Name: Shaniah Rose Hope M. Sumaoang	**Date Performed: Sept. 22, 2023**
Course/Section: CPE 232-CPE31S5	**Date Submitted: Sept. 27, 2023**
Instructor: Engr. Roman Richard	**Semester and SY: 1st Sem SY 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**:

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? **By issuing "git pull," the server can copy files from the GitHub repository to the Virtual Box. In this case, the displayed output reads "Already up to date" because the files in the GitHub repository were already current. This command ensures that the server is updated with any changes from the repository.**

```
srhmshan@localmachine:~$ cd CPE232_Sumaoang
srhmshan@localmachine:~/CPE232_Sumaoang$ git pull
Already up to date.
srhmshan@localmachine:~/CPE232_Sumaoang$
```

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
[ubuntu_servers]
192.168.5.45 ansible_user=srhmshan
192.168.5.59 ansible_user=srhmshan

[centos]
192.168.5.228 ansible_user=srhmshan
```

2.1 IP addresses in the inventory file.

```
srhmshan@localmachine:~/CPE232_Sumaoang$ ansible-playbook --ask-become-pass inst
all_apache.yml
BECOME password:

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

TASK [Gathering Facts] *********************************************************
ok: [192.168.5.228]
fatal: [192.168.5.45]: UNREACHABLE! => {"changed": false, "msg": "Failed to conn
ect to the host via ssh: ssh: connect to host 192.168.5.45 port 22: Connection t
imed out", "unreachable": true}
fatal: [192.168.5.59]: UNREACHABLE! => {"changed": false, "msg": "Failed to conn
ect to the host via ssh: ssh: connect to host 192.168.5.59 port 22: Connection t
imed out", "unreachable": true}

TASK [update repository index] *************************************************
[WARNING]: Updating cache and auto-installing missing dependency: python-apt
fatal: [192.168.5.228]: FAILED! => {"changed": false, "cmd": "apt-get update", "
msg": "[Errno 2] No such file or directory", "rc": 2}

PLAY RECAP *********************************************************************
192.168.5.228              : ok=1    changed=0    unreachable=0    failed=1    s
kipped=0    rescued=0    ignored=0
192.168.5.45               : ok=0    changed=0    unreachable=1    failed=0    s
kipped=0    rescued=0    ignored=0
192.168.5.59               : ok=0    changed=0    unreachable=1    failed=0    s
kipped=0    rescued=0    ignored=0
```

```
srhmshan@localmachine:~/CPE232_Sumaoang$ ansible-playbook --ask-become-pass inst
all_apache.yml
BECOME password:

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

TASK [Gathering Facts] *********************************************************
ok: [192.168.5.45]
ok: [192.168.5.59]
fatal: [192.168.5.228]: UNREACHABLE! => {"changed": false, "msg": "Failed to con
nect to the host via ssh: ssh: connect to host 192.168.5.228 port 22: Connection
 timed out", "unreachable": true}

TASK [update repository index] *************************************************
changed: [192.168.5.45]
changed: [192.168.5.59]

TASK [install apache2 package] *************************************************
ok: [192.168.5.59]
ok: [192.168.5.45]

TASK [add PHP support for apache] **********************************************
ok: [192.168.5.59]
ok: [192.168.5.45]

PLAY RECAP *********************************************************************
192.168.5.228              : ok=0    changed=0    unreachable=1    failed=0    s
kipped=0    rescued=0    ignored=0
192.168.5.45               : ok=4    changed=1    unreachable=0    failed=0    s
kipped=0    rescued=0    ignored=0
192.168.5.59               : ok=4    changed=1    unreachable=0    failed=0    s
```

2.2 I open, run and close CentOS and the 2 servers respectively because it crashes when opened at the same time.

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"
```

3.1 Editing install_apache.yml

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

```
srhmshan@localmachine:~/CPE232_Sumaoang$ ansible-playbook --ask-become-pass install_apac
he.yml
BECOME password:

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

TASK [Gathering Facts] *********************************************************
ok: [192.168.5.228]
fatal: [192.168.5.45]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to t
he host via ssh: ssh: connect to host 192.168.5.45 port 22: Connection timed out", "unre
achable": true}
fatal: [192.168.5.59]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to t
he host via ssh: ssh: connect to host 192.168.5.59 port 22: Connection timed out", "unre
achable": true}

TASK [update repository index] *************************************************
skipping: [192.168.5.228]

TASK [install apache2 package] *************************************************
skipping: [192.168.5.228]

TASK [add PHP support for apache] **********************************************
skipping: [192.168.5.228]

PLAY RECAP *********************************************************************
192.168.5.228              : ok=1    changed=0    unreachable=0    failed=0    skipped=3
    rescued=0    ignored=0
192.168.5.45               : ok=0    changed=0    unreachable=1    failed=0    skipped=0
    rescued=0    ignored=0
192.168.5.59               : ok=0    changed=0    unreachable=1    failed=0    skipped=0
    rescued=0    ignored=0
```

```
srhmshan@localmachine:~/CPE232_Sumaoang$ ansible-playbook --ask-become-pass install_apac
he.yml
BECOME password:

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

TASK [Gathering Facts] *********************************************************
ok: [192.168.5.45]
ok: [192.168.5.59]
fatal: [192.168.5.228]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to
the host via ssh: ssh: connect to host 192.168.5.228 port 22: Connection timed out", "un
reachable": true}

TASK [update repository index] *************************************************
changed: [192.168.5.59]
changed: [192.168.5.45]

TASK [install apache2 package] *************************************************
ok: [192.168.5.59]
ok: [192.168.5.45]

TASK [add PHP support for apache] **********************************************
ok: [192.168.5.45]
ok: [192.168.5.59]

PLAY RECAP *********************************************************************
192.168.5.228              : ok=0    changed=0    unreachable=1    failed=0    skipped=0
    rescued=0    ignored=0
192.168.5.45               : ok=4    changed=1    unreachable=0    failed=0    skipped=0
    rescued=0    ignored=0
192.168.5.59               : ok=4    changed=1    unreachable=0    failed=0    skipped=0
    rescued=0    ignored=0
```

3.2 I open, run and close CentOS and the 2 servers respectively because it crashes when opened at the same time.

CentOS (3 tasks) were skipped in the playbook, as those specific tasks were intended for Ubuntu.

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.

4.1 Editing install_apache.yml

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

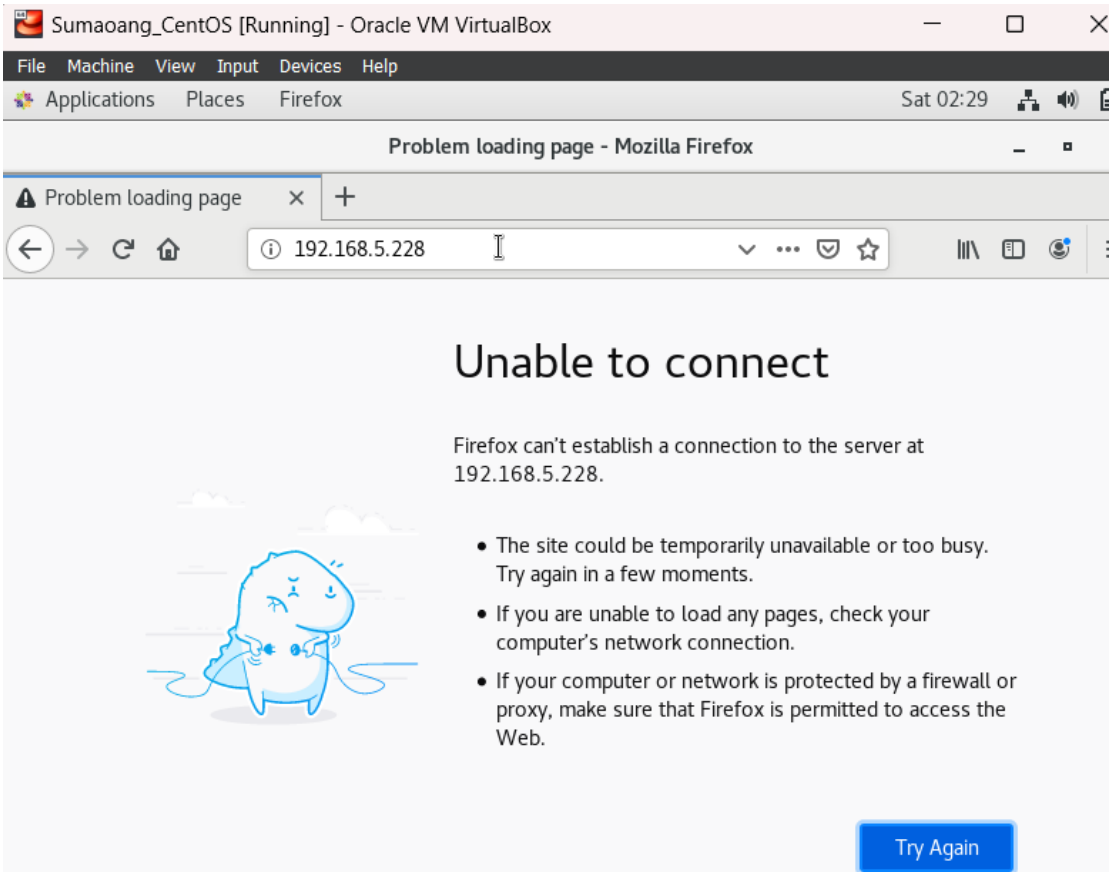
```
srhmshan@localmachine:~/CPE232_Sumaoang$ ansible-playbook --ask-become-pass install_apache.yml
BECOME password:

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

TASK [Gathering Facts] *********************************************************
ok: [192.168.5.228]
fatal: [192.168.5.45]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to the host via s
sh: ssh: connect to host 192.168.5.45 port 22: Connection timed out", "unreachable": true}
fatal: [192.168.5.59]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to the host via s
sh: ssh: connect to host 192.168.5.59 port 22: Connection timed out", "unreachable": true}

TASK [update repository index] *************************************************
skipping: [192.168.5.228]

TASK [install apache2 package] *************************************************
skipping: [192.168.5.228]

TASK [add PHP support for apache] **********************************************
skipping: [192.168.5.228]

TASK [update repository index] *************************************************
ok: [192.168.5.228]

TASK [install apache2 package] *************************************************
changed: [192.168.5.228]

TASK [add PHP support for apache] **********************************************
changed: [192.168.5.228]

PLAY RECAP *********************************************************************
192.168.5.228              : ok=4    changed=2    unreachable=0    failed=0    skipped=3    rescued=0
    ignored=0
192.168.5.45               : ok=0    changed=0    unreachable=1    failed=0    skipped=0    rescued=0
    ignored=0
192.168.5.59               : ok=0    changed=0    unreachable=1    failed=0    skipped=0    rescued=0
    ignored=0
```

```
TASK [Gathering Facts] *********************************************************************************
ok: [192.168.5.45]
ok: [192.168.5.59]
fatal: [192.168.5.228]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to the host via
ssh: ssh: connect to host 192.168.5.228 port 22: Connection timed out", "unreachable": true}

TASK [update repository index] *************************************************************************
changed: [192.168.5.45]
changed: [192.168.5.59]

TASK [install apache2 package] *************************************************************************
ok: [192.168.5.59]
ok: [192.168.5.45]

TASK [add PHP support for apache] **********************************************************************
ok: [192.168.5.59]
ok: [192.168.5.45]

TASK [update repository index] *************************************************************************
skipping: [192.168.5.45]
skipping: [192.168.5.59]

TASK [install apache2 package] *************************************************************************
skipping: [192.168.5.45]
skipping: [192.168.5.59]

TASK [add PHP support for apache] **********************************************************************
skipping: [192.168.5.45]
skipping: [192.168.5.59]

PLAY RECAP *********************************************************************************************
192.168.5.228              : ok=0    changed=0    unreachable=1    failed=0    skipped=0    rescued=0
    ignored=0
192.168.5.45               : ok=4    changed=1    unreachable=0    failed=0    skipped=3    rescued=0
    ignored=0
192.168.5.59               : ok=4    changed=1    unreachable=0    failed=0    skipped=3    rescued=0
    ignored=0
```

4.2 I open, run and close CentOS and the 2 servers respectively because it crashes when opened at the same time.

It is evident that CentOS skipped three tasks in the playbook since those three were meant for Ubuntu.

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.

```
[srhmshan@CentOS ~]$ systemctl status httpd
● 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)
```

5.2 Issue the following command to start the service:

sudo systemctl start httpd

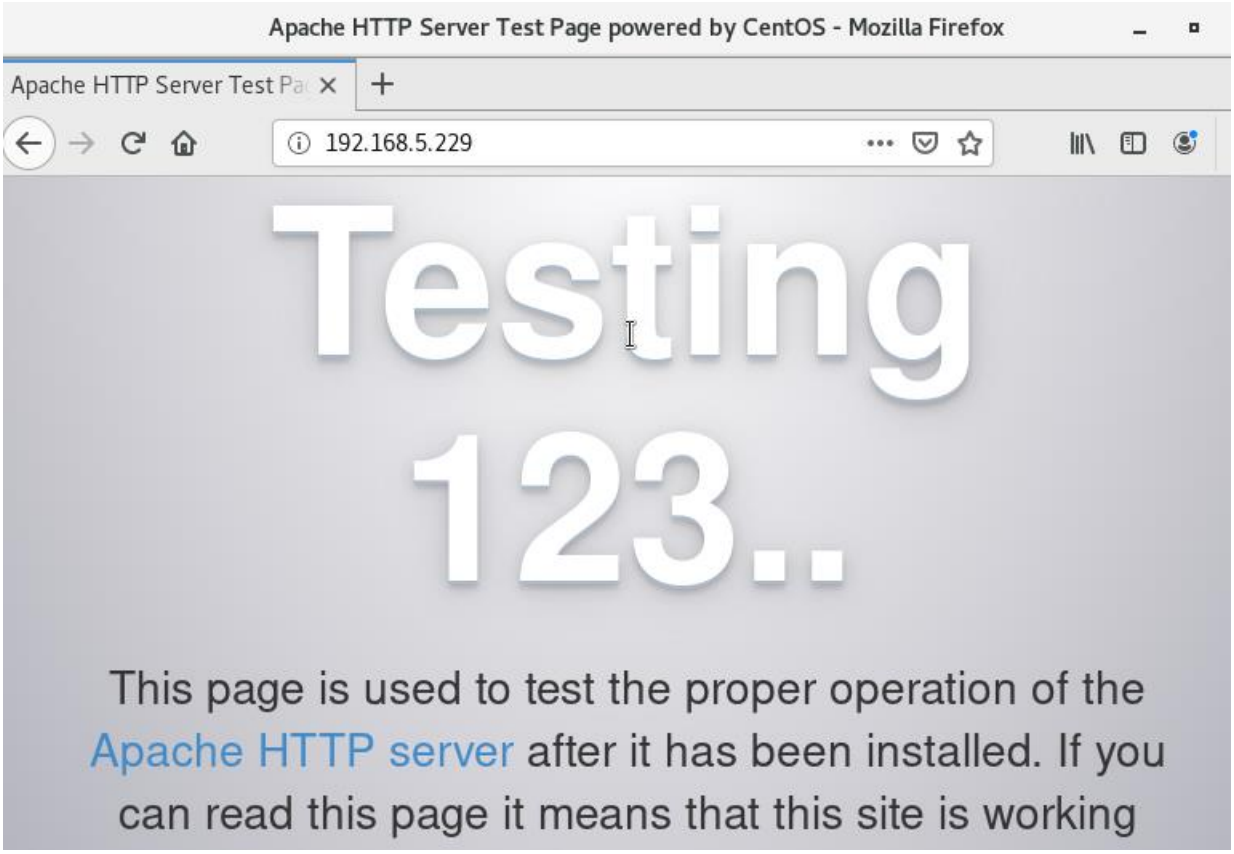
(When prompted, enter the sudo password)

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

(The result should be a success)

```
[srhmshan@CentOS ~]$ sudo systemctl start httpd
[sudo] password for srhmshan:
[srhmshan@CentOS ~]$ 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? **Yes** (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.

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

1.1 Editing the install_apache.yml file

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

```
srhmshan@localmachine:~/CPE232_Sumaoang$ ansible-playbook --ask-become-pass install_apache.yml
BECOME password:

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

TASK [Gathering Facts] *********************************************************************************
ok: [192.168.5.229]
fatal: [192.168.5.45]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to the host via s
sh: ssh: connect to host 192.168.5.45 port 22: Connection timed out", "unreachable": true}
fatal: [192.168.5.59]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to the host via s
sh: ssh: connect to host 192.168.5.59 port 22: Connection timed out", "unreachable": true}

TASK [update repository index] *************************************************************************
skipping: [192.168.5.229]

TASK [install apache2 and php packages for Ubuntu] *****************************************************
skipping: [192.168.5.229]

TASK [update repository index for CentOS] **************************************************************
ok: [192.168.5.229]

TASK [install apache2 package and php packages for CentOS] *********************************************
ok: [192.168.5.229]

PLAY RECAP *********************************************************************************************
192.168.5.229              : ok=3    changed=0    unreachable=0    failed=0    skipped=2    rescued=0
    ignored=0
192.168.5.45               : ok=0    changed=0    unreachable=1    failed=0    skipped=0    rescued=0
    ignored=0
192.168.5.59               : ok=0    changed=0    unreachable=1    failed=0    skipped=0    rescued=0
    ignored=0
```

```
srhmshan@localmachine:~/CPE232_Sumaoang$ ansible-playbook --ask-become-pass install_apache.yml
BECOME password:

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

TASK [Gathering Facts] *********************************************************************
ok: [192.168.5.59]
ok: [192.168.5.45]
fatal: [192.168.5.229]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to the host via
ssh: ssh: connect to host 192.168.5.229 port 22: Connection timed out", "unreachable": true}

TASK [update repository index] *************************************************************
changed: [192.168.5.59]
changed: [192.168.5.45]

TASK [install apache2 and php packages for Ubuntu] *****************************************
ok: [192.168.5.45]
ok: [192.168.5.59]

TASK [update repository index for CentOS] **************************************************
skipping: [192.168.5.45]
skipping: [192.168.5.59]

TASK [install apache2 package and php packages for CentOS] *********************************
skipping: [192.168.5.45]
skipping: [192.168.5.59]

PLAY RECAP *********************************************************************************
192.168.5.229              : ok=0    changed=0    unreachable=1    failed=0    skipped=0    rescued=0
    ignored=0
192.168.5.45               : ok=3    changed=1    unreachable=0    failed=0    skipped=2    rescued=0
    ignored=0
192.168.5.59               : ok=3    changed=1    unreachable=0    failed=0    skipped=2    rescued=0
    ignored=0
```

1.2 I open, run and close CentOS and the 2 servers respectively because it crashes when opened at the same time.\

Activated the updated 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"
```

Make sure to save the file and exit.

```
  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
      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"
```

2.1 Editing the install_apache.yml file

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

```
srhmshan@localmachine:~/CPE232_Sumaoang$ ansible-playbook --ask-become-pass install_apache.yml
BECOME password:

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

TASK [Gathering Facts] *********************************************************************************
ok: [192.168.5.229]
fatal: [192.168.5.45]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to the host via s
sh: ssh: connect to host 192.168.5.45 port 22: Connection timed out", "unreachable": true}
fatal: [192.168.5.59]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to the host via s
sh: ssh: connect to host 192.168.5.59 port 22: Connection timed out", "unreachable": true}

TASK [install apache2 and php packages for Ubuntu] *****************************************************
skipping: [192.168.5.229]

TASK [install apache2 package and php packages for CentOS] *********************************************
ok: [192.168.5.229]

PLAY RECAP *********************************************************************************************
192.168.5.229              : ok=2    changed=0    unreachable=0    failed=0    skipped=1    rescued=0
    ignored=0
192.168.5.45               : ok=0    changed=0    unreachable=1    failed=0    skipped=0    rescued=0
    ignored=0
192.168.5.59               : ok=0    changed=0    unreachable=1    failed=0    skipped=0    rescued=0
    ignored=0
```

```
srhmshan@localmachine:~/CPE232_Sumaoang$ ansible-playbook --ask-become-pass install_apache.yml
BECOME password:

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

TASK [Gathering Facts] *********************************************************************************
ok: [192.168.5.59]
ok: [192.168.5.45]
fatal: [192.168.5.229]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to the host via
ssh: ssh: connect to host 192.168.5.229 port 22: Connection timed out", "unreachable": true}

TASK [install apache2 and php packages for Ubuntu] *****************************************************
ok: [192.168.5.45]
ok: [192.168.5.59]

TASK [install apache2 package and php packages for CentOS] *********************************************
skipping: [192.168.5.45]
skipping: [192.168.5.59]

PLAY RECAP *********************************************************************************************
192.168.5.229              : ok=0    changed=0    unreachable=1    failed=0    skipped=0    rescued=0
    ignored=0
192.168.5.45               : ok=2    changed=0    unreachable=0    failed=0    skipped=1    rescued=0
    ignored=0
192.168.5.59               : ok=2    changed=0    unreachable=0    failed=0    skipped=1    rescued=0
    ignored=0
```

2.2 I open, run and close CentOS and the 2 servers respectively because it crashes when opened at the same time.

Another activation of the updated 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
```

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

```
TASK [install apache and php] ***************************************************************
fatal: [192.168.5.229]: FAILED! => {"msg": "The task includes an option with an undefined variable. T
he error was: 'apache_package' is undefined\n\nThe error appears to be in '/home/srhmshan/CPE232_Suma
oang/install_apache.yml': line 6, column 5, but may\nbe elsewhere in the file depending on the exact
syntax problem.\n\nThe offending line appears to be:\n\n\n  - name: install apache and php\n    ^ her
e\n"}
```

The task wasn't 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
[ubuntu_servers]
192.168.5.45 apache_package=apache2 php_package=libapache2-mod-php
192.168.5.59 apache_package=apache2 php_package=libapache2-mod-php

[centos]
192.168.5.229 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

```
  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
```

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

```
srhmshan@localmachine:~/CPE232_Sumaoang$ ansible-playbook --ask-become-pass install_apache.yml
BECOME password:

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

TASK [Gathering Facts] *********************************************************************************
ok: [192.168.5.45]
ok: [192.168.5.230]
ok: [192.168.5.59]

TASK [install apache and php] **************************************************************************
ok: [192.168.5.59]
ok: [192.168.5.45]
ok: [192.168.5.230]

PLAY RECAP *********************************************************************************************
192.168.5.230              : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0
    ignored=0
192.168.5.45               : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0
    ignored=0
192.168.5.59               : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0
    ignored=0
```

A successful output was obtained upon executing the playbook after upgrading my RAM.

Supplementary Activity:

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

```
  GNU nano 6.2                                  /etc/ansible/hosts
[localhost]
127.0.0.1 ansible_connection=local

[CentOS]
192.168.56.230
```

```
  GNU nano 6.2                              install_apache_redhat.yml
---
- hosts: CentOS
  become: true
  tasks:

  - name: install apache in redhat OS
    package:
      name: httpd
      state: latest
      update_cache: yes
    when: ansible_distribution == "CentOS"
```

```
srhmshan@localmachine:~/CPE232_Sumaoang$ ansible-playbook --ask-become-pass install_apache_redhat.yml
BECOME password:

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

TASK [Gathering Facts] *********************************************************************************
ok: [192.168.5.230]

TASK [install apache in redhat OS] *********************************************************************
ok: [192.168.5.230]

PLAY RECAP *********************************************************************************************
192.168.5.230              : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0
    ignored=0
```

The play recap indicates an "ok" status, affirming that the CentOS server has the apache2 package installed and the repository was previously updated.

Reflections:

Answer the following:

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

   The importance of refactoring playbook code lies in its ability to enhance readability and accommodate the incorporation of new features. This procedure streamlines additional code while keeping the program's output and behavior unchanged.

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

   When working with a playbook, the "when" command becomes essential for integrating multiple conditions into the process.

Conclusion:

During this activity, I've learned some useful lessons about working with Linux-based systems like Ubuntu and CentOS 7. One key takeaway is the importance of using the "when" command wisely in playbooks. I've also learned that simplifying and improving code, known as code refactoring, is crucial. It's like a skill because it helps keep playbook code easy to understand and modify, making it easier to add new features without messing up the program's current behavior. All of this has contributed to making automation solutions more efficient and reliable.