Name: Jose Mari T. Dela Peña	Date Performed: 10/07/2024
Course/Section: CPE31S2	Date Submitted: 10/09/2024
Instructor: Engr. Robin Valenzuela	Semester and SY: 1st Sem 2024-2025

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
```

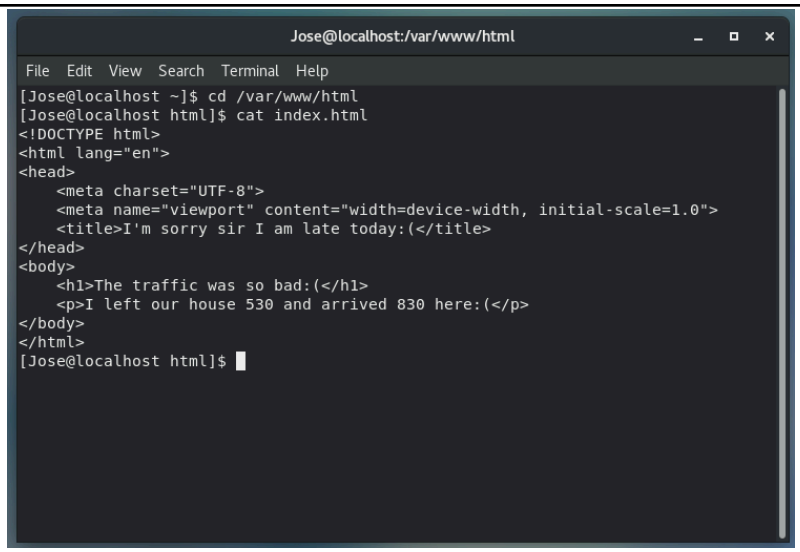
3. Run the playbook *site.yml*. Describe the changes.

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

TASK [Gathering Facts] ********************************************************
ok: [192.168.56.100]
ok: [192.168.56.103]

TASK [install apache and php for Ubuntu servers] ******************************
skipping: [192.168.56.103]
ok: [192.168.56.100]

TASK [install apache and php for CentOS servers] ******************************
skipping: [192.168.56.100]
ok: [192.168.56.103]

TASK [start httpd (CentOS)] ***************************************************
skipping: [192.168.56.100]
ok: [192.168.56.103]

TASK [copy default html file for site] ****************************************
changed: [192.168.56.103]
changed: [192.168.56.100]
```

Explanation: It successfully replicated the html on my web servers specifically on the index.html.

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.

```
jose@server1:/var/www/html$ cat index.html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>I'm sorry sir I am late today:(</title>
</head>
<body>
    <h1>The traffic was so bad:(</h1>
    <p>I left our house 530 and arrived 830 here:(</p>
</body>
</html>
jose@server1:/var/www/html$
```



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

```
jose@workstation:~/CPE212ACT7.1$ git add files
jose@workstation:~/CPE212ACT7.1$ git add site.yml
jose@workstation:~/CPE212ACT7.1$ git add inventory
jose@workstation:~/CPE212ACT7.1$ git add ansible.cfg
jose@workstation:~/CPE212ACT7.1$ git commit -m "please no more traffic"
[main 136c2c3] please no more traffic
 4 files changed, 117 insertions(+)
 create mode 100644 ansible.cfg
 create mode 100644 files/default_site.html
 create mode 100644 inventory
 create mode 100644 site.yml
jose@workstation:~/CPE212ACT7.1$ git push origin main
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 2 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (7/7), 1.31 KiB | 1.31 MiB/s, done.
Total 7 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:Liglig14/CPE212ACT7.1.git
   16c16d1..136c2c3  main -> main
jose@workstation:~/CPE212ACT7.1$
```

Explanation: After I executed the site.yml and the html was replicated in web servers, I was able to push my current files onto my Github Repository.

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.
3. Run the playbook. Describe the output.

```
PLAY [192.168.56.100] *********************************************************************

TASK [Gathering Facts] *********************************************************************
ok: [192.168.56.100]

TASK [install unzip] ***********************************************************************
ok: [192.168.56.100]

TASK [install terraform] *******************************************************************
changed: [192.168.56.100]
```

Explanation: It successfully installed unzip and terraform on my chosen server.

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

```
jose@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.

Common commands:
    apply              Builds or changes infrastructure
    console            Interactive console for Terraform interpolations
    destroy            Destroy Terraform-managed infrastructure
    env                Workspace management
    fmt                Rewrites config files to canonical format
    get                Download and install modules for the configuration
    graph              Create a visual graph of Terraform resources
    import             Import existing infrastructure into Terraform
    init               Initialize a Terraform working directory
    login              Obtain and save credentials for a remote host
    logout             Remove locally-stored credentials for a remote host
```

Explanation: It is now verified that terraform is installed as it shows the guide commands available with terraform.

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)

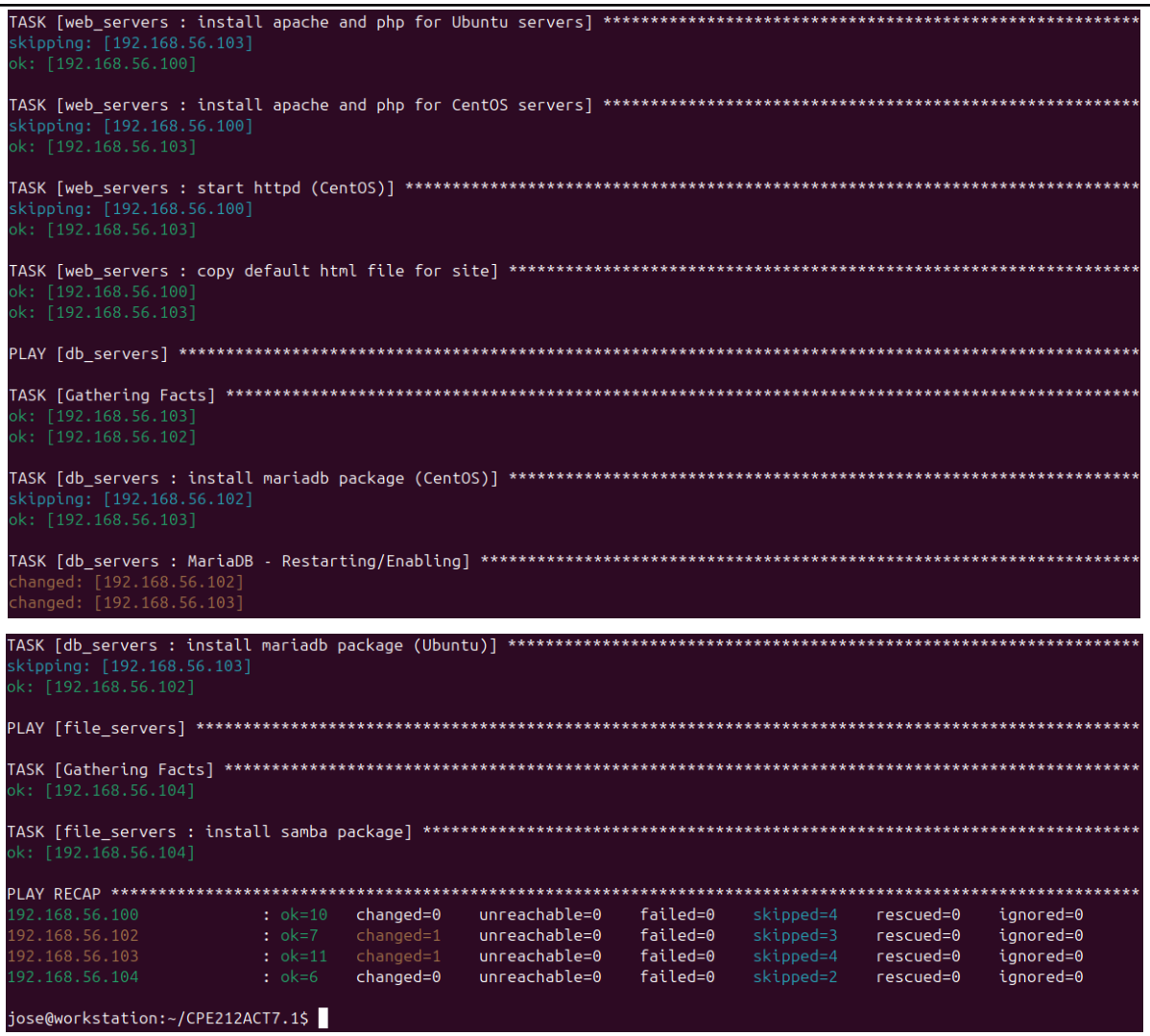
```
---
- hosts: all
  become: true
  pre_tasks:

  - 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.

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.
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.

```
jose@workstation:~/CPE212ACT7.1$ ansible-playbook --ask-become-pass site.yml
BECOME password:

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

TASK [Gathering Facts] *********************************************************************************************
ok: [192.168.56.100]
ok: [192.168.56.103]
ok: [192.168.56.102]
ok: [192.168.56.104]

TASK [update repository index (CentOS)] ****************************************************************************
skipping: [192.168.56.100]
skipping: [192.168.56.102]
skipping: [192.168.56.104]
ok: [192.168.56.103]

TASK [update repository index (Ubuntu)] ****************************************************************************
skipping: [192.168.56.103]
ok: [192.168.56.104]
ok: [192.168.56.100]
ok: [192.168.56.102]

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

TASK [Gathering Facts] *********************************************************************************************
ok: [192.168.56.100]
ok: [192.168.56.103]
ok: [192.168.56.104]
ok: [192.168.56.102]
```

```
TASK [base : install updates (CentOS)] *****************************************************************************
skipping: [192.168.56.100]
skipping: [192.168.56.104]
skipping: [192.168.56.102]
ok: [192.168.56.103]

TASK [base : install updates (Ubuntu)] *****************************************************************************
skipping: [192.168.56.103]
ok: [192.168.56.100]
ok: [192.168.56.104]
ok: [192.168.56.102]

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

TASK [Gathering Facts] *********************************************************************************************
ok: [192.168.56.100]

TASK [workstations : install unzip] ********************************************************************************
ok: [192.168.56.100]

TASK [workstations : install terraform] ****************************************************************************
ok: [192.168.56.100]

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

TASK [Gathering Facts] *********************************************************************************************
ok: [192.168.56.100]
ok: [192.168.56.103]

TASK [web_servers : install apache and php for Ubuntu servers] *****************************************************
skipping: [192.168.56.103]
ok: [192.168.56.100]
```



Explanation: After successfully creating roles and tasks directories I added only the specific task in each of the main.yml. When I executed the site.yml it executed the specific tasks in each main.yml in each tasks directories.

Reflections:

Answer the following:

1. What is the importance of creating roles?

- The importance of creating roles is that tasks are not jam packed in a single playbook. Each role is separated from one another, it promotes user-friendliness, effectiveness, and clarity.

2. What is the importance of managing files?

- The importance of managing files is that it enhances the user experience by not having difficulty finding files that are necessary in common operations. If we do not manage our files properly then we are halfway failed with our tasks since it will consume time and show irresponsibility.

Github Repository Link: https://github.com/Liglig14/CPE212ACT7.1