Name: Espiritu, Diego Angelo G.	Date Performed: 10/02/2023
Course/Section: CPE232/CPE31S6	Date Submitted: 10/02/2023
Instructor: Dr. Jonathan Vidal Taylar	Semester and SY: 1st sem 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

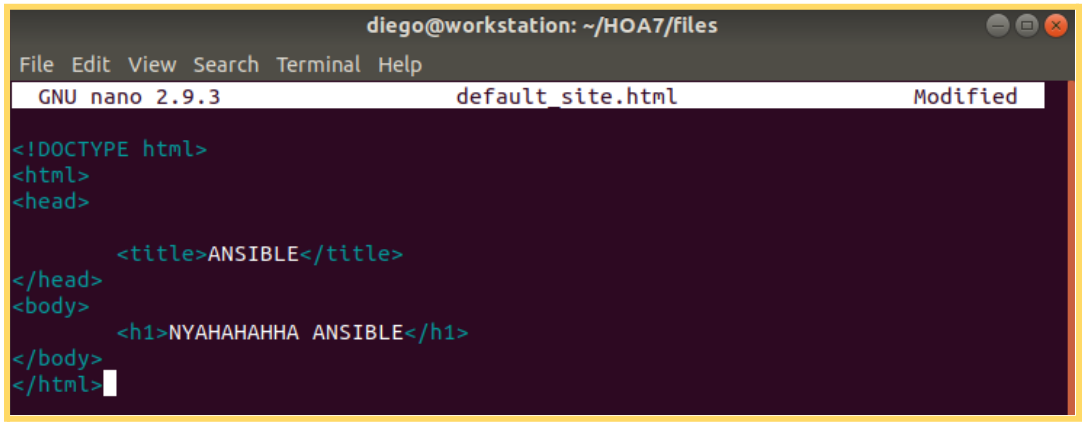
2. **Discussion**:

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.

```
diego@workstation:~$ cd HOA7
diego@workstation:~/HOA7$ nano site.yml
diego@workstation:~/HOA7$ ls
ansible.cfg  files  inventory  README.md  site.yml
diego@workstation:~/HOA7$ cd files
diego@workstation:~/HOA7/files$ nano default_site.html
diego@workstation:~/HOA7/files$ ls
default_site.html
diego@workstation:~/HOA7/files$
```



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

```
- hosts: web_server
  become: true
  tasks:

  - name: copy default html file for site
    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.
   -The changes that I observed is that the copy default html file for the site is successful for server 2 and CentOS. That's why there are changes.

```
TASK [copy default html file for site] ************************************
*
ok: [192.168.56.102]
ok: [192.168.56.107]
```

```
PLAY RECAP ********************************************************************
*
192.168.56.102             : ok=7    changed=0    unreachable=0    failed=0
skipped=3    rescued=0    ignored=0
192.168.56.103             : ok=5    changed=1    unreachable=0    failed=0
skipped=2    rescued=0    ignored=0
192.168.56.107             : ok=9    changed=1    unreachable=0    failed=0
skipped=3    rescued=0    ignored=0

diego@workstation:~/HOA7$
```

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.

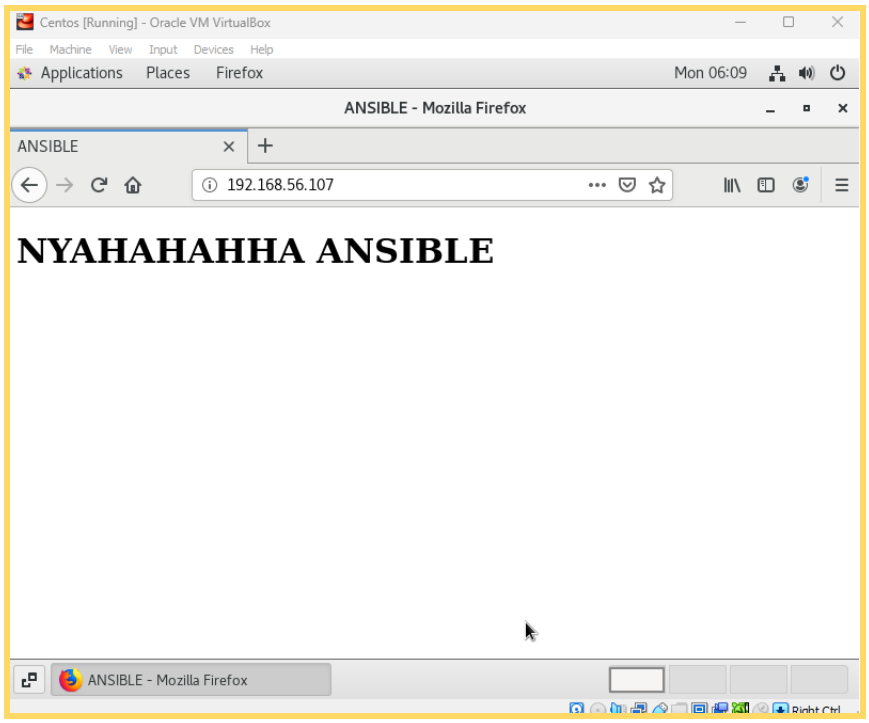
-To check if the saved default site HTML is working, you need to access each server and change the directory to var/www/html. Then, you can use the cat command to see if it's working. In CentOS, you can browse to Mozilla and enter the IP address of the server to see the default server HTML. The words that you put into the file htm will be printed."

Server 1

```
diego@server1:~$ cd /var/www/html
diego@server1:/var/www/html$ ls
index.html
diego@server1:/var/www/html$ cat index.html
<!DOCTYPE html>
<html>
<head>

        <title>ANSIBLE</title>
</head>
<body>
        <h1>NYAHAHAHHA ANSIBLE</h1>
</body>
</html>
diego@server1:/var/www/html$
```

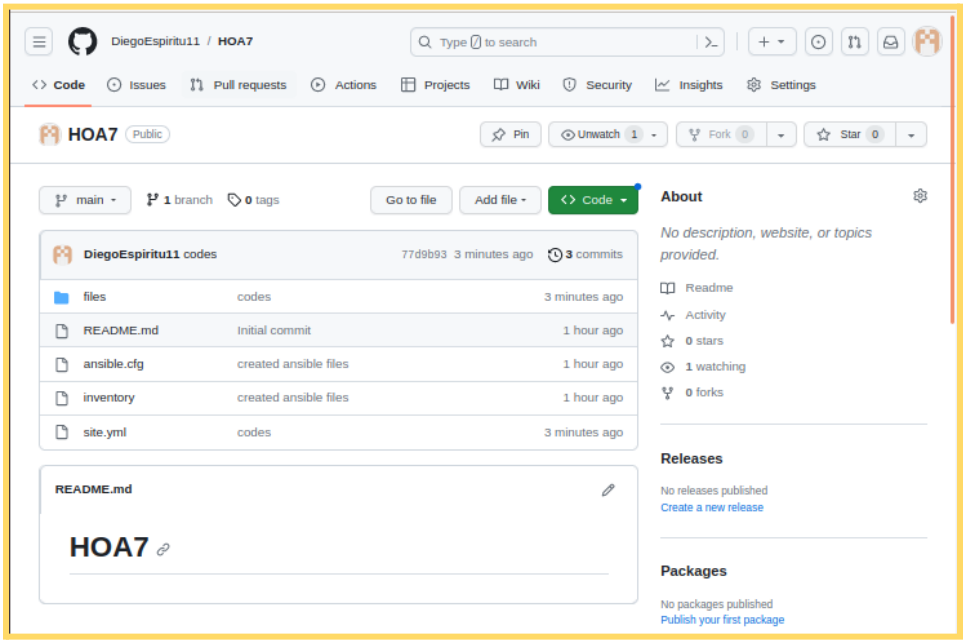
CentOS



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

```
diego@workstation:~/HOA7$ git add files
diego@workstation:~/HOA7$ git add site.yml
diego@workstation:~/HOA7$ git commit -m "codes"
[main 77d9b93] codes
 2 files changed, 102 insertions(+)
 create mode 100644 files/default_site.html
 create mode 100644 site.yml
diego@workstation:~/HOA7$ git push origin main
Username for 'https://github.com': DiegoEspiritu11
Password for 'https://DiegoEspiritu11@github.com':
Counting objects: 9, done.
Delta compression using up to 2 threads.
Compressing objects: 100% (8/8), done.
Writing objects: 100% (9/9), 1.36 KiB | 1.36 MiB/s, done.
Total 9 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/DiegoEspiritu11/HOA7.git
   5480f11..77d9b93  main -> main
diego@workstation:~/HOA7$ cd files
diego@workstation:~/HOA7/files$ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
diego@workstation:~/HOA7/files$
```



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

```
- 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_0.12.2$
        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.

```
  GNU nano 2.9.3                          inventory

[web_server]
192.168.56.102
192.168.56.107

[db_server]
192.168.56.103
192.168.56.107

[file_server]
192.168.56.102

[workstations]
192.168.56.103
```

3. Run the playbook. Describe the output.
-I installed Terraform on the server (server2) that I specified in the inventory, which included the servers in the workstations group.

```
TASK [install unzip] ***********************************************************
*
ok: [192.168.56.103]

TASK [install terraform] *******************************************************
*
changed: [192.168.56.103]

PLAY [web_server] **************************************************************
*
```

```
PLAY RECAP *********************************************************************
*
192.168.56.102             : ok=7    changed=0    unreachable=0    failed=0
skipped=3    rescued=0    ignored=0
192.168.56.103             : ok=8    changed=2    unreachable=0    failed=0
skipped=2    rescued=0    ignored=0
192.168.56.107             : ok=9    changed=1    unreachable=0    failed=0
skipped=3    rescued=0    ignored=0

diego@workstation:~/HOA7$
```

4. On the Ubuntu remote workstation, type terraform to verify installation of terraform. Describe the output.
-I used ssh and went to server2 and typed the command (terraform –version) to check if it was installed.

```
diego@workstation:~$ ssh diego@server2
Welcome to Ubuntu 18.04.6 LTS (GNU/Linux 5.4.0-150-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

Expanded Security Maintenance for Infrastructure is not enabled.

0 updates can be applied immediately.

108 additional security updates can be applied with ESM Infra.
Learn more about enabling ESM Infra service for Ubuntu 18.04 at
https://ubuntu.com/18-04

New release '20.04.6 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Your Hardware Enablement Stack (HWE) is supported until April 2023.
Last login: Mon Oct  2 18:31:58 2023 from 192.168.56.101
diego@server2:~$ terraform --version
Terraform v0.12.28

Your version of Terraform is out of date! The latest version
is 1.5.7. You can update by downloading from https://www.terraform.io/downloads
.html
diego@server2:~$
```

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.

```
 GNU nano 2.9.3                          site.yml

---

- 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_server
  become: true
  roles:
    - web_server
```

```
- hosts: db_server
  become: true
  roles:
    - db_server

- hosts: file_server
  become: true
  roles:
    - file_server
```

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.

```
diego@workstation:~/HOA7$ mkdir roles
diego@workstation:~/HOA7$ cd roles
diego@workstation:~/HOA7/roles$ mkdir base
diego@workstation:~/HOA7/roles$ mkdir web_server
diego@workstation:~/HOA7/roles$ mkdir file_server
diego@workstation:~/HOA7/roles$ mkdir db_server
diego@workstation:~/HOA7/roles$ mkdir workstations
diego@workstation:~/HOA7/roles$
```

```
diego@workstation:~/HOA7/roles$ cd base
diego@workstation:~/HOA7/roles/base$ mkdir tasks
diego@workstation:~/HOA7/roles/base$ cd -
/home/diego/HOA7/roles
diego@workstation:~/HOA7/roles$ cd web_server
diego@workstation:~/HOA7/roles/web_server$ mkdir tasks
diego@workstation:~/HOA7/roles/web_server$ cd -
/home/diego/HOA7/roles
diego@workstation:~/HOA7/roles$ cd file_server
diego@workstation:~/HOA7/roles/file_server$ mkdir tasks
diego@workstation:~/HOA7/roles/file_server$ cd -
/home/diego/HOA7/roles
diego@workstation:~/HOA7/roles$ cd db_server
diego@workstation:~/HOA7/roles/db_server$ mkdir tasks
diego@workstation:~/HOA7/roles/db_server$ cd -
/home/diego/HOA7/roles
diego@workstation:~/HOA7/roles$ cd workstations
diego@workstation:~/HOA7/roles/workstations$ mkdir tasks
diego@workstation:~/HOA7/roles/workstations$
```

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.

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

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

TASK [Gathering Facts] *********************************************************
*
ok: [192.168.56.107]
ok: [192.168.56.103]
ok: [192.168.56.102]

TASK [update repository index (CentOS)] ****************************************
*
skipping: [192.168.56.102]
skipping: [192.168.56.103]
ok: [192.168.56.107]

TASK [install updates (Ubuntu)] ************************************************
*
skipping: [192.168.56.107]
ok: [192.168.56.102]
ok: [192.168.56.103]

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

TASK [Gathering Facts] *********************************************************
*
```

```
TASK [Gathering Facts] *********************************************************
*
ok: [192.168.56.102]
ok: [192.168.56.103]
ok: [192.168.56.107]

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

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

TASK [workstations : install unzip] ********************************************
*
ok: [192.168.56.103]

TASK [workstations : install terraform] ****************************************
*
ok: [192.168.56.103]

PLAY [web_server] **************************************************************
*

TASK [Gathering Facts] *********************************************************
*
ok: [192.168.56.107]
ok: [192.168.56.102]
```

```
TASK [Gathering Facts] *********************************************************
*
ok: [192.168.56.107]
ok: [192.168.56.102]

TASK [web_server : copy default html file for site] ****************************
*
ok: [192.168.56.102]
ok: [192.168.56.107]

TASK [web_server : install apache and php for Ubuntu servers] ******************
*
skipping: [192.168.56.107]
ok: [192.168.56.102]

TASK [web_server : install apache and php for CentOS servers] ******************
*
skipping: [192.168.56.102]
ok: [192.168.56.107]

TASK [web_server : start httpd (CentOS)] ***************************************
*
skipping: [192.168.56.102]
ok: [192.168.56.107]

PLAY [db_server] ***************************************************************
*

TASK [Gathering Facts] *********************************************************
```

```
TASK [Gathering Facts] *********************************************************
*
ok: [192.168.56.103]
ok: [192.168.56.107]

TASK [db_server : install mariadb package (CentOS)] ***************************
*
skipping: [192.168.56.103]
ok: [192.168.56.107]

TASK [db_server : install mariadb package (Ubuntu)] ***************************
*
skipping: [192.168.56.107]
ok: [192.168.56.103]

TASK [db_server : Mariadb- Restarting/Enabling] ********************************
*
changed: [192.168.56.103]
changed: [192.168.56.107]

PLAY [file_server] *************************************************************
*

TASK [Gathering Facts] *********************************************************
*
ok: [192.168.56.102]

TASK [file_server : install samba package] *************************************
```

```
TASK [file_server : install samba package] *************************************
*
ok: [192.168.56.102]

PLAY RECAP *********************************************************************
*
192.168.56.102             : ok=8    changed=0    unreachable=0    failed=0
skipped=3    rescued=0    ignored=0
192.168.56.103             : ok=9    changed=1    unreachable=0    failed=0
skipped=2    rescued=0    ignored=0
192.168.56.107             : ok=10   changed=1    unreachable=0    failed=0
skipped=3    rescued=0    ignored=0

diego@workstation:~/HOA7$
```

Reflections:

Answer the following:

1. What is the importance of creating roles?
   -Roles in Ansible are essential because they allow us to divide long playbooks into multiple files, making them easier to understand and work with. Roles are used to automate specific components and isolate them, making it easier to manage and maintain the playbook.
2. What is the importance of managing files?
   -Managing files is important because it simplifies work, reduces confusion, and makes it easier to locate errors and the parts that need to be configured or fixed.

Conclusion:

In this activity I learned a lot about roles and how to copy a file to a server. This activity teaches us on how to manage files and the importance of creating roles. I am hoping to learn more and be more knowledgeable.