Name: ROALLOS, Jean Gabriel Vincent G.	**Date Performed:** 10 / 03 / 2025
Course/Section: CPE212 - CPE31S2	**Date Submitted:** 10 / 03 / 2025
Instructor: Engr. Robin Valenzuela	**Semester and SY:** 1st, 2025 - 2026

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.
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
```

site.yaml

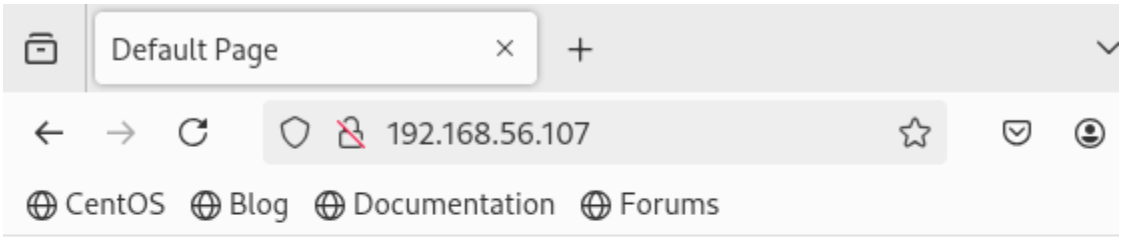
```
20    - hosts: web_servers
50      - name: Copy default HTML File for Site
51        tags: apache, apache2, httpd
52        copy:
53          src: default_site.html
54          dest: /var/www/html/index.html
55          owner: root
56          group: root
57          mode: 0644
```

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

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

TASK [Gathering Facts] ********************************
ok: [192.168.56.112]
ok: [192.168.56.107]

TASK [Install Apache2 and PHP for Ubuntu Servers] ******
skipping: [192.168.56.107]
ok: [192.168.56.112]

TASK [Install Apache2 and PHP for CentOS Servers] ******
skipping: [192.168.56.112]
ok: [192.168.56.107]

TASK [Start httpd (CentOS)] ***************************
skipping: [192.168.56.112]
changed: [192.168.56.107]

TASK [Copy default HTML File for Site] ****************
ok: [192.168.56.112]
changed: [192.168.56.107]
```

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.

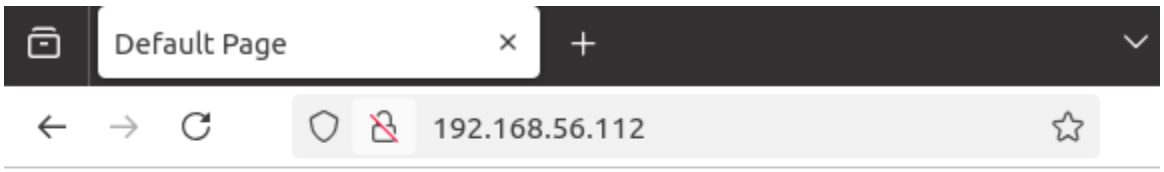
```
[roallos-centos@vbox ~]$ cat /var/www/html/index.html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Default Page</title>
</head>
<body>
    <h1>Welcome to the Default Page</h1>
    <p>This is a simple HTML document.</p>
</body>[roallos-centos@vbox ~]$
```



Welcome to the Default Page

This is a simple HTML document.

```
roallos-ubuntu@server1:~$ cat /var/www/html/index.html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Default Page</title>
</head>
<body>
    <h1>Welcome to the Default Page</h1>
    <p>This is a simple HTML document.</p>
</body>roallos-ubuntu@server1:~$
```



Welcome to the Default Page

This is a simple HTML document.

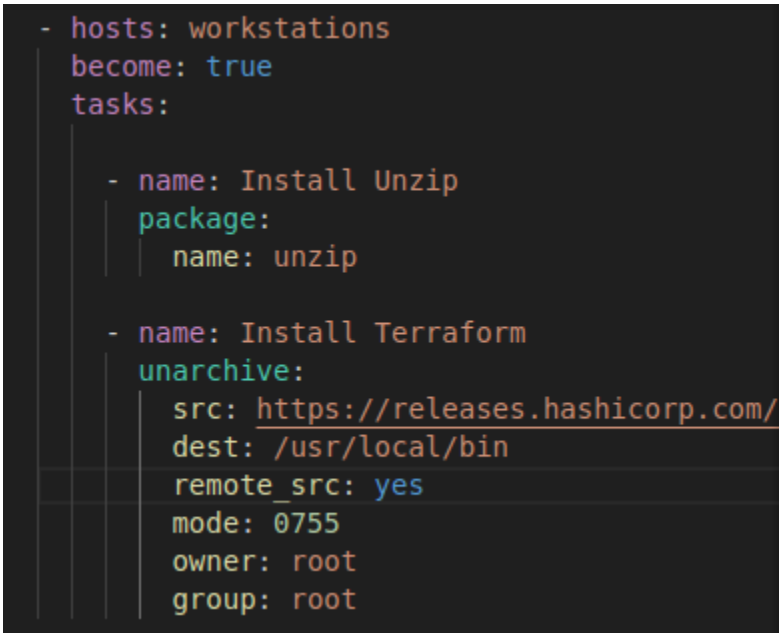
5. Sync your local repository with GitHub and describe the changes.

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



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

```
[UbuntuServer1]
192.168.56.112

[UbuntuServer2]
192.168.56.106

[CentOSServer2]
192.168.56.107 ansible_user=roallos-centos

[db_servers:children]
UbuntuServer1

[web_servers:children]
UbuntuServer1
CentOSServer2

[file_servers:children]
CentOSServer2

[workstations:children]
UbuntuServer1 # 192.168.56.112
```

3. Run the playbook. Describe the output.

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

TASK [Gathering Facts] *****************
ok: [192.168.56.112]

TASK [Install Unzip] *******************
ok: [192.168.56.112]

TASK [Install Terraform] ***************
changed: [192.168.56.112]
```

4. On the Ubuntu remote workstation, type terraform to verify installation of terraform. Describe the output.

```
roallos-ubuntu@server1:~$ terraform
Usage: terraform [-version] [-help] <command> [args]

The available commands for execution are listed below.
The most common, useful commands are shown first, followed by
less common or more advanced commands. If you're just getting
started with Terraform, stick with the common commands. For the
other commands, please read the help and docs before usage.
```

```
roallos-ubuntu@server1:~$ terraform -version
Terraform v0.12.28

Your version of Terraform is out of date! The latest version
is 1.13.3. You can update by downloading from https://www.terraform.io/download
s.html
roallos-ubuntu@server1:~$
```

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.

site-copy.yaml

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

  - name: Install Updates (CentOS)
    tags: always
    dnf:
      update_only: yes
      # update_cache: yes
    changed_when: false
    when: ansible_distribution == "CentOS"

  - name: Install Updates (Ubuntu)
    tags: always
    apt:
      # upgrade: dist
      update_cache: yes
    changed_when: false
    when: ansible_distribution == "Ubuntu"
```

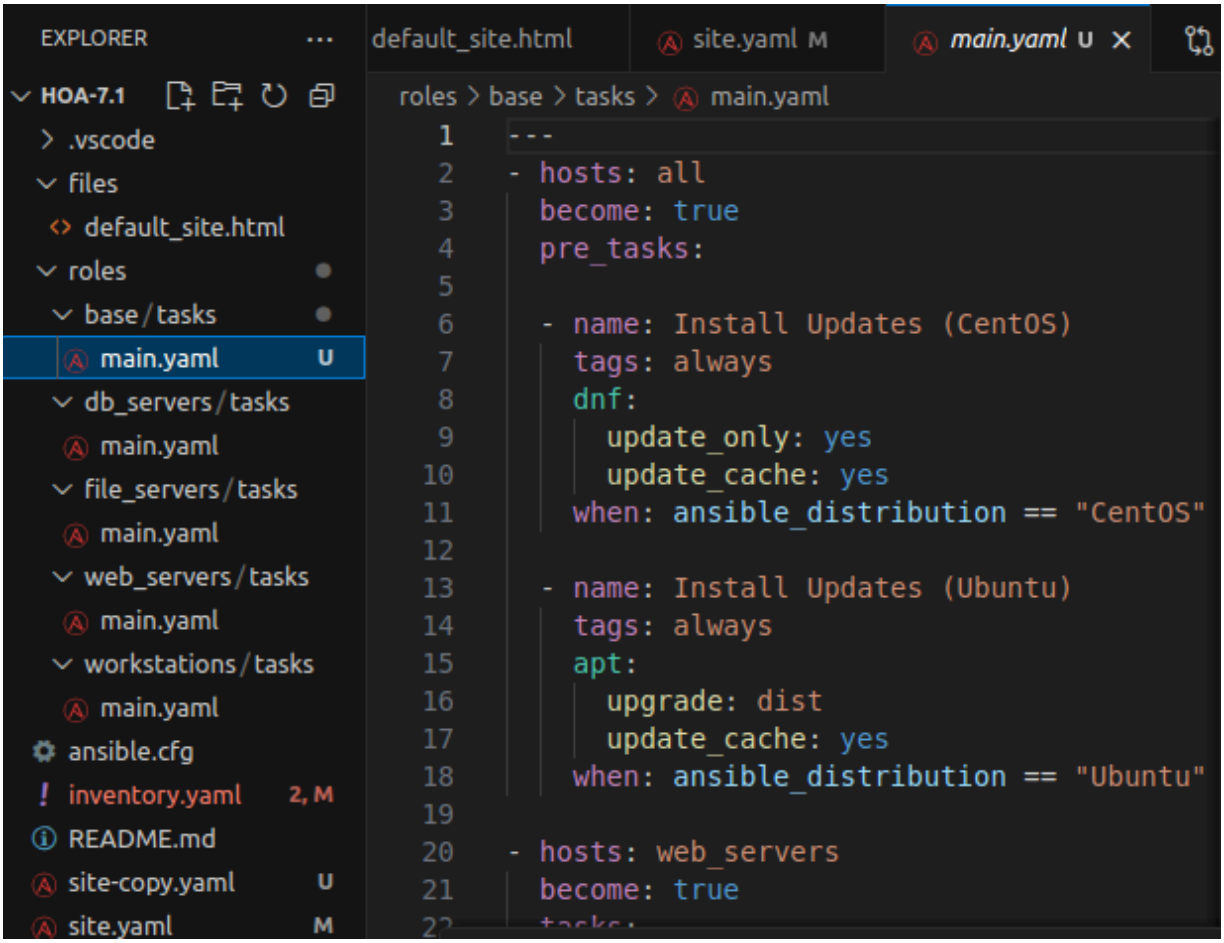
site-copy.yaml

```
22  - hosts: all
23    become: true
24    roles:
25      - base
26
27  - hosts: workstations
28    become: true
29    roles:
30      - workstations
31
32  - hosts: web_servers
33    become: true
34    roles:
35      - web_servers
36
37  - hosts: db_servers
38    become: true
39    roles:
40      - db_servers
41
42  - hosts: file_servers
43    become: true
44    roles:
45      - file_servers
```

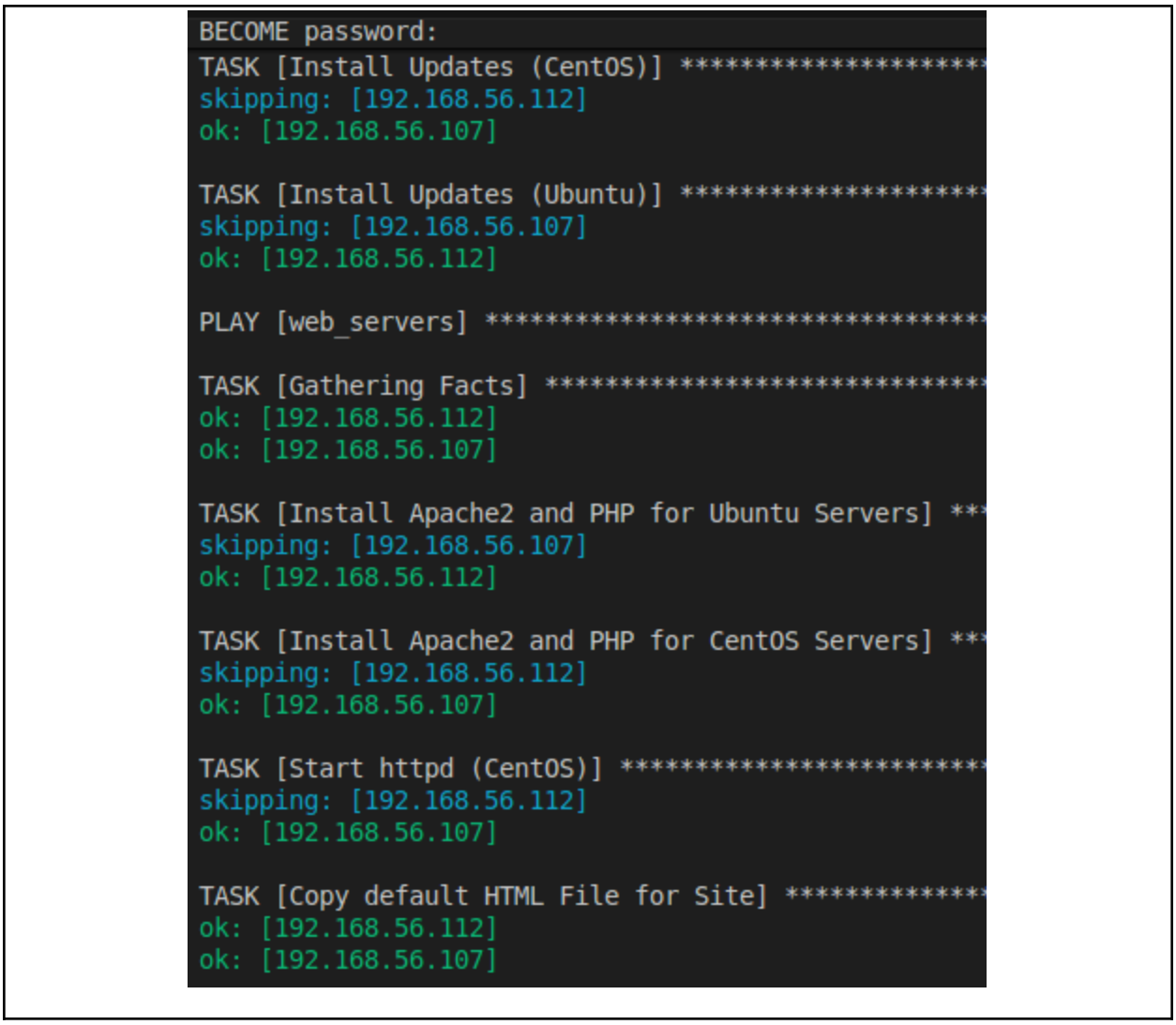
2. Under the same directory, create a new directory and name it roles. Enter the roles directory and create new directories: base, web_servers, file_servers, db_servers and workstations. For each directory, create a directory and name it tasks.

```
roallos-ubuntu@workstation:~/HOA-7.1$ ls
ansible.cfg  files  inventory.yaml  README.md  site-copy.yaml  site.yaml
roallos-ubuntu@workstation:~/HOA-7.1$ mkdir -p roles/{base,web_servers,file_servers,db
_servers,workstations}/tasks
roallos-ubuntu@workstation:~/HOA-7.1$ ls -lah roles
total 28K
drwxrwxr-x 7 roallos-ubuntu roallos-ubuntu 4.0K Oct  3 18:14 .
drwxrwxr-x 6 roallos-ubuntu roallos-ubuntu 4.0K Oct  3 18:14 ..
drwxrwxr-x 3 roallos-ubuntu roallos-ubuntu 4.0K Oct  3 18:14 base
drwxrwxr-x 3 roallos-ubuntu roallos-ubuntu 4.0K Oct  3 18:14 db_servers
drwxrwxr-x 3 roallos-ubuntu roallos-ubuntu 4.0K Oct  3 18:14 file_servers
drwxrwxr-x 3 roallos-ubuntu roallos-ubuntu 4.0K Oct  3 18:14 web_servers
drwxrwxr-x 3 roallos-ubuntu roallos-ubuntu 4.0K Oct  3 18:14 workstations
roallos-ubuntu@workstation:~/HOA-7.1$ ls roles/db_servers/
tasks
roallos-ubuntu@workstation:~/HOA-7.1$
```

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.



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


BECOME password:
TASK [Install Updates (CentOS)] ********************
skipping: [192.168.56.112]
ok: [192.168.56.107]
TASK [Install Updates (Ubuntu)] ********************
skipping: [192.168.56.107]
ok: [192.168.56.112]
PLAY [web_servers] *********************************
TASK [Gathering Facts] *****************************
ok: [192.168.56.112]
ok: [192.168.56.107]
TASK [Install Apache2 and PHP for Ubuntu Servers] ***
skipping: [192.168.56.107]
ok: [192.168.56.112]
TASK [Install Apache2 and PHP for CentOS Servers] ***
skipping: [192.168.56.112]
ok: [192.168.56.107]
TASK [Start httpd (CentOS)] ************************
skipping: [192.168.56.112]
ok: [192.168.56.107]
TASK [Copy default HTML File for Site] *************
ok: [192.168.56.112]
ok: [192.168.56.107]

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

TASK [Gathering Facts] ********************
ok: [192.168.56.112]

TASK [Install Unzip] **********************
ok: [192.168.56.112]

TASK [Install Terraform] ******************
ok: [192.168.56.112]

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

TASK [Gathering Facts] ********************
ok: [192.168.56.112]

TASK [Install MariaDB Package (CentOS)] *
skipping: [192.168.56.112]

TASK [MariaDB Restarting/Enabling] ******
changed: [192.168.56.112]

TASK [Install MariaDB Package (Ubuntu)] *
ok: [192.168.56.112]

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

TASK [Gathering Facts] ********************
ok: [192.168.56.107]
```

```
TASK [Install Samba Package] ***************************************************
ok: [192.168.56.107]

PLAY RECAP *********************************************************************
192.168.56.106             : ok=0    changed=0    unreachable=1    failed=0    skipped
=0    rescued=0    ignored=0
192.168.56.107             : ok=8    changed=0    unreachable=0    failed=0    skipped
=2    rescued=0    ignored=0
192.168.56.112             : ok=11   changed=1    unreachable=0    failed=0    skipped
=4    rescued=0    ignored=0

roallos-ubuntu@workstation:~/HOA-7.1$
```

192.168.56.106 is an Ubuntu server that is turned off.

Reflections:
Answer the following:

1. What is the importance of creating roles?

   The creation of roles for Ansible implements modularity for the system management of multiple servers and server groups from the inventory. Groups within the roles can be given or assigned a specific process.

2. What is the importance of managing files?

   As the implementation of roles for Ansible increases requirement for file management because of the increased count of files. This segments the parts of the desired processes for each of the server groups.