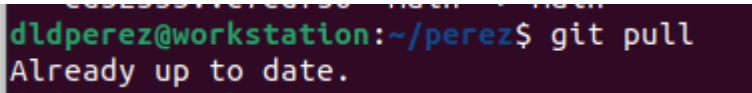


Name: Dean Lenard D. Perez	Date Performed: 27-09-24
Course/Section: CPE 212 / CPE31S21	Date Submitted: 29-09-24
Instructor: Engr Robin Valenzuela	Semester and SY: 1 st Sem 2024
Activity 5: Consolidating Playbook plays	
1. Objectives: 1.1 Use when command in playbook for different OS distributions 1.2 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>dldperez@workstation:~/perez\$ git pull Already up to date.</pre> <p>Nothing was updated because there were no new changes in the repository.</p>	

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."

```
didperez@workstation: ~/perez$ cat inventory
192.168.56.113
192.168.56.114
```

Added the IP address of the CentOS in the inventory.

```
didperez@workstation: ~/perez$ ansible-playbook --ask-become-pass install_apache.yml
PLAY [all] *****
TASK [Gathering Facts] *****
ok: [192.168.56.113]
[DEPRECATION WARNING]: Distribution centos 9 on host 192.168.56.114 should use
/usr/libexec/platform-python, but is using /usr/bin/python for backward compatibility with prior
Ansible releases. A future Ansible release will default to using the discovered platform python
for this host. See
https://docs.ansible.com/ansible/2.10/reference_appendices/interpreter_discovery.html for more
information. This feature will be removed in version 2.12. Deprecation warnings can be disabled by
setting deprecation_warnings=False in ansible.cfg.
ok: [192.168.56.114]
TASK [update repository index] *****
[WARNING]: Updating cache and auto-installing missing dependency: python3-apt
Fatal: [192.168.56.114]: FAILED! == ("changed": false, "cmd": "apt-get update", "msg": "[Errno 2] No such file or directory: b'apt-
get'", "rc": 2)
changed: [192.168.56.113]
TASK [install apache2 package] *****
ok: [192.168.56.113]
TASK [add PHP support for apache] *****
ok: [192.168.56.113]
PLAY RECAP *****
192.168.56.113      : ok=4  changed=1  unreachable=0  failed=0  skipped=0  rescued=0  ignored=0
192.168.56.114      : ok=1  changed=0  unreachable=0  failed=1  skipped=0  rescued=0  ignored=0
```

After adding the IP address of CentOS, the result shows that it has failed on the CentOS server.

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.

```
dldperez@workstation: ~/perez
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"

^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute  ^C Location
^X Exit      ^R Read File ^L Replace   ^U Paste     ^J Justify  ^_ Go To Line
```

This is the configured `install_apache.yml`.

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

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

PLAY [all] *****

TASK [Gathering Facts] *****
ok: [192.168.56.113]
[DEPRECATION WARNING]: Distribution centos 9 on host 192.168.56.114 should use /usr/libexec/platform-python, but is using /usr/bin/python for backward compatibility with prior Ansible releases. A future Ansible release will default to using the discovered platform python for this host. See https://docs.ansible.com/ansible/2.10/reference_appendices/interpreter_discovery.html for more information. This feature will be removed in version 2.12. Deprecation warnings can be disabled by setting deprecation_warnings=False in ansible.cfg.
ok: [192.168.56.114]

TASK [update repository index] *****
skipping: [192.168.56.114]
changed: [192.168.56.113]

TASK [install apache2 package] *****
skipping: [192.168.56.114]
ok: [192.168.56.113]

TASK [add PHP support for apache] *****
skipping: [192.168.56.114]
ok: [192.168.56.113]

PLAY RECAP *****
192.168.56.113      : ok=4    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
192.168.56.114      : ok=1    changed=0    unreachable=0    failed=0    skipped=3    rescued=0    ignored=0

dldperez@workstation: ~/perez$
```

The changes were applied to the Ubuntu server which is the server1 and was skipped on the CentOS server because of the conditional statement that will only install it on an Ubuntu server.

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.

```
dldperez@workstation: ~/perez
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"

    - 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"

^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify
```

EDITED `install_apache.yml`.

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

```
dldperez@workstation: ~/perez
dldperez@workstation:~/perez$ nano install_apache.yml
dldperez@workstation:~/perez$ ansible-playbook --ask-become-pass install_apache.yml
BECOME password:

PLAY [all] *****

TASK [Gathering Facts] *****
ok: [192.168.56.113]
[DEPRECATION WARNING]: Distribution centos 9 on host 192.168.56.114 should use
/usr/libexec/platform-python, but is using /usr/bin/python for backward
compatibility with prior Ansible releases. A future Ansible release will
default to using the discovered platform python for this host. See https://docs
.ansible.com/ansible/2.10/reference_appendices/interpreter_discovery.html for
more information. This feature will be removed in version 2.12. Deprecation
warnings can be disabled by setting deprecation_warnings=False in ansible.cfg.
ok: [192.168.56.114]

TASK [update repository index] *****
skipping: [192.168.56.114]
changed: [192.168.56.113]

TASK [install apache2 package] *****
skipping: [192.168.56.114]
ok: [192.168.56.113]

TASK [add PHP support for apache] *****
skipping: [192.168.56.114]
ok: [192.168.56.113]

TASK [update repository index] *****
skipping: [192.168.56.113]
ok: [192.168.56.114]

TASK [install apache2 package] *****
skipping: [192.168.56.113]
changed: [192.168.56.114]

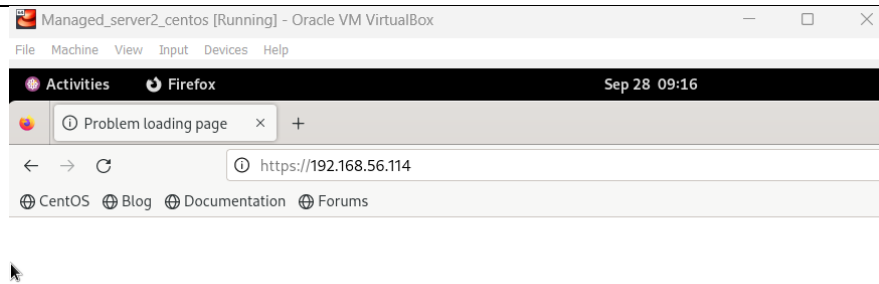
TASK [add PHP support for apache] *****
skipping: [192.168.56.113]
changed: [192.168.56.114]

PLAY RECAP *****
192.168.56.113      : ok=4    changed=1    unreachable=0    failed=0    skipped=3    rescued=0    ignored=0
192.168.56.114      : ok=4    changed=2    unreachable=0    failed=0    skipped=3    rescued=0    ignored=0

dldperez@workstation:~/perez$
```

The result shows that it will only install to servers that follows the condition and it skips the server when it doesn't follow the conditions.

5. To verify the installations, go to CentOS VM and type its IP address on the browser. Was it successful? The answer is no. It's because the httpd service or the Apache HTTP server in the CentOS is not yet active. Thus, you need to activate it first.



Unable to connect

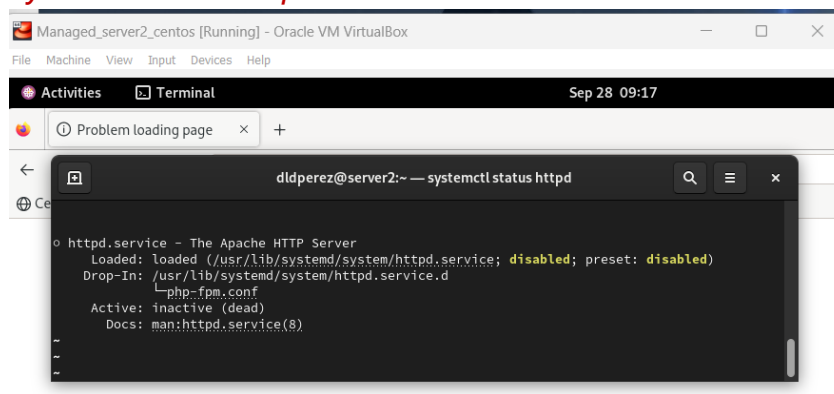
An error occurred during a connection to 192.168.56.114.

- 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.

This shows that it is unsuccessful meaning that httpd is not running yet.

5.1 To activate, go to the CentOS VM terminal and enter the following:

systemctl status httpd



It command shows that http is disabled.

The result of this command tells you that the service is inactive.

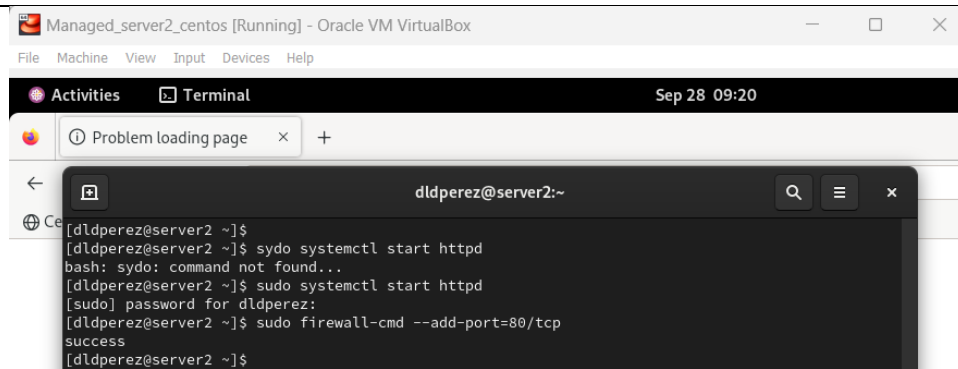
5.2 Issue the following command to start the service:

sudo systemctl start httpd

(When prompted, enter the sudo password)

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

(The result should be a success)



```
Managed_server2_centos [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help

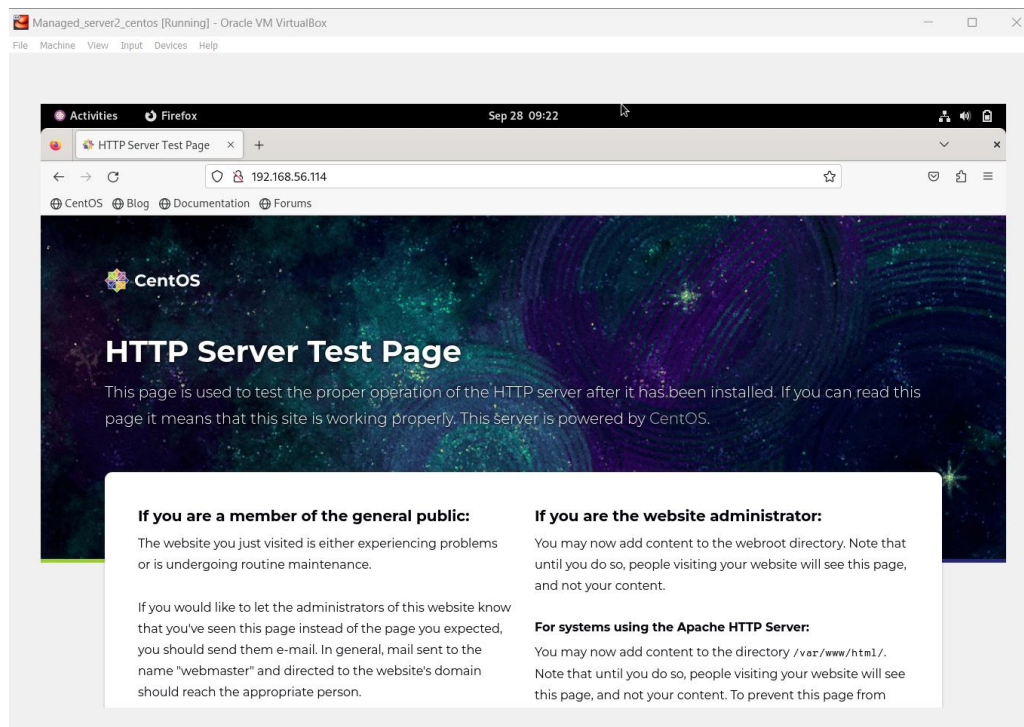
Activities Terminal Sep 28 09:20

Problem loading page x +

dldperez@server2:~$
[dldperez@server2 ~]$ sydo systemctl start httpd
bash: sydo: command not found...
[dldperez@server2 ~]$ sudo systemctl start httpd
[sudo] password for dldperez:
[dldperez@server2 ~]$ sudo firewall-cmd --add-port=80/tcp
success
[dldperez@server2 ~]$
```

The commands ran successfully.

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)



After starting the http service and configuring the firewall, the IP address was successful in showing the Http Server test page.

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"
```

```
dldperez@workstation: ~/perez
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 package 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 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.

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

PLAY [all] *****

TASK [Gathering Facts] *****
ok: [192.168.56.113]
ok: [192.168.56.114]

TASK [update repository index Ubuntu] *****
skipping: [192.168.56.114]
changed: [192.168.56.113]

TASK [install apache2 package for Ubuntu] *****
skipping: [192.168.56.114]
ok: [192.168.56.113]

TASK [update repository index for CentOS] *****
skipping: [192.168.56.113]
ok: [192.168.56.114]

TASK [install apache and php packages packages for CentOS] *****
skipping: [192.168.56.113]
ok: [192.168.56.114]

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

dldperez@workstation:~/perez$
```

It was successful and the condition that was set was followed.

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"

```

```

dldperez@workstation: ~/perez
GNU nano 6.2                                install_apache.yml *
---
- 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 == "Ubuntu"

    - name: install apache and php packages 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.

```
dldperez@workstation: ~/perez
dldperez@workstation:~/perez$ nano install_apache.yml
dldperez@workstation:~/perez$
dldperez@workstation:~/perez$ ansible-playbook --ask-become-pass install_apache.yml
BECOME password:

PLAY [all] *****

TASK [Gathering Facts] *****
ok: [192.168.56.113]
ok: [192.168.56.114]

TASK [install apache2 and php package for Ubuntu] *****
skipping: [192.168.56.114]
ok: [192.168.56.113]

TASK [install apache and php packages packages for CentOS] *****
skipping: [192.168.56.113]
ok: [192.168.56.114]

PLAY RECAP *****
192.168.56.113      : ok=2    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
192.168.56.114      : ok=2    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0

dldperez@workstation:~/perez$
```

It was successful and the condition that was set was followed but now the list is getting smaller.

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
```

```
dldperez@workstation: ~/perez
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.

```
dldperez@workstation: ~/perez
dldperez@workstation:~/perez$ nano install_apache.yml
dldperez@workstation:~/perez$ ansible-playbook --ask-become-pass install_apache.yml
BECOME password:

PLAY [all] *****

TASK [Gathering Facts] *****
ok: [192.168.56.113]
ok: [192.168.56.114]

TASK [install apache and php] *****
fatal: [192.168.56.113]: FAILED! => ["msg": "The task includes an option with an undefined variable. The error was: 'apache_package' is undefined\nThe error appears to be in '/home/dldperez/perez/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  - name: install apache and php\n  ^ here\n"]
fatal: [192.168.56.114]: FAILED! => ["msg": "The task includes an option with an undefined variable. The error was: 'apache_package' is undefined\nThe error appears to be in '/home/dldperez/perez/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  - name: install apache and php\n  ^ here\n"]

PLAY RECAP *****
192.168.56.113      : ok=1    changed=0    unreachable=0    failed=1    skipped=0    rescued=0    ignored=0
192.168.56.114      : ok=1    changed=0    unreachable=0    failed=1    skipped=0    rescued=0    ignored=0
```

It failed because we created the variables but it did not have a value.

- 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.

```
dldperez@workstation: ~/perez
GNU nano 6.2 inventory
192.168.56.113 apache_package=apache2 php_package=libapache2-mod-php
192.168.56.114 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](#)

```
dldperez@workstation: ~/perez
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.

```
dldperez@workstation: ~/perez
dldperez@workstation:~/perez$ nano inventory
dldperez@workstation:~/perez$ nano install_apache.yml
dldperez@workstation:~/perez$ ansible-playbook --ask-become-pass install_apache.yml
BECOME password:

PLAY [all] *****

TASK [Gathering Facts] *****
ok: [192.168.56.113]
ok: [192.168.56.114]

TASK [install apache and php] *****
ok: [192.168.56.113]
ok: [192.168.56.114]

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

It was successful and the condition that was set was followed and the list looks neat.

Supplementary Activity:

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

```
dldperez@workstation: ~/perez
GNU nano 6.2                                install_in_RedHatOS.yml *
---
- hosts: all
  become: yes
  tasks:
    - name: Install Apache on Red Hat OS
      yum:
        name: httpd
        state: present
        when: ansible_distribution == "Red Hat"
```

Created a playbook file *install_in_redHatOS.yml*.

```
dldperez@workstation: ~/perez
dldperez@workstation:~/perez$ nano install_in_RedHatOS.yml
dldperez@workstation:~/perez$ ansible-playbook --ask-become-pass install_in_RedHatOS.yml
BECOME password:

PLAY [all] *****

TASK [Gathering Facts] *****
ok: [192.168.56.113]
ok: [192.168.56.114]

TASK [Install Apache on Red Hat OS] *****
skipping: [192.168.56.113]
skipping: [192.168.56.114]

PLAY RECAP *****
192.168.56.113      : ok=1    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
192.168.56.114      : ok=1    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
```

Executed the playbook file but it doesn't change anything because the conditional statement was not met.

GITHUB LINK

<https://github.com/dldperez/perez.git>

Reflections:

Answer the following:

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

It is important because it makes the playbook efficient and easier to maintain, it also looks neat and clean. It is useful when we will install a lot of packages or when it is a large-scale installation, doing it one-by-one is not productive and ideal, refactoring it makes our life easier.

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

We use it when we want the command or script to have a condition when it is being executed. It is used for conditional logic. It will determine which command will execute based on the condition that the user specified.

