Name: Hermitano, Johnny C.	**Date Performed: 10/13/2022**
Course/Section: CPE31S23	**Date Submitted: 10/13/2022**
Instructor: Engr. Jonathan Taylar	**Semester and SY: 1st sem/sy:2022-2023**

Activity 7: Managing Files and Creating Roles in Ansible

1. **Objectives:**

1.1 Manage files in remote servers

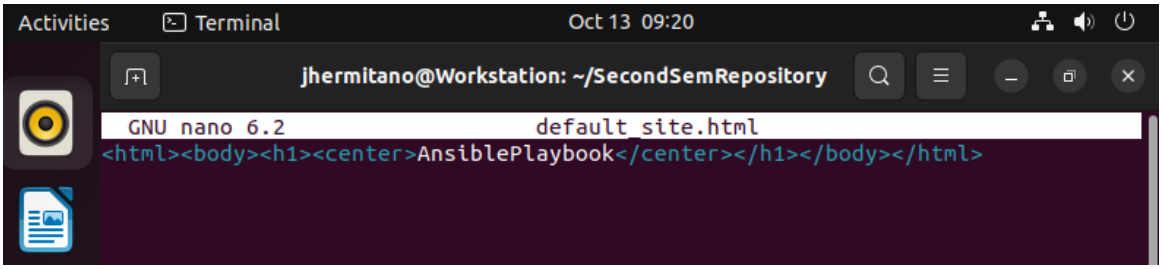
1.2 Implement roles in ansible

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.



In this picture, shows what inside my default_site.html for this activity.

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
```

```
    service:
      name: httpd
      state: started
    when: ansible_distribution == "CentOS"

  - name: copy defualt html for file
    tags: apache,apache2,httpd
    copy:
      src: default_site.html
      dest: /var/www/html/index.html
      owner: root
      group: root
      mode: 0644

- hosts: db_servers
  become: true
  tasks:
```

In this part, I inserted the said command on my playbook between the db_servers and web_servers.

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

```
jhermitano@Workstation:~/SecondSemRepository$ ansible-playbook --ask-become-pas
s site.yml
BECOME password:

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

TASK [Gathering Facts] *********************************************************
*
ok: [192.168.56.106]
ok: [192.168.56.115]

TASK [install updates (CentOS)] ************************************************
*
skipping: [192.168.56.106]
ok: [192.168.56.115]

TASK [install updates (Ubuntu)] ************************************************
*
skipping: [192.168.56.115]
ok: [192.168.56.106]

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

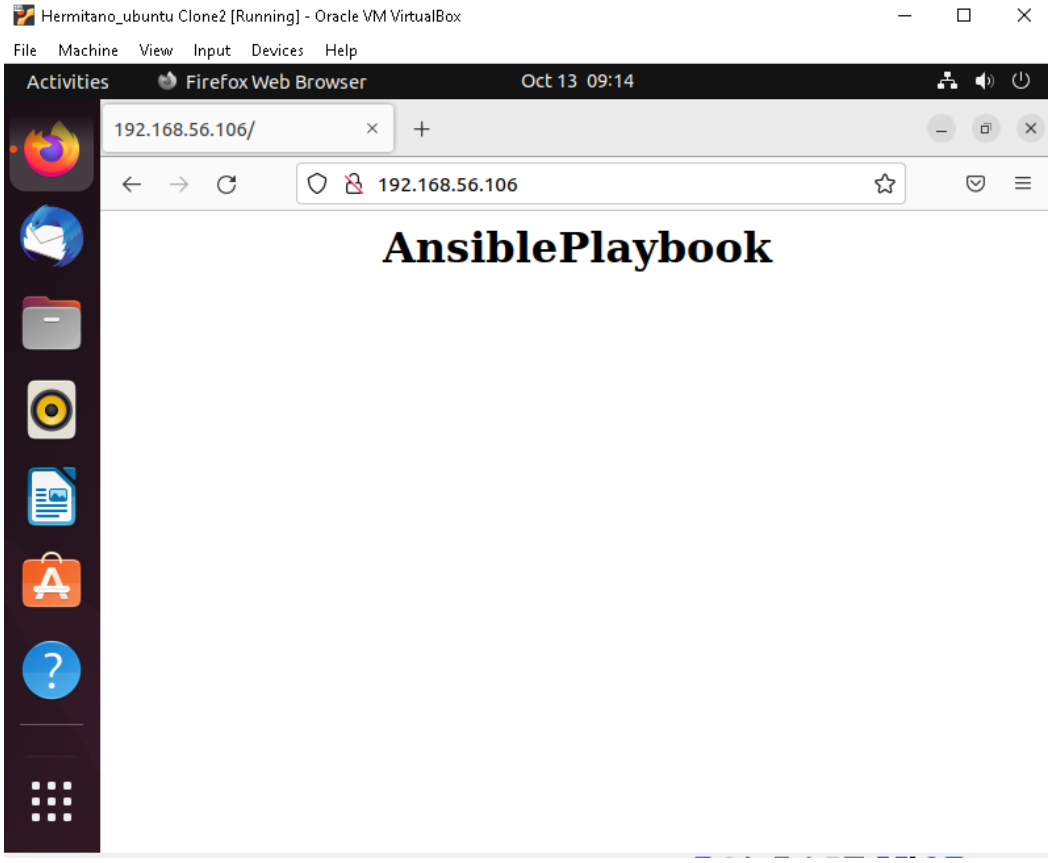
```
TASK [Gathering Facts] *********************************************************
*
ok: [192.168.56.115]
ok: [192.168.56.106]

TASK [install apache and php for Ubuntu Servers] *******************************
*
skipping: [192.168.56.115]
ok: [192.168.56.106]

TASK [install apache and php for CentOS Servers] *******************************
*
skipping: [192.168.56.106]
ok: [192.168.56.115]

TASK [start httpd (CentOS)] ****************************************************
*
skipping: [192.168.56.106]
ok: [192.168.56.115]

TASK [copy defualt html for file] **********************************************
*
changed: [192.168.56.106]
changed: [192.168.56.115]
```

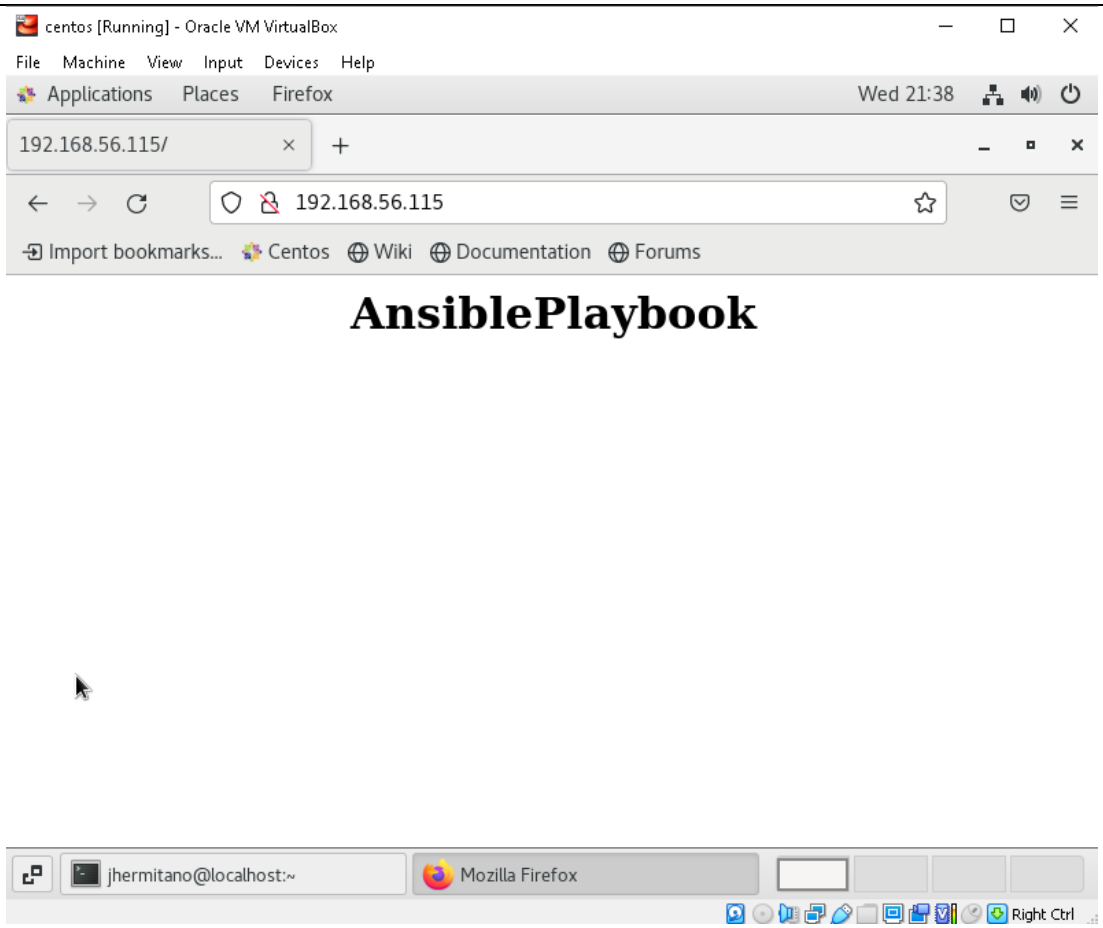
```
PLAY [db_servers] **************************************************************
*

TASK [Gathering Facts] *********************************************************
*
ok: [192.168.56.115]

TASK [install mariadb package (CentOS)] ****************************************
*
ok: [192.168.56.115]

TASK [Mariadb- Restarting/Enabling] ********************************************
*
changed: [192.168.56.115]

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

TASK [Gathering Facts] *********************************************************
*
ok: [192.168.56.106]

TASK [install samba package] ***************************************************
*
ok: [192.168.56.106]
```

```
PLAY RECAP *********************************************************************
*
192.168.56.106             : ok=7    changed=0    unreachable=0    failed=0
skipped=3    rescued=0    ignored=0
192.168.56.115             : ok=9    changed=1    unreachable=0    failed=0
skipped=2    rescued=0    ignored=0
```

The screenshots show the result of playing 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.





The picture above shows the virtual result or the actual changes on the servers from playing the playbook.

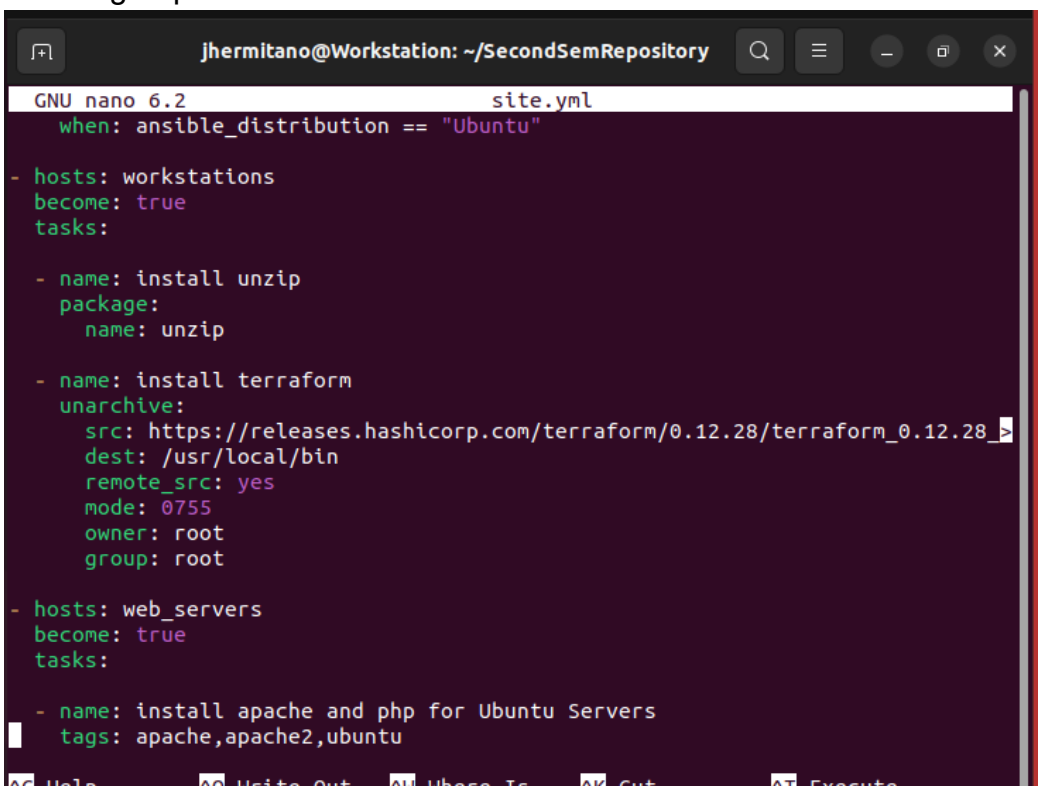
5. Sync your local repository with GitHub and describe the changes.

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



The screenshot shows the edited part of the playbook where I inserted the said command.

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

```
jhermitano@Workstation: ~/SecondSemReposit
  GNU nano 6.2                                  inventory
[web_servers]
192.168.56.115
192.168.56.106

[workstations]
192.168.56.106

[db_servers]
192.168.56.115

[file_servers]
192.168.56.106
```

The screenshot shows the inventory I am using for this activity.

3. Run the playbook. Describe the output.

```
jhermitano@Workstation:~/SecondSemRepository$ sudo nano inventory
jhermitano@Workstation:~/SecondSemRepository$ ansible-playbook --ask-become-pas
s site.yml
BECOME password:

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

TASK [Gathering Facts] *********************************************************
*
ok: [192.168.56.106]
ok: [192.168.56.115]

TASK [install updates (CentOS)] ************************************************
*
skipping: [192.168.56.106]
ok: [192.168.56.115]

TASK [install updates (Ubuntu)] ************************************************
*
skipping: [192.168.56.115]
ok: [192.168.56.106]

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

```
TASK [Gathering Facts] ***********************************************************
*
ok: [192.168.56.106]

TASK [install unzip] *************************************************************
*
ok: [192.168.56.106]

TASK [install terraform] *********************************************************
*
changed: [192.168.56.106]

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

TASK [Gathering Facts] ***********************************************************
*
ok: [192.168.56.106]
ok: [192.168.56.115]

TASK [install apache and php for Ubuntu Servers] *********************************
*
skipping: [192.168.56.115]
ok: [192.168.56.106]

TASK [install apache and php for CentOS Servers] *********************************
*
skipping: [192.168.56.106]
ok: [192.168.56.115]

TASK [start httpd (CentOS)] ******************************************************
*
skipping: [192.168.56.106]
ok: [192.168.56.115]

TASK [copy defualt html for file] ************************************************
*
ok: [192.168.56.106]
changed: [192.168.56.115]

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

TASK [Gathering Facts] ***********************************************************
*
ok: [192.168.56.115]

TASK [install mariadb package (CentOS)] ******************************************
*
ok: [192.168.56.115]

TASK [Mariadb- Restarting/Enabling] **********************************************
*
changed: [192.168.56.115]

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

```
TASK [Gathering Facts] *********************************************************
*
ok: [192.168.56.106]

TASK [install samba package] ***************************************************
*
ok: [192.168.56.106]

PLAY RECAP *********************************************************************
*
192.168.56.106             : ok=10   changed=1    unreachable=0    failed=0
skipped=3    rescued=0    ignored=0
192.168.56.115             : ok=9    changed=2    unreachable=0    failed=0
skipped=2    rescued=0    ignored=0

jhermitano@Workstation:~/SecondSemRepository$
```

The screenshots show the result of playing the playbook.

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

```
jhermitano@Server2:~$ terraform -version
Terraform v0.12.28

Your version of Terraform is out of date! The latest version
is 1.3.2. You can update by downloading from https://www.terraform.io/downloads
.html
```

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)

```
jhermitano@Workstation:~/SecondSemRepository$ cat site.yml
---

- hosts: all
  become: true
  pre_tasks:

  - name: update repository index (CentOS)
    tags: always
    dnf:
      update_cache: yes
      change_when: false
    when: ansible_distribution == "CentOS"

  - name: update repository index (Ubuntu)
    tags: always
    apt:
      update_cache: yes
      change_when: false
    when: ansible_distribution == "Ubuntu"

- hosts: all
  become: true
  roles:
    - base

- hosts: all
  become: true
  roles:
    - workstation

- hosts: all
  become: true
  roles:
    - web_servers

- hosts: all
  become: true
  roles:
    - db_servers

- hosts: all
  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,

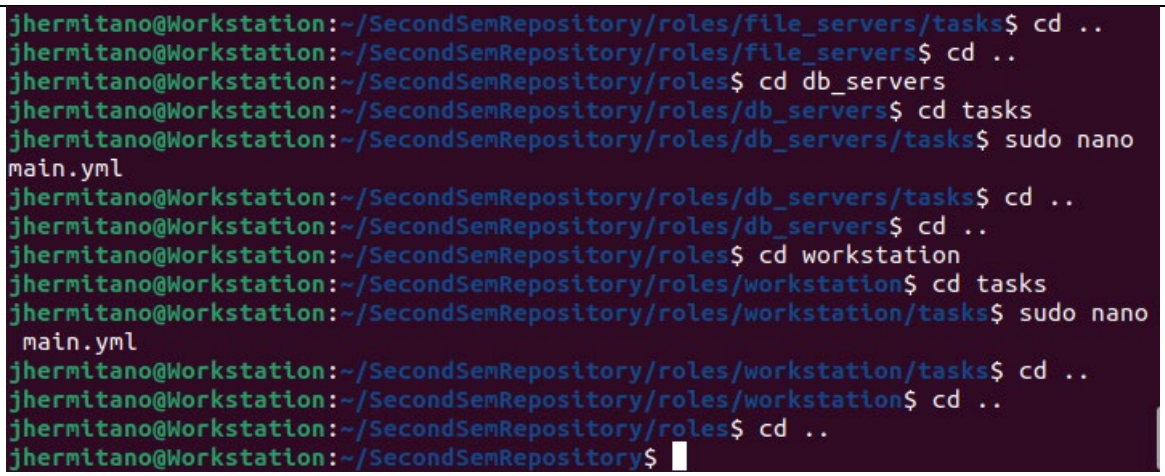
db_servers and workstations. For each directory, create a directory and name it tasks.

```
jhermitano@Workstation:~/SecondSemRepository$ mkdir roles
jhermitano@Workstation:~/SecondSemRepository$ cd roles
jhermitano@Workstation:~/SecondSemRepository/roles$ mkdir base
jhermitano@Workstation:~/SecondSemRepository/roles$ mkdir web_servers
jhermitano@Workstation:~/SecondSemRepository/roles$ mkdir file_servers
jhermitano@Workstation:~/SecondSemRepository/roles$ mkdir db_servers
jhermitano@Workstation:~/SecondSemRepository/roles$ mkdir workstation
jhermitano@Workstation:~/SecondSemRepository/roles$ ls
base  db_servers  file_servers  web_servers  workstation
```

```
jhermitano@Workstation:~/SecondSemRepository/roles$ cd base
jhermitano@Workstation:~/SecondSemRepository/roles/base$ mkdir tasks
jhermitano@Workstation:~/SecondSemRepository/roles/base$ cd ..
jhermitano@Workstation:~/SecondSemRepository/roles$ cd web_servers
jhermitano@Workstation:~/SecondSemRepository/roles/web_servers$ mkdir tasks
jhermitano@Workstation:~/SecondSemRepository/roles/web_servers$ cd ..
jhermitano@Workstation:~/SecondSemRepository/roles$ cd file_servers
jhermitano@Workstation:~/SecondSemRepository/roles/file_servers$ mkdir tasks
jhermitano@Workstation:~/SecondSemRepository/roles/file_servers$ cd ..
jhermitano@Workstation:~/SecondSemRepository/roles$ cd db_servers
jhermitano@Workstation:~/SecondSemRepository/roles/db_servers$ mkdir tasks
jhermitano@Workstation:~/SecondSemRepository/roles/db_servers$ cd ..
jhermitano@Workstation:~/SecondSemRepository/roles$ cd workstation
jhermitano@Workstation:~/SecondSemRepository/roles/workstation$ mkdir tasks
jhermitano@Workstation:~/SecondSemRepository/roles/workstation$ cd ..
jhermitano@Workstation:~/SecondSemRepository/roles$
```

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.

```
jhermitano@Workstation:~/SecondSemRepository$ cd roles
jhermitano@Workstation:~/SecondSemRepository/roles$ cd base
jhermitano@Workstation:~/SecondSemRepository/roles/base$ cd tasks
jhermitano@Workstation:~/SecondSemRepository/roles/base/tasks$ sudo nano main.y
ml
jhermitano@Workstation:~/SecondSemRepository/roles/base/tasks$ cd ..
jhermitano@Workstation:~/SecondSemRepository/roles/base$ cd ..
jhermitano@Workstation:~/SecondSemRepository/roles$ cd web_servers
jhermitano@Workstation:~/SecondSemRepository/roles/web_servers$ cd tasks
jhermitano@Workstation:~/SecondSemRepository/roles/web_servers/tasks$ sudo nano
 main.yml
jhermitano@Workstation:~/SecondSemRepository/roles/web_servers/tasks$ cd ..
jhermitano@Workstation:~/SecondSemRepository/roles/web_servers$ cd ..
jhermitano@Workstation:~/SecondSemRepository/roles$ cd file_servers
jhermitano@Workstation:~/SecondSemRepository/roles/file_servers$ cd tasks
jhermitano@Workstation:~/SecondSemRepository/roles/file_servers/tasks$ sudo nan
o main.yml
```



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

Reflections:

Answer the following:

1. What is the importance of creating roles?

   For completely independent or interdependent collections of variables, tasks, files, templates, and modules, roles offer a framework. The role is the main tool in Ansible for splitting a playbook into different files.

2. What is the importance of managing files?

   File organization strategies are crucial since they can keep your computer structured and make it simpler to identify and access the files you require.