Name: Torrecampo, Juan Piolo S.	**Date Performed:** Oct 11, 2022
Course/Section: CPE 232 / CPE31S22	**Date Submitted:** Oct 11, 2022
Instructor: Dr. Jonathan Taylar	**Semester and SY:** 1st Sem, 2022 - 2023

Activity 7: Managing Files and Creating Roles in Ansible

1. **Objectives:**

1.1 Manage files in remote servers

1.2 Implement roles in ansible

2. **Discussion**:

In this activity, we look at the concept of copying a file to a server. We are going to create a file into our git repository and use Ansible to grab that file and put it into a particular place so that we could do things like customize a default website, or maybe install a default configuration file. We will also implement roles to consolidate plays.

Task 1: Create a file and copy it to remote servers

1. Using the previous directory we created, create a directory, and named it "*files*." Create a file inside that directory and name it "*default_site.html*." Edit the file and put basic HTML syntax. Any content will do, as long as it will display text later. Save the file and exit.

```
piolo@workstation:~/CPE232_piolo/ansible$ mkdir files
piolo@workstation:~/CPE232_piolo/ansible$ ll
total 48
drwxrwxr-x 3 piolo piolo  4096 Oct 11 07:58 ./
drwxrwxr-x 4 piolo piolo  4096 Oct  4 09:20 ../
-rw-rw-r-- 1 piolo piolo   124 Oct  4 11:46 ansible.cfg
-rw-rw-r-- 1 piolo piolo  1971 Oct  4 11:46 ansible-guide.txt
drwxrwxr-x 2 piolo piolo  4096 Oct 11 07:58 files/
-rw-rw-r-- 1 piolo piolo   840 Oct  4 11:46 install_apache2.yml
-rw-rw-r-- 1 piolo piolo   473 Oct  4 11:46 install_apache.yml
-rw-rw-r-- 1 piolo piolo   580 Oct  4 11:46 inventory
-rw-rw-r-- 1 piolo piolo 12288 Oct  4 11:46 .inventory.swp
-rw-rw-r-- 1 piolo piolo  1175 Oct  4 11:46 site.yml
```

Figure 1.1. Creating a directory named "files".

```
piolo@workstation:~/CPE232_piolo/ansible$ touch files/default_site.html
piolo@workstation:~/CPE232_piolo/ansible$ ll files
total 8
drwxrwxr-x 2 piolo piolo 4096 Oct 11 07:59 ./
drwxrwxr-x 3 piolo piolo 4096 Oct 11 07:58 ../
-rw-rw-r-- 1 piolo piolo    0 Oct 11 07:59 default_site.html
```

Figure 1.2. Creating a HTML file named "default_site.html" inside of the "files" directory.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Sample Webpage</title>
  </head>
  <body>
    <h1>This is a sample webpage.</h1>
    <h3>Sample Article 1</h3>
    <p>Lorem ipsum dolor sit amet, officia excepteur ex fugiat reprehenderit enim labore culpa sint ad nisi Lorem pariatur mollit ex esse exercitation amet.
Nisi anim cupidatat excepteur officia. Reprehenderit nostrud nostrud ipsum Lorem est aliquip amet voluptate voluptate dolor minim nulla est proident. Nostrud
 officia pariatur ut officia. Sit irure elit esse ea nulla sunt ex occaecat reprehenderit commodo officia dolor Lorem duis laboris cupidatat officia voluptat
e. Culpa proident adipisicing id nulla nisi laboris ex in Lorem sunt duis officia eiusmod. Aliqua reprehenderit commodo ex non excepteur duis sunt velit enim
. Voluptate laboris sint cupidatat ullamco ut ea consectetur et est culpa et culpa duis.</p>
    <h3>Sample Article 2</h3>
    <p>Lorem ipsum dolor sit amet, officia excepteur ex fugiat reprehenderit enim labore culpa sint ad nisi Lorem pariatur mollit ex esse exercitation amet.
Nisi anim cupidatat excepteur officia. Reprehenderit nostrud nostrud ipsum Lorem est aliquip amet voluptate voluptate dolor minim nulla est proident. Nostrud
 officia pariatur ut officia. Sit irure elit esse ea nulla sunt ex occaecat reprehenderit commodo officia dolor Lorem duis laboris cupidatat officia voluptat
e. Culpa proident adipisicing id nulla nisi laboris ex in Lorem sunt duis officia eiusmod. Aliqua reprehenderit commodo ex non excepteur duis sunt velit enim
. Voluptate laboris sint cupidatat ullamco ut ea consectetur et est culpa et culpa duis.</p>
    <h3>Sample Article 3</h3>
    <p>Lorem ipsum dolor sit amet, officia excepteur ex fugiat reprehenderit enim labore culpa sint ad nisi Lorem pariatur mollit ex esse exercitation amet.
Nisi anim cupidatat excepteur officia. Reprehenderit nostrud nostrud ipsum Lorem est aliquip amet voluptate voluptate dolor minim nulla est proident. Nostrud
 officia pariatur ut officia. Sit irure elit esse ea nulla sunt ex occaecat reprehenderit commodo officia dolor Lorem duis laboris cupidatat officia voluptat
e. Culpa proident adipisicing id nulla nisi laboris ex in Lorem sunt duis officia eiusmod. Aliqua reprehenderit commodo ex non excepteur duis sunt velit enim
. Voluptate laboris sint cupidatat ullamco ut ea consectetur et est culpa et culpa duis.</p>
    <h3>Sample Article 4</h3>
    <p>Lorem ipsum dolor sit amet, officia excepteur ex fugiat reprehenderit enim labore culpa sint ad nisi Lorem pariatur mollit ex esse exercitation amet.
Nisi anim cupidatat excepteur officia. Reprehenderit nostrud nostrud ipsum Lorem est aliquip amet voluptate voluptate dolor minim nulla est proident. Nostrud
 officia pariatur ut officia. Sit irure elit esse ea nulla sunt ex occaecat reprehenderit commodo officia dolor Lorem duis laboris cupidatat officia voluptat
e. Culpa proident adipisicing id nulla nisi laboris ex in Lorem sunt duis officia eiusmod. Aliqua reprehenderit commodo ex non excepteur duis sunt velit enim
. Voluptate laboris sint cupidatat ullamco ut ea consectetur et est culpa et culpa duis.</p>
  </body>
</html>
```

Figure 1.3. Adding some contents inside of default_site.html.

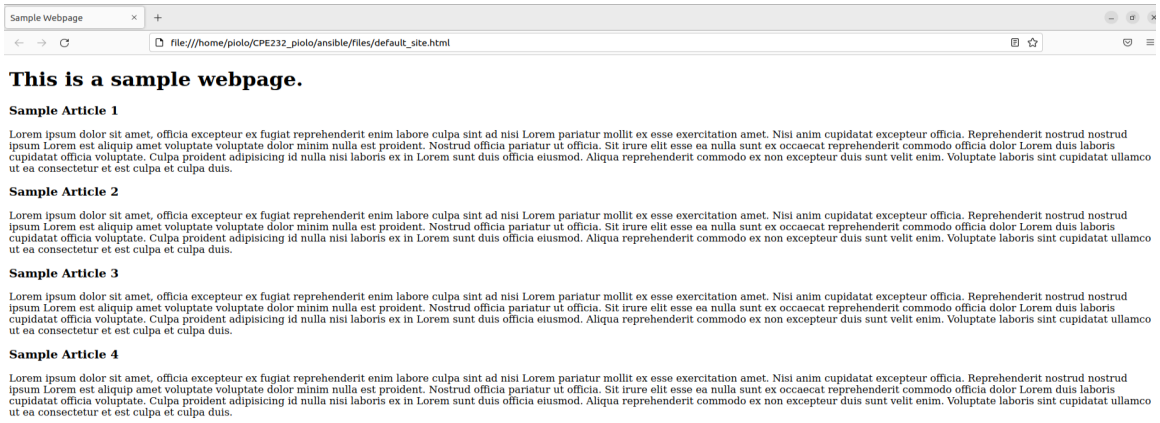

Figure 1.4. Viewing the webpage named "default_site.html" in firefox.

2. Edit the *site.yml* file and just below the *web_servers* play, create a new file to copy the default html file for site:
   - name: copy default html file for site

     tags: apache, apache2, httpd

     copy:

       src: default_site.html

       dest: /var/www/html/index.html

       owner: root

       group: root

       mode: 0644

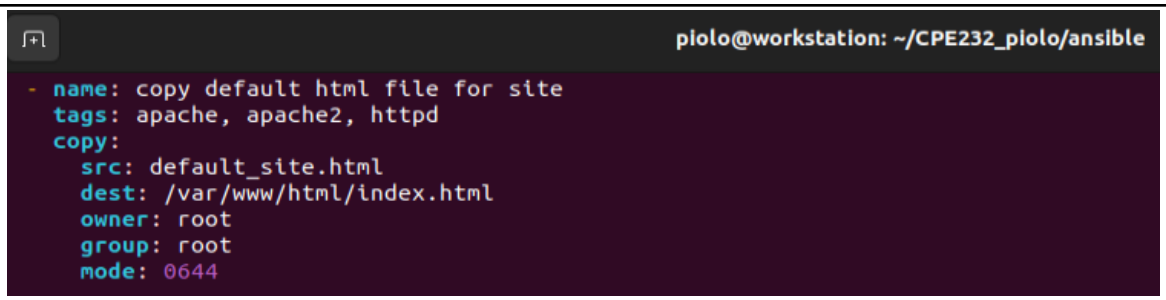

Figure 1.5. The screenshot above shows the appended task in site.yml.

3. Run the playbook *site.yml*. Describe the changes.

- **There are changes in the web _servers group. This task is pertaining to the task that was created in the previous step which created an "index.html" to the target system and copy the content of the source html file.**

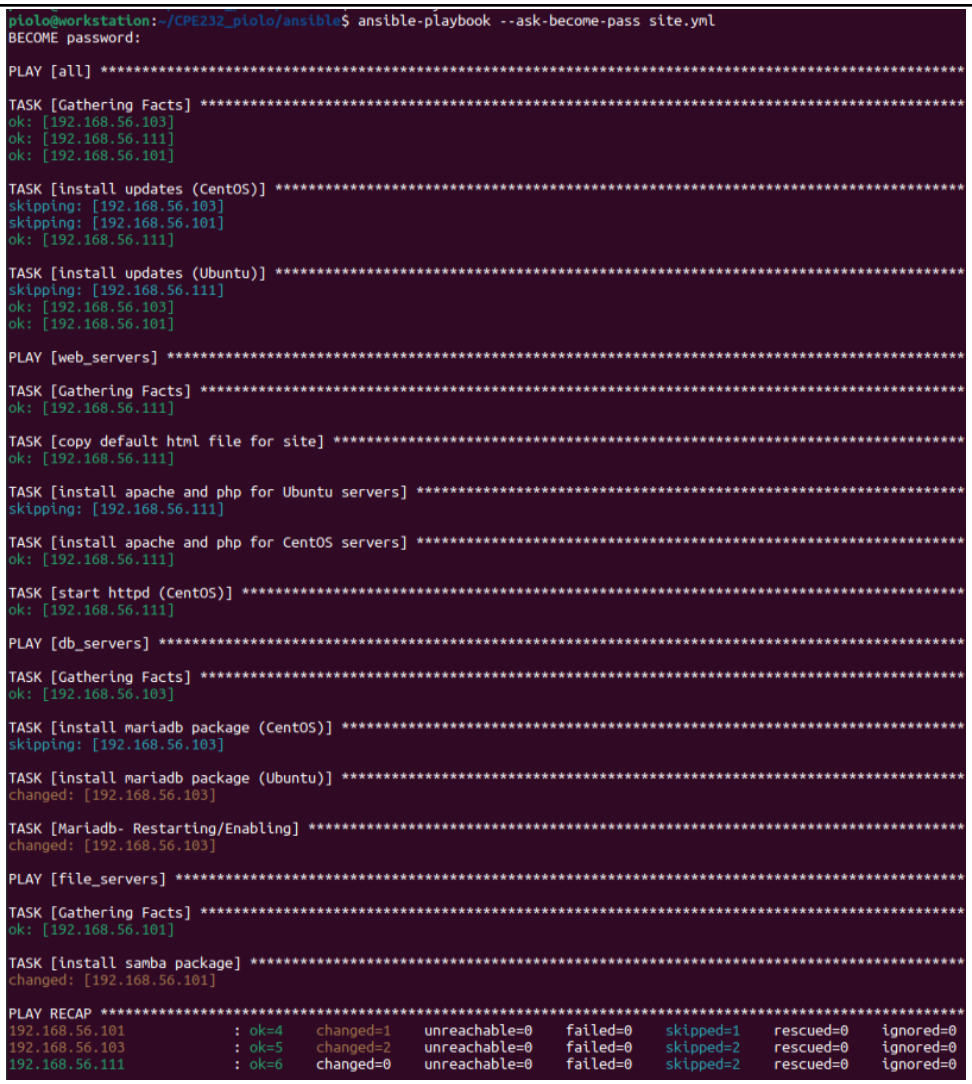

Figure 1.6. The screenshot above shows the result after executing the playbook.

4. Go to the remote servers (*web_servers*) listed in your inventory. Use cat command to check if the index.html is the same as the local repository file (*default_site.html*). Do both for Ubuntu and CentOS servers. On the CentOS server, go to the browser and type its IP address. Describe the output.

Figure 1.7. The screenshot above shows the output when doing a cat command in CentOS.

Figure 1.8. This shows that the created "default_site.yml" is running in the web server.

5. Sync your local repository with GitHub and describe the changes.

Task 2: Download a file and extract it to a remote server

1. Edit the site.yml. Just before the web_servers play, create a new play:
   - hosts: workstations
     become: true
     tasks:

- name: install unzip
  package:
    name: unzip

- name: install terraform
  unarchive:
    src: https://releases.hashicorp.com/terraform/0.12.28/terraform_0.12.28_linux_amd64.zip
    dest: /usr/local/bin
    remote_src: yes
    mode: 0755
    owner: root
    group: root

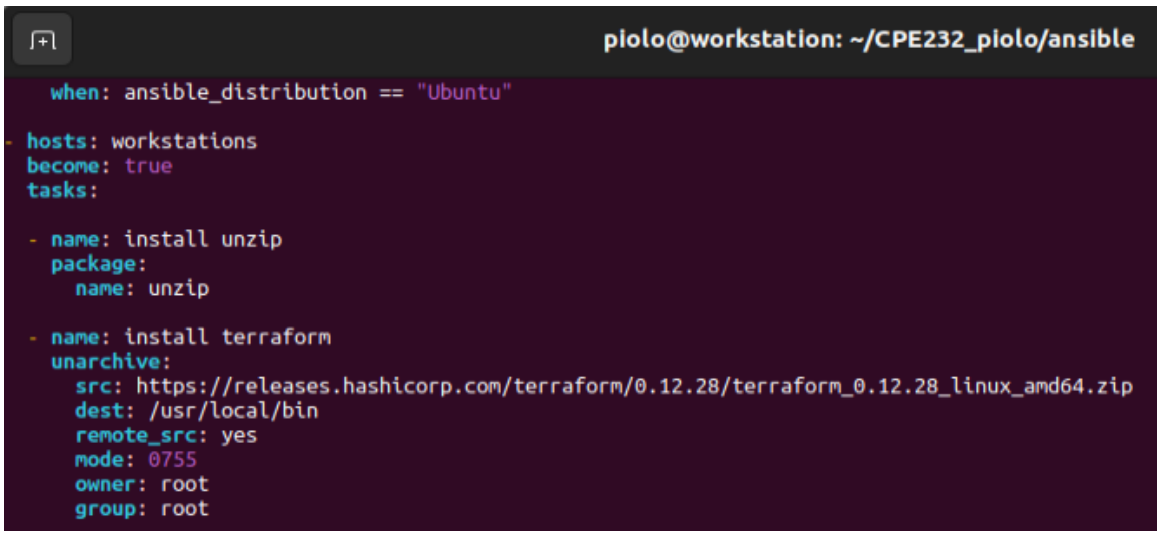

Figure 2.1. The screenshot above shows the appended task for site.yml.

2. Edit the inventory file and add workstations group. Add any Ubuntu remote server. Make sure to remember the IP address.

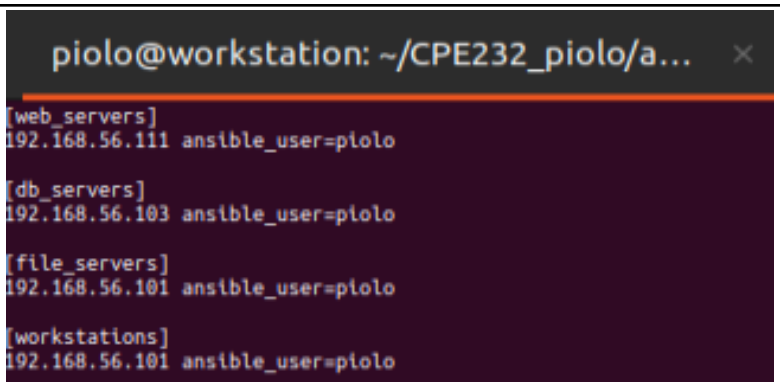

Figure 2.2. The screenshot above shows the edited inventory file.

3. Run the playbook. Describe the output.

- **The output has changes in the assigned workstation server. It installs the terraform**

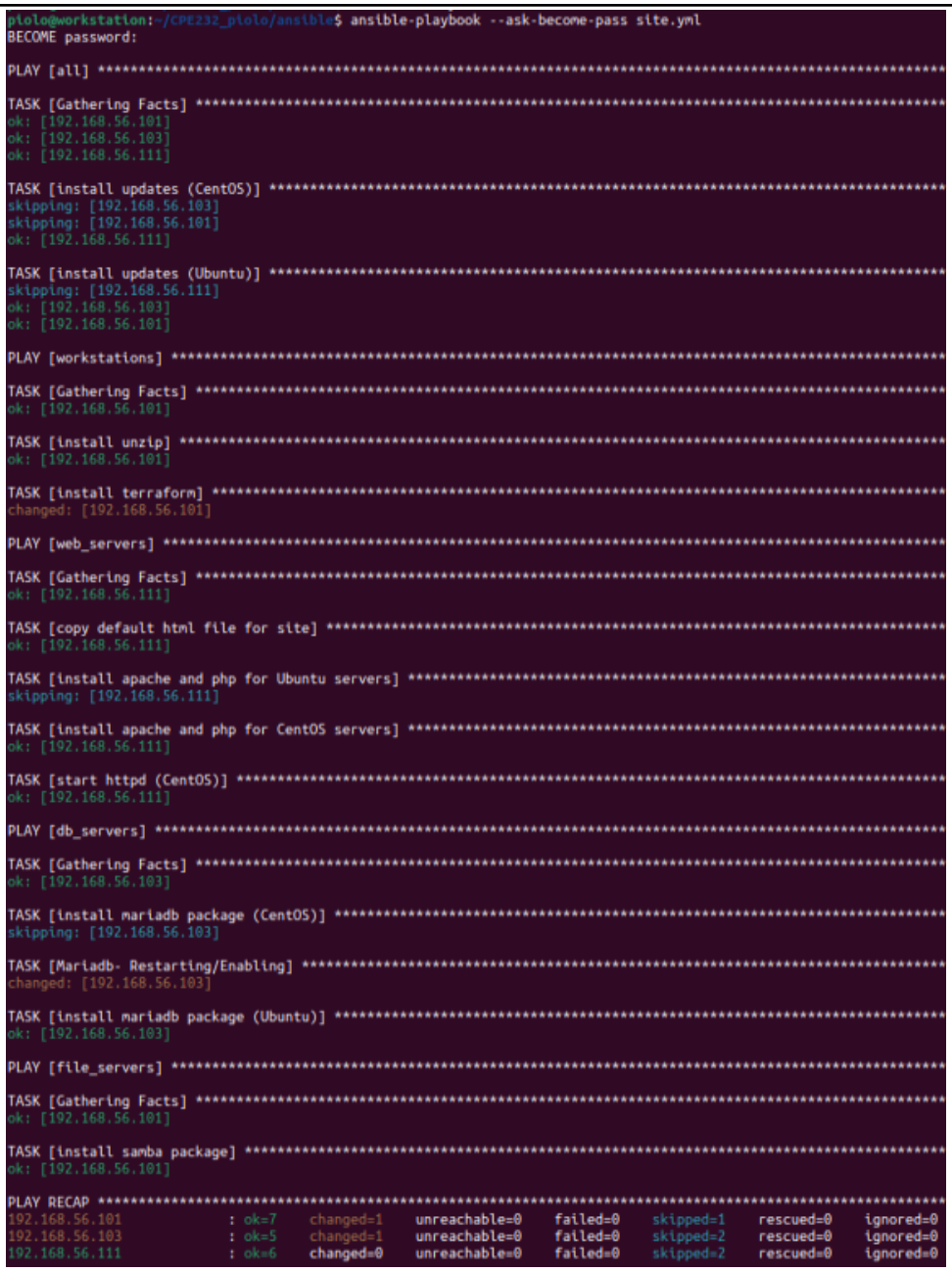

Figure 2.3. The screenshot above shows the result after executing the playbook.

4. On the Ubuntu remote workstation, type terraform to verify installation of terraform. Describe the output.
    - **The output gives me the available command in terraform. It means that the terraform is being recognized by the server.**

Figure 2.4. The screenshot above shows the verification of terraform in Ubuntu server 2.

Task 3: Create roles

1. Edit the site.yml. Configure roles as follows: (make sure to create a copy of the old site.yml file because you will be copying the specific plays for all groups)

```
---
- hosts: all
  become: true
  pre_tasks:

  - name: update repository index (CentOS)
    tags: always
    dnf:
      update_cache: yes
    changed_when: false
    when: ansible_distribution == "CentOS"
  - name: install updates (Ubuntu)
    tags: always
    apt:
      update_cache: yes
    changed_when: false
    when: ansible_distribution == "Ubuntu"

- hosts: all
  become: true
  roles:
    -  base

- hosts: workstations
  become: true
  roles:
    - workstations

- hosts: web_servers
  become: true
  roles:
    - web_servers

- hosts: db_servers
  become: true
  roles:
    - db_servers

- hosts: file_servers
  become: true
  roles:
    - file_servers
```

Save the file and exit.

```
piolo@workstation:~/CPE232_piolo/ansible$ cp site.yml site_copy.yml
piolo@workstation:~/CPE232_piolo/ansible$ ll
total 40
drwxrwxr-x 3 piolo piolo 4096 Oct 11 09:43 ./
drwxrwxr-x 4 piolo piolo 4096 Oct  4 09:20 ../
-rw-rw-r-- 1 piolo piolo  124 Oct  4 11:46 ansible.cfg
-rw-rw-r-- 1 piolo piolo 1971 Oct  4 11:46 ansible-guide.txt
drwxrwxr-x 2 piolo piolo 4096 Oct 11 08:13 files/
-rw-rw-r-- 1 piolo piolo  840 Oct  4 11:46 install_apache2.yml
-rw-rw-r-- 1 piolo piolo  473 Oct  4 11:46 install_apache.yml
-rw-rw-r-- 1 piolo piolo  197 Oct 11 09:40 inventory
-rw-rw-r-- 1 piolo piolo 2204 Oct 11 09:43 site_copy.yml
-rw-rw-r-- 1 piolo piolo 2204 Oct 11 09:25 site.yml
piolo@workstation:~/CPE232_piolo/ansible$
```

Figure 3.1. Making a copy of site.yml.

piolo@workstation: ~/CPE232_piolo/ansible

```yaml
---

- hosts: all
  become: true
  tasks:

  - name: install updates (CentOS)
    tags: always
    dnf:
      update_cache: yes
    changed_when: false
    when: ansible_distribution == "CentOS"

  - name: install updates (Ubuntu)
    tags: always
    apt:
      update_cache: yes
    changed_when: false
    when: ansible_distribution == "Ubuntu"


- hosts: all
  become: true
  roles:
    - base

- hosts: workstations
  become: true
  roles:
    - workstations

- hosts: web_servers
  become: true
  roles:
    - web_servers

- hosts: db_servers
  become: true
  roles:
    - db_servers

- hosts: file_servers
  become: true
  roles:
    - file_servers
```

Figure 3.2. The screenshot above shows the content of site.yml

2. Under the same directory, create a new directory and name it roles. Enter the roles directory and create new directories: base, web_servers, file_servers, db_servers and workstations. For each directory, create a directory and name it tasks.

```
piolo@workstation:~/CPE232_piolo/ansible$ mkdir roles
piolo@workstation:~/CPE232_piolo/ansible$ ll
total 44
drwxrwxr-x 4 piolo piolo 4096 Oct 11 09:48 ./
drwxrwxr-x 4 piolo piolo 4096 Oct  4 09:20 ../
-rw-rw-r-- 1 piolo piolo  124 Oct  4 11:46 ansible.cfg
-rw-rw-r-- 1 piolo piolo 1971 Oct  4 11:46 ansible-guide.txt
drwxrwxr-x 2 piolo piolo 4096 Oct 11 08:13 files/
-rw-rw-r-- 1 piolo piolo  840 Oct  4 11:46 install_apache2.yml
-rw-rw-r-- 1 piolo piolo  473 Oct  4 11:46 install_apache.yml
-rw-rw-r-- 1 piolo piolo  197 Oct 11 09:40 inventory
drwxrwxr-x 2 piolo piolo 4096 Oct 11 09:48 roles/
-rw-rw-r-- 1 piolo piolo 2204 Oct 11 09:43 site_copy.yml
-rw-rw-r-- 1 piolo piolo  663 Oct 11 09:45 site.yml
piolo@workstation:~/CPE232_piolo/ansible$
```

Figure 3.3. Creating the roles directory.

```
piolo@workstation:~/CPE232_piolo/ansible/roles$ mkdir base web_servers file_servers db_servers workstations
piolo@workstation:~/CPE232_piolo/ansible/roles$ ll
total 28
drwxrwxr-x 7 piolo piolo 4096 Oct 11 09:54 ./
drwxrwxr-x 4 piolo piolo 4096 Oct 11 09:52 ../
drwxrwxr-x 2 piolo piolo 4096 Oct 11 09:54 base/
drwxrwxr-x 2 piolo piolo 4096 Oct 11 09:54 db_servers/
drwxrwxr-x 2 piolo piolo 4096 Oct 11 09:54 file_servers/
drwxrwxr-x 2 piolo piolo 4096 Oct 11 09:54 web_servers/
drwxrwxr-x 2 piolo piolo 4096 Oct 11 09:54 workstations/
piolo@workstation:~/CPE232_piolo/ansible/roles$
```

Figure 3.4. Creating the base, web_servers, file_servers, db_servers and workstations directories.

```
piolo@workstation:~/CPE232_piolo/ansible/roles$ ls base
tasks
piolo@workstation:~/CPE232_piolo/ansible/roles$ ls db_servers
tasks
piolo@workstation:~/CPE232_piolo/ansible/roles$ ls file_servers
tasks
piolo@workstation:~/CPE232_piolo/ansible/roles$ ls web_servers
tasks
piolo@workstation:~/CPE232_piolo/ansible/roles$ ls workstations
tasks
piolo@workstation:~/CPE232_piolo/ansible/roles$
```

Figure 3.5. Listing all of the contents of the recently created directories after creating the tasks directories.

3. Go to tasks for all directory and create a file. Name it main.yml. In each of the tasks for all directories, copy and paste the code from the old site.yml file. Show all contents of main.yml files for all tasks.

```
piolo@workstation:~/CPE232_piolo/ansible/roles$ ll
total 28
drwxrwxr-x 7 piolo piolo 4096 Oct 11 09:54 ./
drwxrwxr-x 4 piolo piolo 4096 Oct 11 09:52 ../
drwxrwxr-x 3 piolo piolo 4096 Oct 11 09:59 base/
drwxrwxr-x 3 piolo piolo 4096 Oct 11 09:59 db_servers/
drwxrwxr-x 3 piolo piolo 4096 Oct 11 09:59 file_servers/
drwxrwxr-x 3 piolo piolo 4096 Oct 11 09:59 web_servers/
drwxrwxr-x 3 piolo piolo 4096 Oct 11 09:59 workstations/
piolo@workstation:~/CPE232_piolo/ansible/roles$ $sudo cat /home/piolo/CPE232_piolo/ansible/site_copy.yml > base/tasks/main.yml
piolo@workstation:~/CPE232_piolo/ansible/roles$ $sudo cat /home/piolo/CPE232_piolo/ansible/site_copy.yml > db_servers/tasks/main.yml
piolo@workstation:~/CPE232_piolo/ansible/roles$ $sudo cat /home/piolo/CPE232_piolo/ansible/site_copy.yml > file_servers/tasks/main.yml
piolo@workstation:~/CPE232_piolo/ansible/roles$ $sudo cat /home/piolo/CPE232_piolo/ansible/site_copy.yml > web_servers/tasks/main.yml
piolo@workstation:~/CPE232_piolo/ansible/roles$ $sudo cat /home/piolo/CPE232_piolo/ansible/site_copy.yml > workstations/tasks/main.yml
piolo@workstation:~/CPE232_piolo/ansible/roles$
```

Figure 3.6. The screenshot above shows the creation of the main.yml in the tasks sub directories

.

4. Run the site.yml playbook and describe the output.

- **The output stays the same even when we organize the tasks in folders.**

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

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

TASK [Gathering Facts] *********************************************************************************************************************
ok: [192.168.56.103]
ok: [192.168.56.101]
ok: [192.168.56.111]

TASK [install updates (CentOS)] ************************************************************************************************************
skipping: [192.168.56.103]
skipping: [192.168.56.101]
ok: [192.168.56.111]

TASK [install updates (Ubuntu)] ************************************************************************************************************
skipping: [192.168.56.111]
The error appears to be in '/home/piolo/Desktop/CPE232_piolo/ansible/roles/workstations/tasks/main.yml': line 3, column 3, but may
be elsewhere in the file depending on the exact syntax problem.

The offending line appears to be:


- hosts: all
ok: [192.168.56.103]
ok: [192.168.56.101]

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

TASK [Gathering Facts] *********************************************************************************************************************
ok: [192.168.56.111]
ok: [192.168.56.103]
ok: [192.168.56.101]

PLAY [workstations] ************************************************************************************************************************

TASK [Gathering Facts] *********************************************************************************************************************
ok: [192.168.56.101]

PLAY [web_servers] *************************************************************************************************************************

TASK [Gathering Facts] *********************************************************************************************************************
ok: [192.168.56.111]

PLAY [db_servers] **************************************************************************************************************************

TASK [Gathering Facts] *********************************************************************************************************************
ok: [192.168.56.103]

PLAY [file_servers] ************************************************************************************************************************

TASK [Gathering Facts] *********************************************************************************************************************
ok: [192.168.50.101]

PLAY RECAP *********************************************************************************************************************************
192.168.56.101             : ok=5    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
192.168.56.103             : ok=4    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
192.168.56.111             : ok=4    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
```

Figure 3.7. The screenshot above shows the result after executing the edited playbook.

Github Link: https://github.com/piolotorrecampo/CPE232_piolo

Reflections:

Answer the following:

1. What is the importance of creating roles?

    - An ansible role provides an additional feature for ansible to make a complex playbook easier. This is by allowing a system administrator to break a huge playbook into multiple files with their respective directory. In this method of breaking down playbooks, we can reuse the subdivided files depending on the task. In task 3 of this activity perform the implementation of roles which we are used to create a role inside of the site.yml and their respective tasks are in the tree directory named "roles".

2. What is the importance of managing files?

    - The importance of managing files in every workflow using ansible is that we can use the pre-built commands of the software to use in performing manipulation of files inside of a remote server. This feature can use to edit the ownership of the file by using the mode variable with its 4 octal value depending on the target ownership. The task 1 and 2 shows some of the commands that are used in managing files. Like src parameter for specifying a path for a file and dest parameter for it destination in the remote address

Conclusion:

This activity successfully fulfill its goal to implement on how we can use ansible in managing file and creating roles in the remote servers. In brief explanation, the task 1 introduces some of parameters used in managing files like scr for specification of file in the local system, dest for the path of the target location inside of the remote server, mode for specification of ownership of the file, owner that pertains to the possessor of the file, and group for the file group possessor.

The task 2 is pretty interesting to execute because it installs a software called terraform. In order to install the terraform, we specify first in the task is the installation of the unzip package. Wherein this package is used to unzip the downloaded zip file of terraform in the second task. After downloading it, it uset to copy the executable file to the /usr/local/bin and specify the mode, root, and group possessor. Where this task is used to fall under the managing files in terms of objective.

On the other hand, task 3 pertains to the next and last object which is the creation of roles. In this task, I completely edit the site.yml file that specifies the name of the directory under the role parameter depending on its hsots parameter. And we have been duplicating the content of the old site.yml under the roles subdirectory named main.yml. In this method we can easily distinguish the files in every task unlike having them in one large playbook file. Overall, this activity gives me a better understanding on how managing files work and creating roles in ansible

Honor Pledge:

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