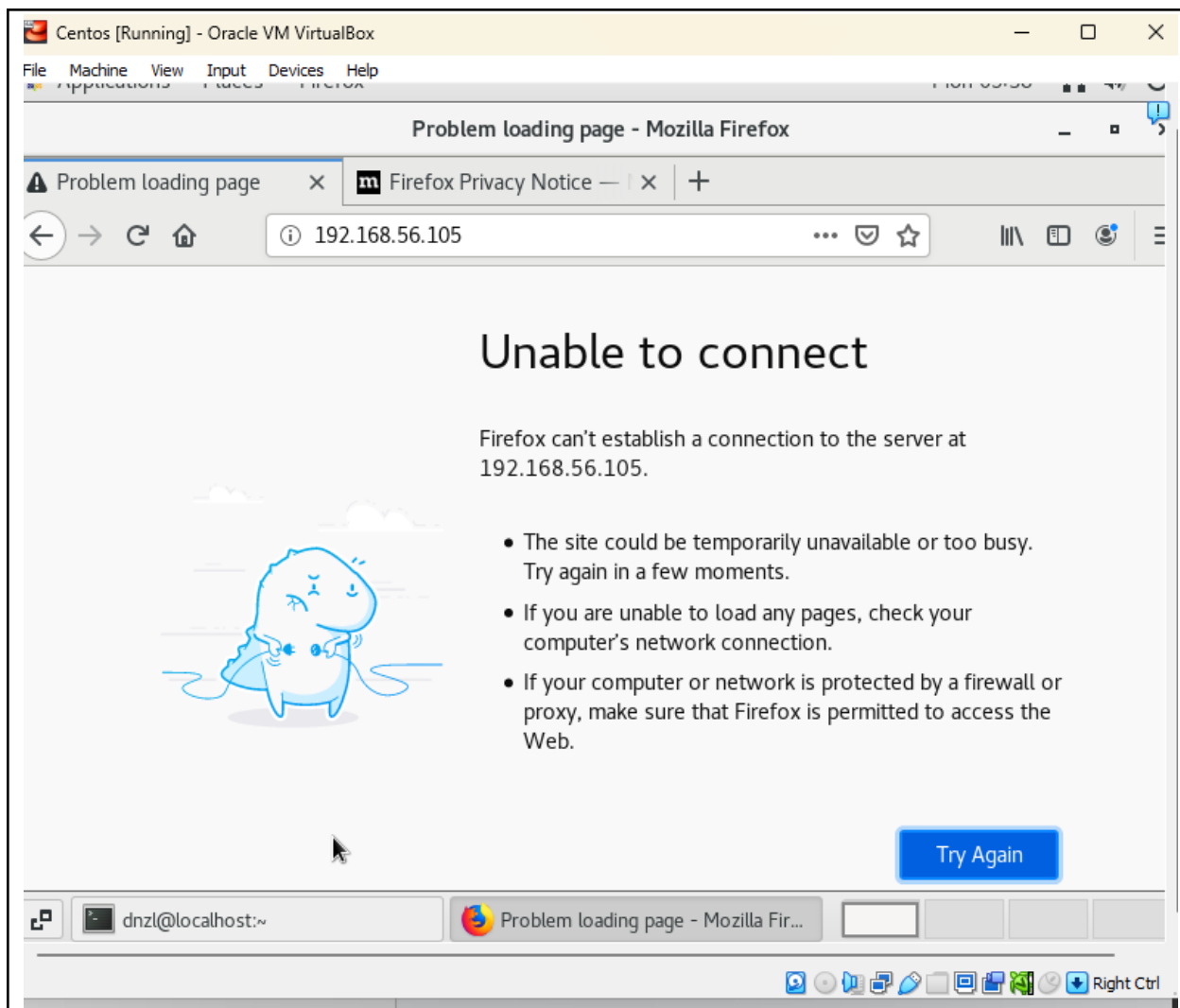
```
192.168.56.102      : ok=4    changed=1    unreachable=0    failed=0
```

```
skipped=3    rescued=0    ignored=0
```

```
192.168.56.105      : ok=4    changed=2    unreachable=0    failed=0
```

```
skipped=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.



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.

```
LINK/ether 52:54:00:44:ca:23 brd ff:ff:ff:ff:ff:ff
[dnzl@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)
[dnzl@localhost ~]$
```

5.2 Issue the following command to start the service:

sudo systemctl start httpd

(When prompted, enter the sudo password)

```
[dnzl@localhost ~]$ sudo systemctl start httpd
[sudo] password for dnzl:
[dnzl@localhost ~]$ systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; vendor preset: disabled)
   Active: active (running) since Mon 2023-09-18 05:40:01 EDT; 25s ago
     Docs: man:httpd(8)
           man:apachectl(8)
  Main PID: 9605 (httpd)
    Status: "Total requests: 0; Current requests/sec: 0; Current traffic: 0 B/sec"
     Tasks: 6
    CGroup: /system.slice/httpd.service
            └─9605 /usr/sbin/httpd -DFOREGROUND
              └─9608 /usr/sbin/httpd -DFOREGROUND
                └─9609 /usr/sbin/httpd -DFOREGROUND
                  └─9610 /usr/sbin/httpd -DFOREGROUND
                    └─9611 /usr/sbin/httpd -DFOREGROUND
                      └─9612 /usr/sbin/httpd -DFOREGROUND

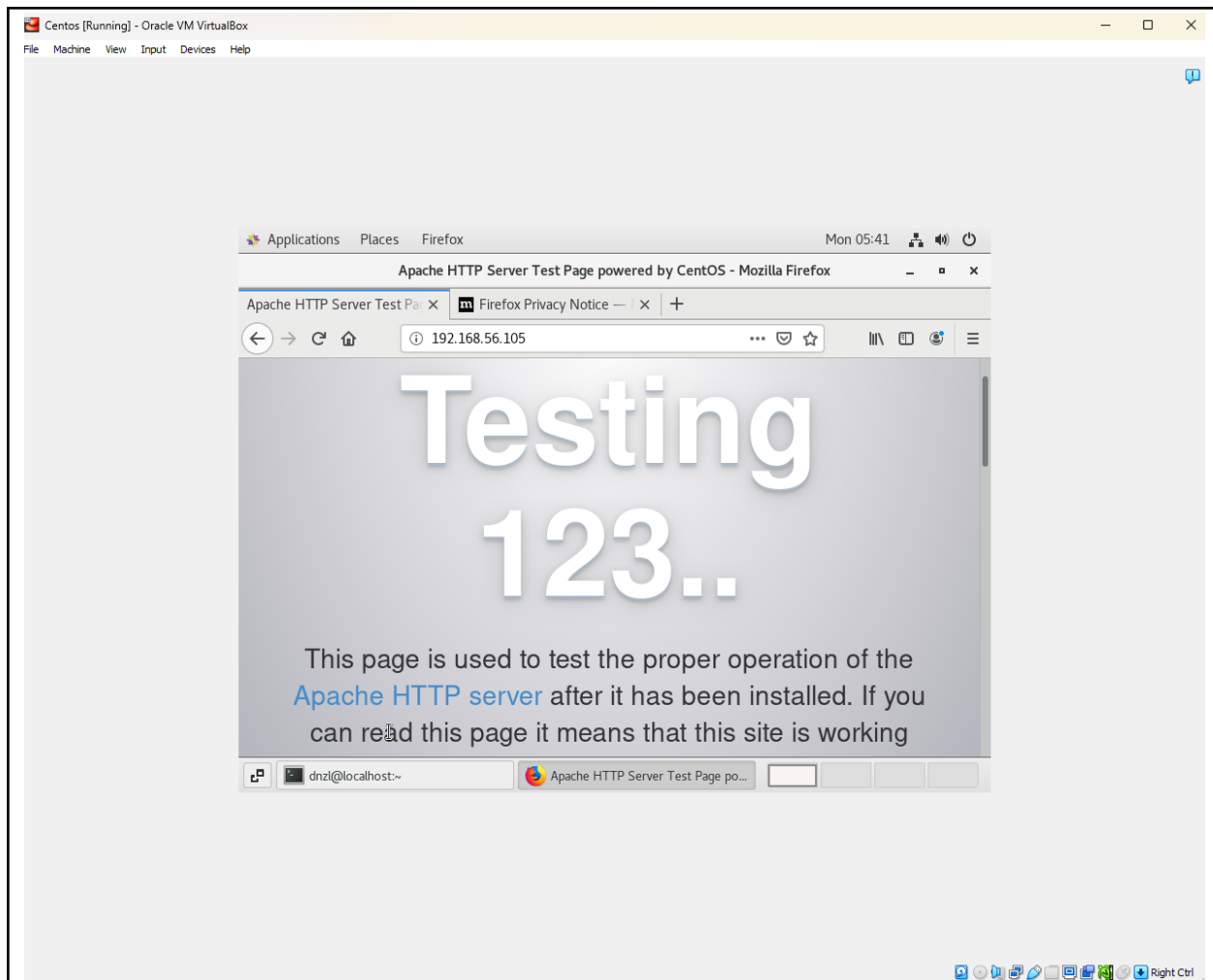
Sep 18 05:40:01 localhost.localdomain systemd[1]: Starting The Apache HTTP Server...
Sep 18 05:40:01 localhost.localdomain httpd[9605]: AH00558: httpd: Could not reliab...e
Sep 18 05:40:01 localhost.localdomain systemd[1]: Started The Apache HTTP Server.
Hint: Some lines were ellipsized, use -l to show in full.
```

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

(The result should be a success)

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

5.3 To verify the service is already running, go to CentOS VM and type its IP address on the browser. Was it successful? (Screenshot the browser)



Task 2: Refactoring playbook

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

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

```

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

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

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

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

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

```

Make sure to save the file and exit.

Activities Terminal Mon 17:49

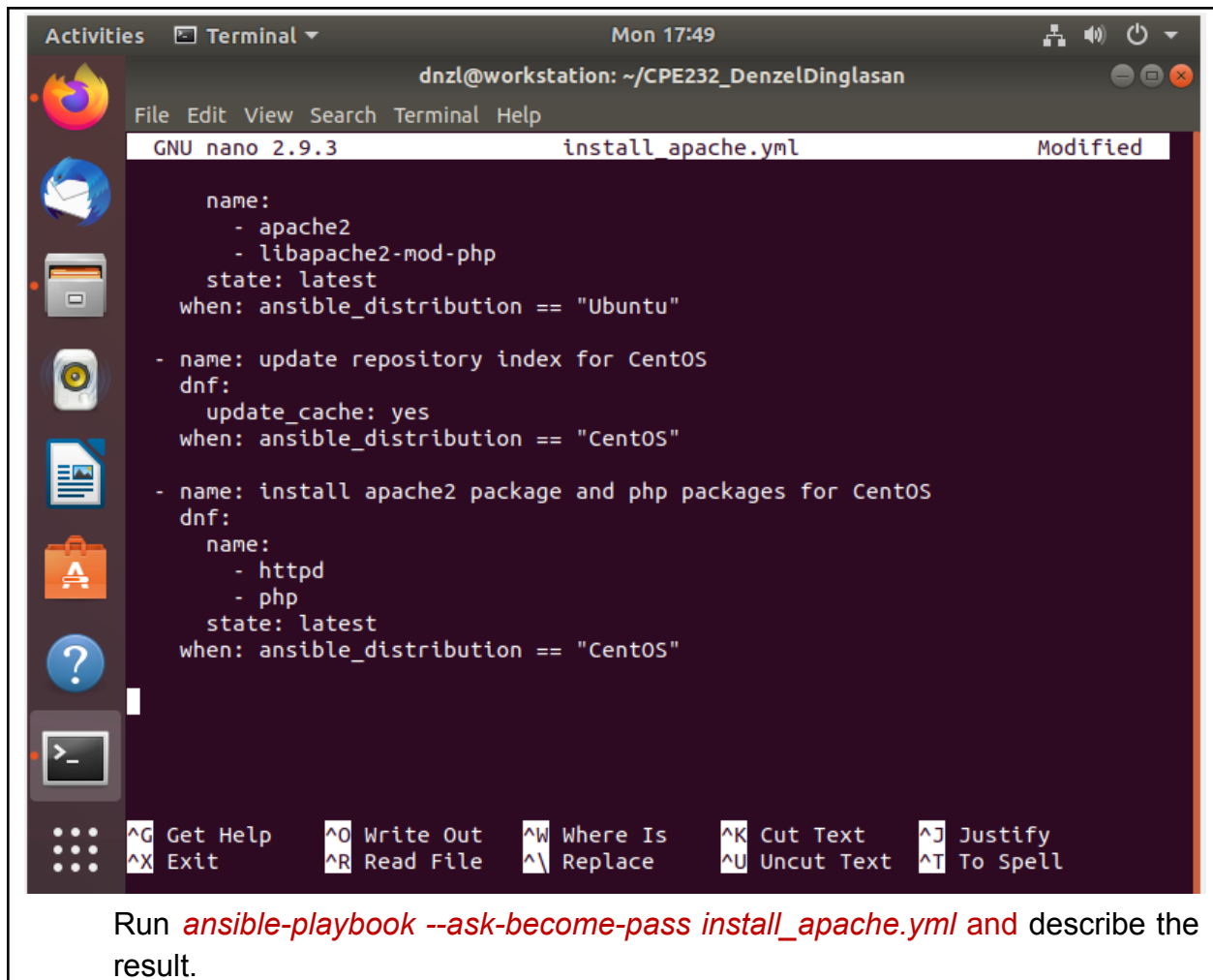
dnzl@workstation: ~/CPE232_DenzelDinglasan

File Edit View Search Terminal Help

GNU nano 2.9.3 install_apache.yml Modified

```
---
- hosts: all
  become: true
  tasks:
    - name: update repository index Ubuntu
      apt:
        update_cache: yes
        when: ansible_distribution == "Ubuntu"
    - name: install apache2 package 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"
```

Get Help Write Out Where Is Cut Text Justify
Exit Read File Replace Uncut Text To Spell



The screenshot shows a terminal window titled "Terminal" with the user "dnzl@workstation" and the path "~/CPE232_DenzelDinglasan". The window displays the GNU nano 2.9.3 editor editing the file "install_apache.yml". The file contains an Ansible playbook with three tasks: installing apache2 and libapache2-mod-php on Ubuntu, updating the repository index for CentOS, and installing apache2, httpd, and php on CentOS. The terminal window has a sidebar with icons for various applications and a bottom status bar with keyboard shortcuts.

```
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 apache2 package and php packages for CentOS
  dnf:
    name:
      - httpd
      - php
    state: latest
  when: ansible_distribution == "CentOS"
```

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

```

dnzl@workstation:~/CPE232_DenzelDinglasan$ ansible-playbook --ask-become-pass i
nsta11_apache.yml
BECOME password:

PLAY [all] *****
*

TASK [Gathering Facts] *****
*
ok: [192.168.56.102]
ok: [192.168.56.105]

TASK [update repository index Ubuntu] *****
*
skipping: [192.168.56.105]
changed: [192.168.56.102]

TASK [install apache2 package and php packages for Ubuntu] *****
*
skipping: [192.168.56.105]
ok: [192.168.56.102]

TASK [update repository index for CentOS] *****
*
skipping: [192.168.56.102]
ok: [192.168.56.105]

TASK [install apache2 package and php packages for CentOS] *****
*
skipping: [192.168.56.102]
ok: [192.168.56.105]

TASK [install apache2 package and php packages for CentOS] *****
*
skipping: [192.168.56.102]
ok: [192.168.56.105]

PLAY RECAP *****
*
192.168.56.102      : ok=3    changed=1    unreachable=0    failed=0
skipped=2    rescued=0    ignored=0
192.168.56.105      : ok=3    changed=0    unreachable=0    failed=0
skipped=2    rescued=0    ignored=0

```

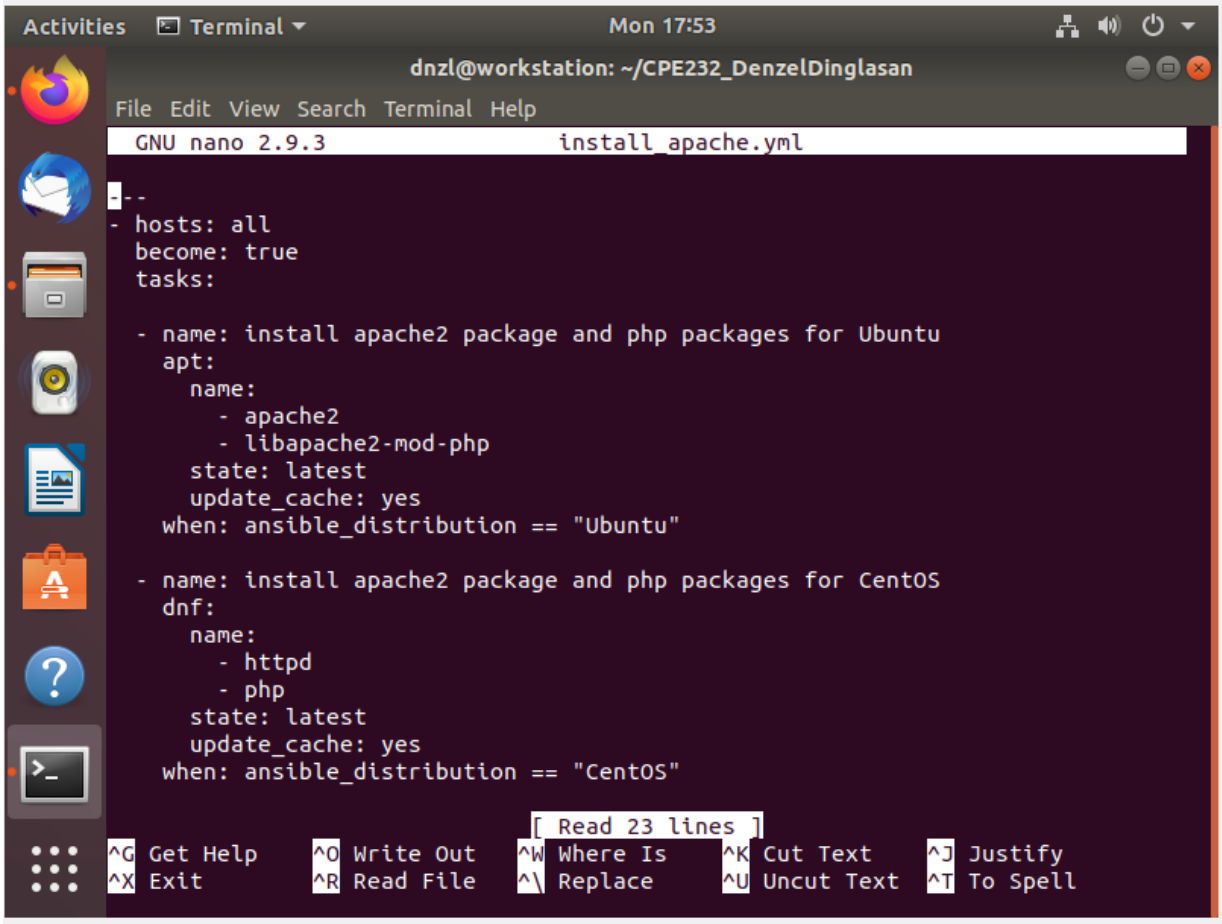
2. Edit the playbook *install_apache.yml* again. In task 2.1, we consolidated the plays into one play. This time we can actually consolidated everything in just 2 plays. This can be done by removing the update repository play and putting the command *update_cache: yes* below the command *state: latest*.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

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

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

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

Make sure to save the file and exit.



Activities Terminal Mon 17:53

dnzl@workstation: ~/CPE232_DenzelDinglasan

File Edit View Search Terminal Help

GNU nano 2.9.3 install_apache.yml

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

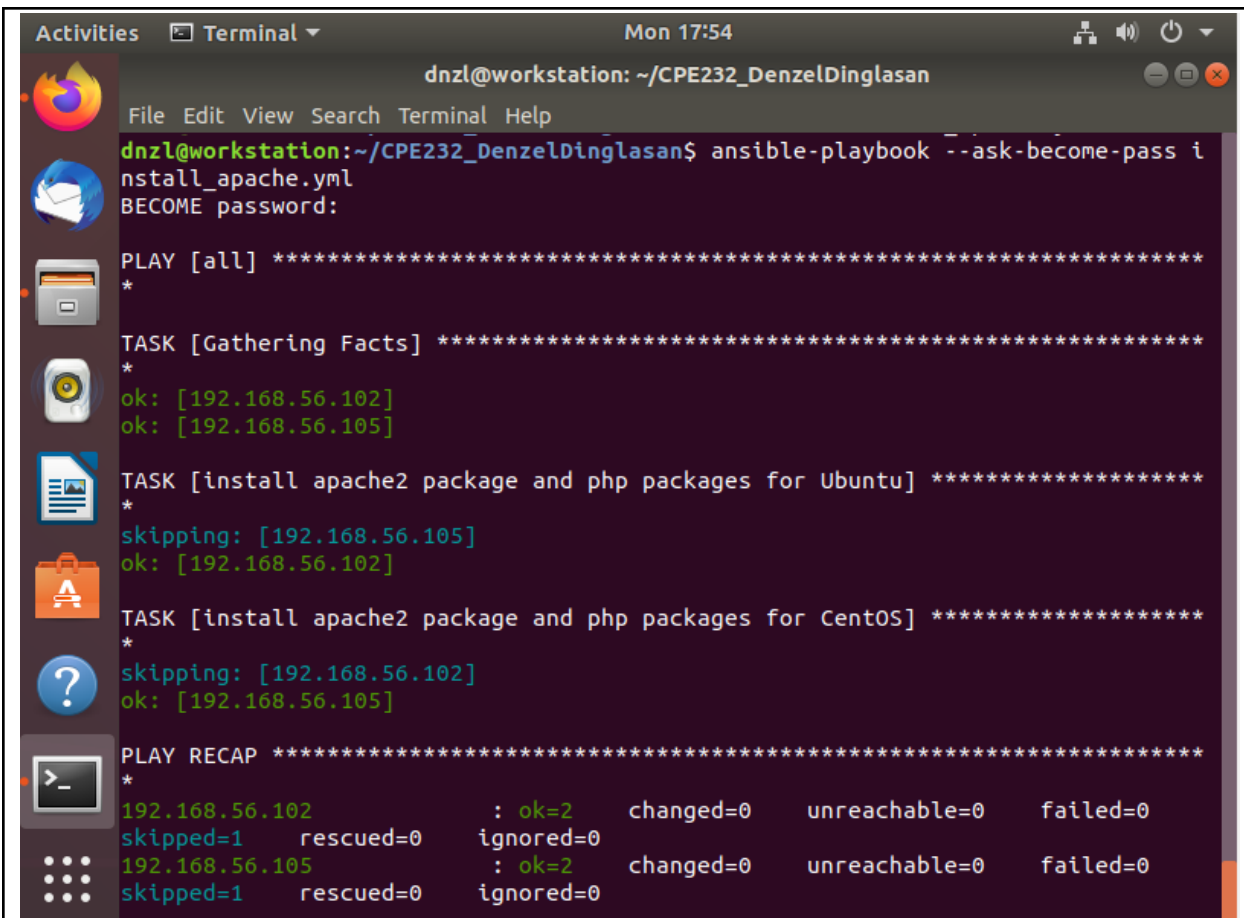
    - name: install apache2 package and php packages for Ubuntu
      apt:
        name:
          - apache2
          - libapache2-mod-php
        state: latest
        update_cache: yes
        when: ansible_distribution == "Ubuntu"

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

[Read 23 lines]

Get Help	Write Out	Where Is	Cut Text	Justify
Exit	Read File	Replace	Uncut Text	To Spell

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



The terminal window shows the execution of an Ansible playbook named `install_apache.yml`. The user is prompted for a password to become the 'pass' user. The playbook runs on two hosts: 192.168.56.102 and 192.168.56.105. The tasks performed are: gathering facts, installing the Apache2 package and PHP packages for Ubuntu on 192.168.56.102, and installing the Apache2 package and PHP packages for CentOS on 192.168.56.105. The final output shows a recap of the results for both hosts.

```
dnzl@workstation: ~/CPE232_DenzelDinglasan
File Edit View Search Terminal Help
dnzl@workstation:~/CPE232_DenzelDinglasan$ ansible-playbook --ask-become-pass i
ninstall_apache.yml
BECOME password:

PLAY [all] *****
*

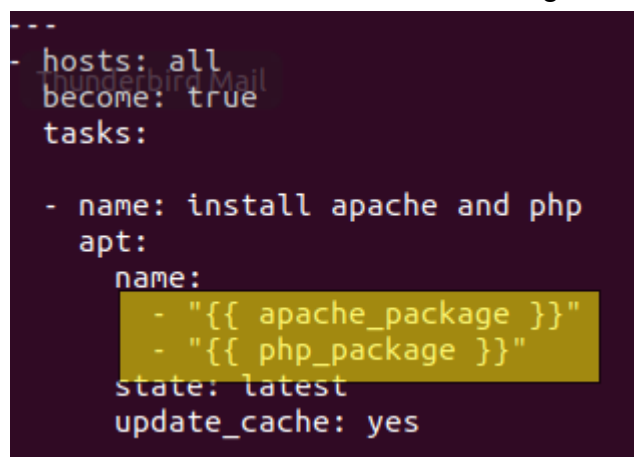
TASK [Gathering Facts] *****
*
ok: [192.168.56.102]
ok: [192.168.56.105]

TASK [install apache2 package and php packages for Ubuntu] *****
*
skipping: [192.168.56.105]
ok: [192.168.56.102]

TASK [install apache2 package and php packages for CentOS] *****
*
skipping: [192.168.56.102]
ok: [192.168.56.105]

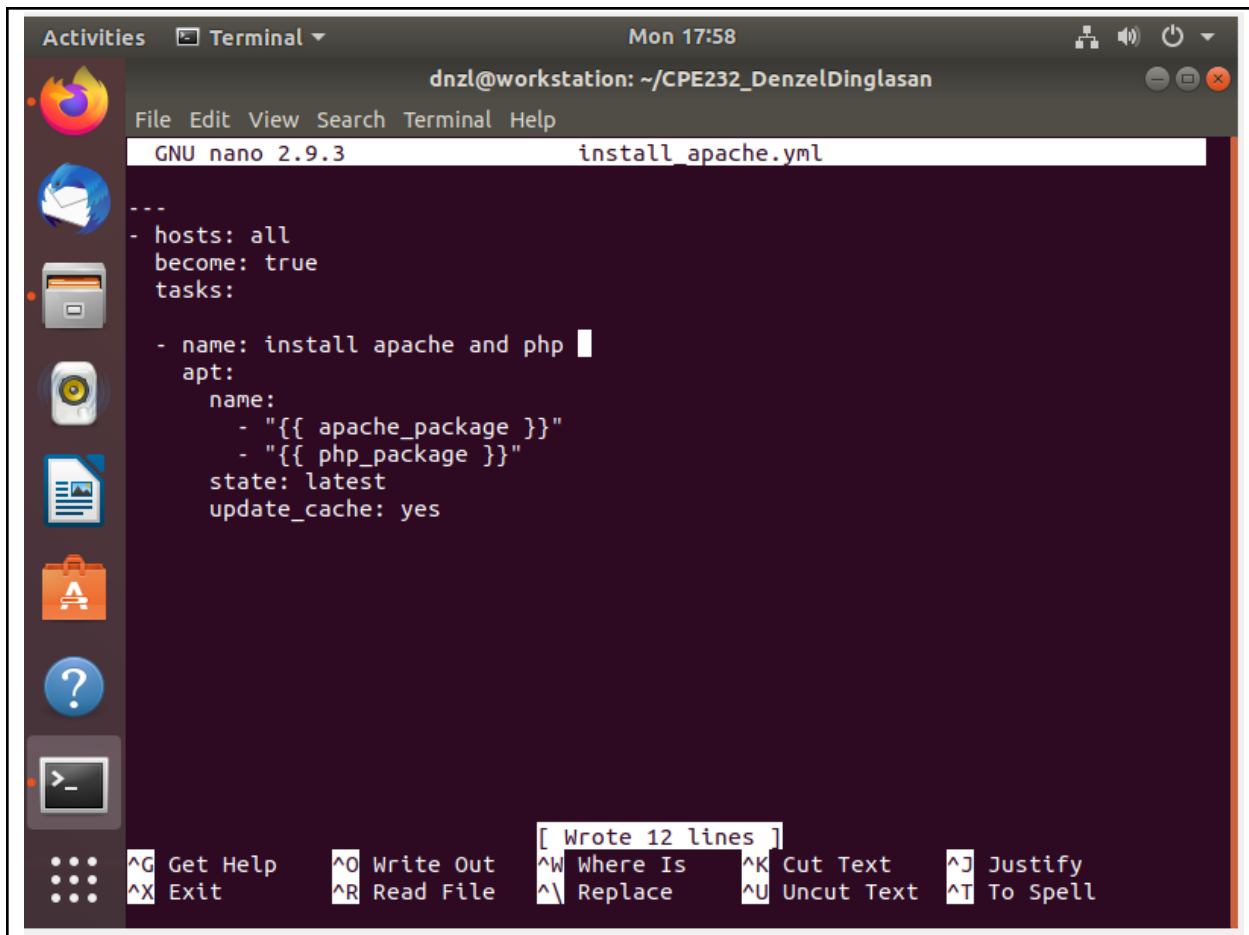
PLAY RECAP *****
*
192.168.56.102      : ok=2    changed=0    unreachable=0    failed=0
skipped=1    rescued=0    ignored=0
192.168.56.105      : ok=2    changed=0    unreachable=0    failed=0
skipped=1    rescued=0    ignored=0
```

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
```



Activities Terminal Mon 17:58

dnzl@workstation: ~/CPE232_DenzelDinglasan

File Edit View Search Terminal Help

GNU nano 2.9.3 install_apache.yml

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

[Wrote 12 lines]

^G Get Help	^O Write Out	^W Where Is	^K Cut Text	^J Justify
^X Exit	^R Read File	^\ Replace	^U Uncut Text	^T To Spell

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

```

dnzl@workstation:~/CPE232_DenzelDinglasan$ ansible-playbook --ask-become-pass i
ninstall_apache.yml
BECOME password:

PLAY [all] *****
*

TASK [Gathering Facts] *****
*
ok: [192.168.56.102]
ok: [192.168.56.105]

TASK [install apache and php] *****
*
fatal: [192.168.56.102]: FAILED! => {"msg": "The task includes an option with a
n undefined variable. The error was: 'apache_package' is undefined\n\nThe error
 appears to be in '/home/dnzl/CPE232_DenzelDinglasan/install_apache.yml': line
 6, column 5, but may\nbe elsewhere in the file depending on the exact syntax pr
 oblem.\n\nThe offending line appears to be:\n\n\n - name: install apache and p
 hp\n    ^ here\n"}
fatal: [192.168.56.105]: FAILED! => {"msg": "The task includes an option with a
n undefined variable. The error was: 'apache_package' is undefined\n\nThe error
 appears to be in '/home/dnzl/CPE232_DenzelDinglasan/install_apache.yml': line
 6, column 5, but may\nbe elsewhere in the file depending on the exact syntax pr
 oblem.\n\nThe offending line appears to be:\n\n\n - name: install apache and p
 hp\n    ^ here\n"}

PLAY RECAP *****
*
```

```

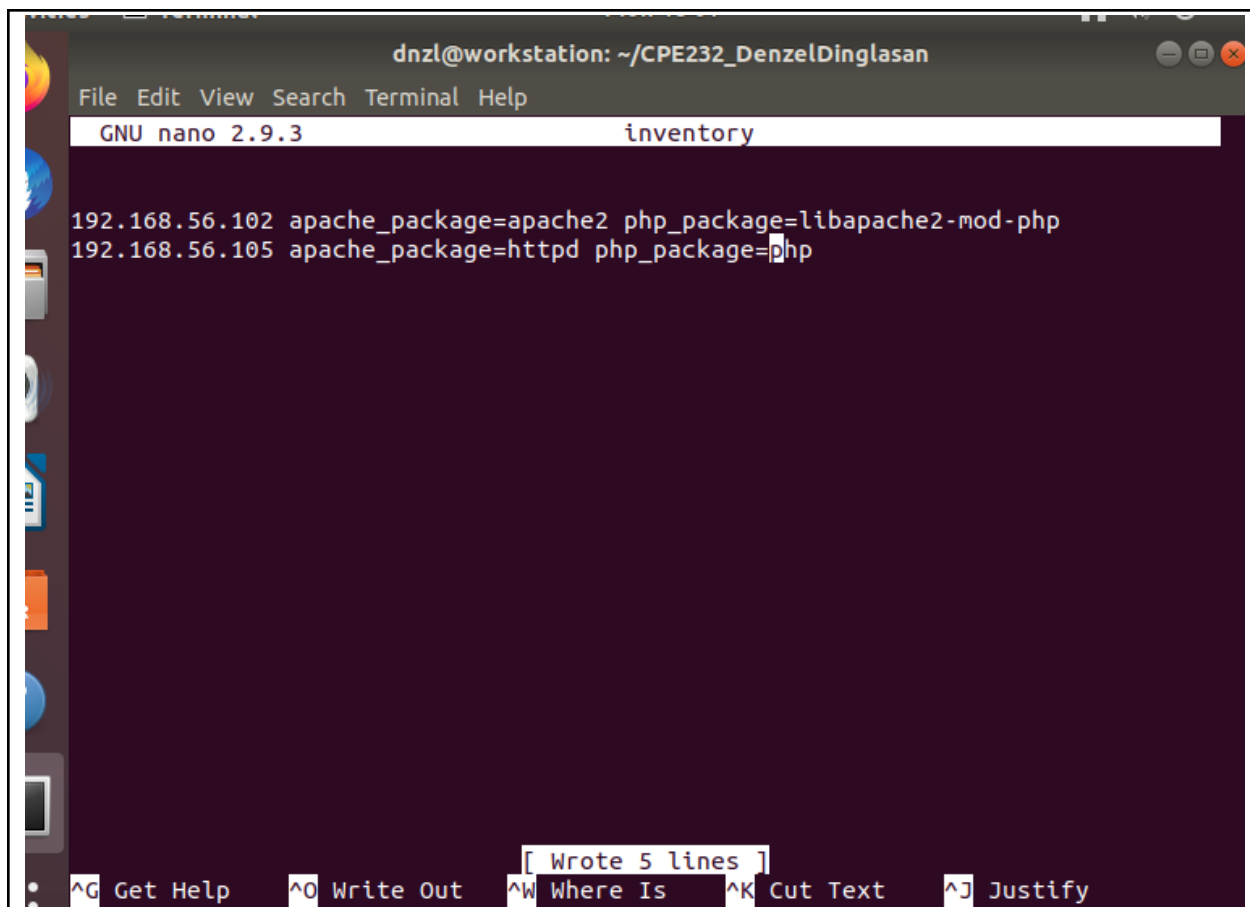
192.168.56.102      : ok=1    changed=0    unreachable=0    failed=1
skipped=0    rescued=0    ignored=0
192.168.56.105      : ok=1    changed=0    unreachable=0    failed=1
skipped=0    rescued=0    ignored=0
```

- 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.



```
dnzl@workstation: ~/CPE232_DenzelDinglasan
File Edit View Search Terminal Help
GNU nano 2.9.3 inventory

192.168.56.102 apache_package=apache2 php_package=libapache2-mod-php
192.168.56.105 apache_package=httpd php_package=php

[ Wrote 5 lines ]
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify
```

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](#)

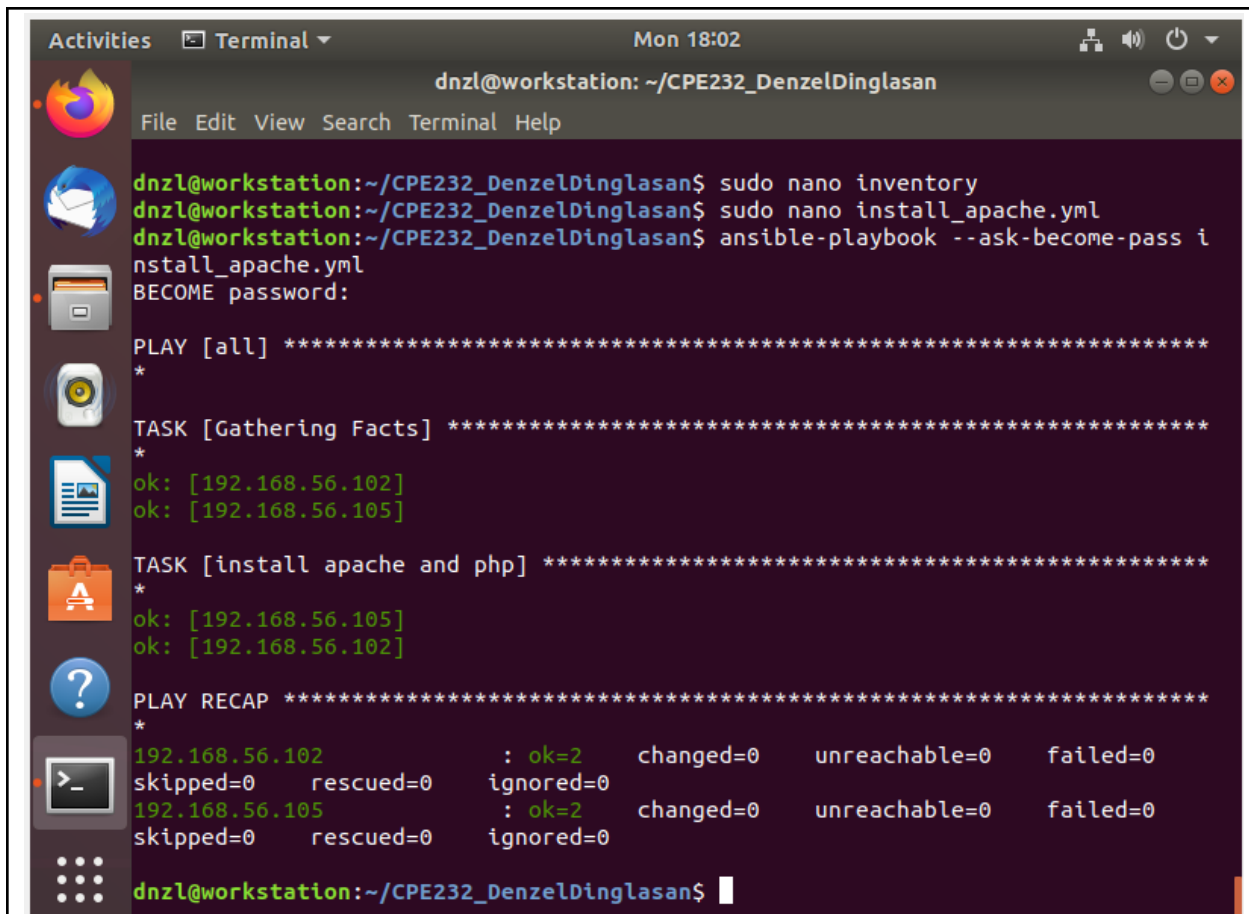
```
dnzl@workstation: ~/CPE232_DenzelDinglasan
File Edit View Search Terminal Help
GNU nano 2.9.3 install_apache.yml Modified

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

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

[ Read 12 lines ]
^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text  ^J Justify
```

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

A screenshot of a Linux terminal window titled 'Terminal' with a timestamp of 'Mon 18:02'. The user 'dnzl@workstation' is in the directory '~/CPE232_DenzelDinglasan'. The terminal shows the execution of an Ansible playbook named 'install_apache.yml'. The output includes sections for 'PLAY [all]', 'TASK [Gathering Facts]', and 'TASK [install apache and php]', each followed by a recap of the results for two hosts: '192.168.56.102' and '192.168.56.105'. The results show that all tasks were successful ('ok=2', 'changed=0', 'unreachable=0', 'failed=0'). The terminal window has a sidebar with various application icons and a menu bar with 'File', 'Edit', 'View', 'Search', 'Terminal', and 'Help'.

```
dnzl@workstation:~/CPE232_DenzelDinglasan$ sudo nano inventory
dnzl@workstation:~/CPE232_DenzelDinglasan$ sudo nano install_apache.yml
dnzl@workstation:~/CPE232_DenzelDinglasan$ ansible-playbook --ask-become-pass i
ninstall_apache.yml
BECOME password:

PLAY [all] *****
*

TASK [Gathering Facts] *****
*
ok: [192.168.56.102]
ok: [192.168.56.105]

TASK [install apache and php] *****
*
ok: [192.168.56.105]
ok: [192.168.56.102]

PLAY RECAP *****
*
192.168.56.102      : ok=2    changed=0    unreachable=0    failed=0
skipped=0    rescued=0    ignored=0
192.168.56.105      : ok=2    changed=0    unreachable=0    failed=0
skipped=0    rescued=0    ignored=0
dnzl@workstation:~/CPE232_DenzelDinglasan$
```

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?
Because it makes the playbook code easier to manage and enhance. Also refactoring it improves the performance of the code.
2. When do we use the “when” command in playbook?
If I’m going to make a conditional statement in my playbook code.

Conclusion:

In this activity, I learned how to refactor playbook codes and also make it run in CentOS. Through this activity, I can make better playbook codes that perform well and efficiently execute their purpose.

