Name: Tamayo, Ray Lan A.	Date Performed: 10/11/2024
Course/Section: CPE212-CPE31S21	Date Submitted: 10/11/2024
Instructor: Engr. Robin Valenzuela	Semester and SY: First 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**:

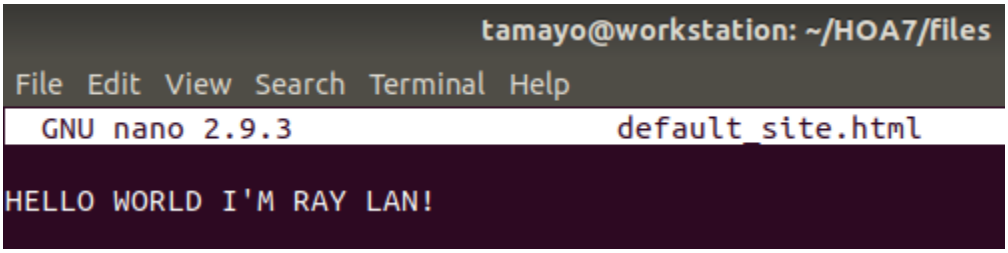
In this activity, we look at the concept of copying a file to a server. We are going to create a file into our git repository and use Ansible to grab that file and put it into a particular place so that we could do things like customize a default website, or maybe install a default configuration file.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

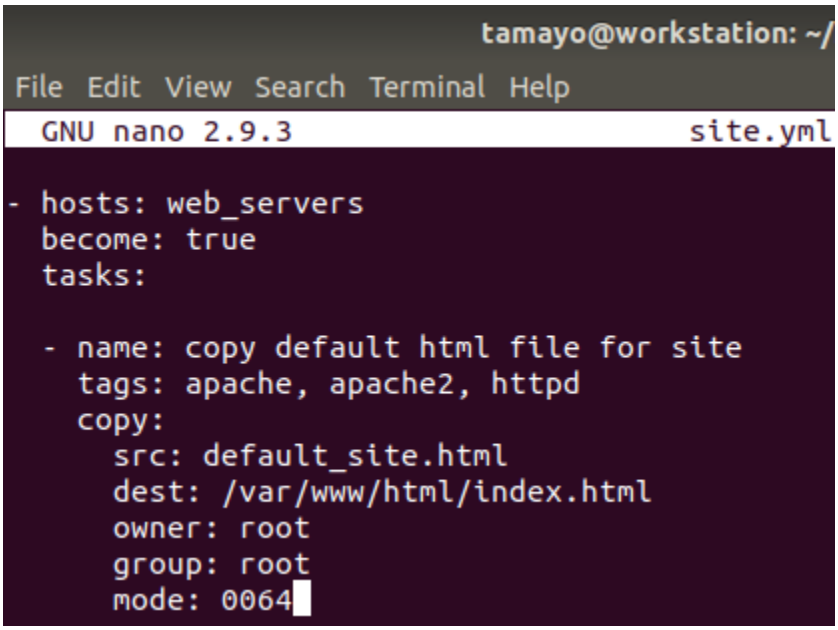
```
tamayo@workstation:~$ git clone git@github.com:raylantamayo/HOA7.git
Cloning into 'HOA7'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (3/3), done.
tamayo@workstation:~$
```

```
tamayo@workstation:~$ cd HOA7
tamayo@workstation:~/HOA7$ mkdir files
tamayo@workstation:~/HOA7$ cd files
tamayo@workstation:~/HOA7/files$ touch default_site.html
tamayo@workstation:~/HOA7/files$ sudo nano default_site.html
tamayo@workstation:~/HOA7/files$
```

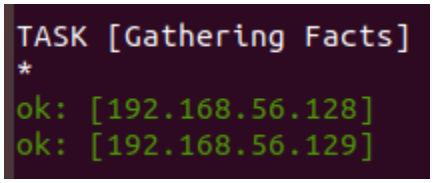


2. Edit the *site.yml* file and just below the *web_servers* play, create a new file to copy the default html file for site:
   - name: copy default html file for site

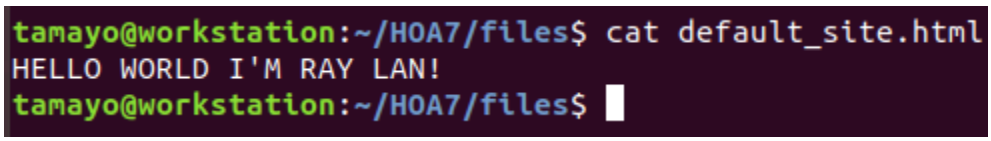
     tags: apache, apache2, httpd
     copy:
       src: default_site.html
       dest: /var/www/html/index.html
       owner: root
       group: root
       mode: 0644



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

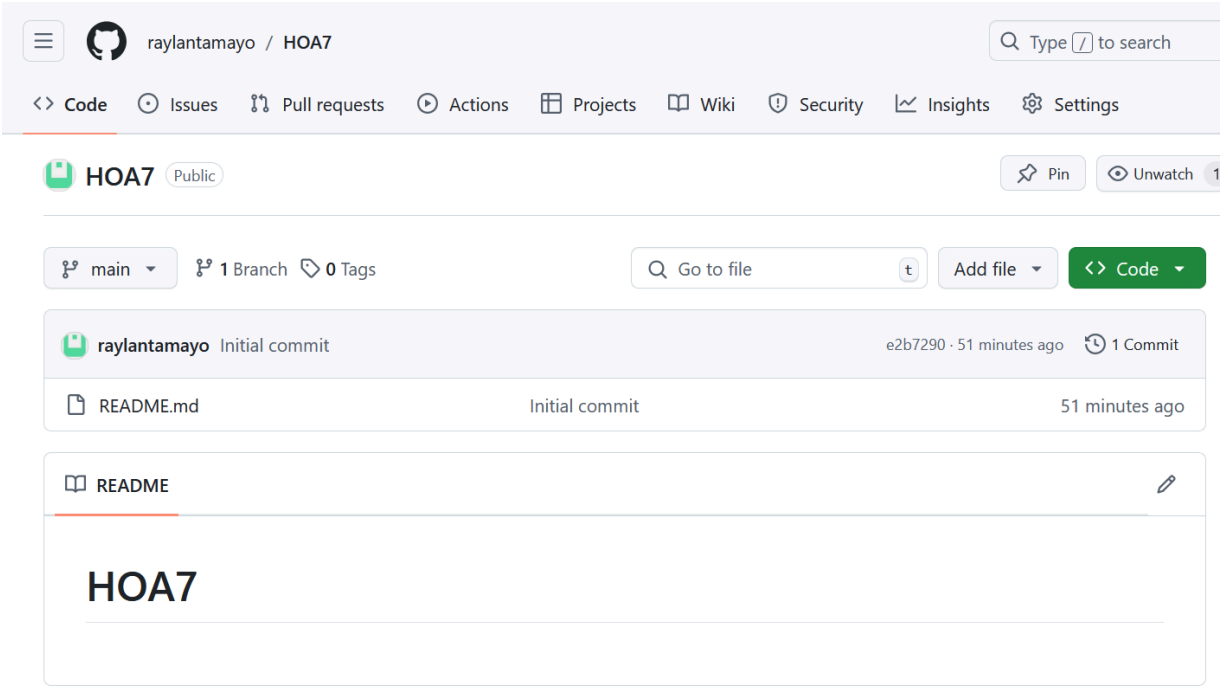


4. Go to the remote servers (*web_servers*) listed in your inventory. Use cat command to check if the index.html is the same as the local repository file (*default_site.html*). Do both for Ubuntu and CentOS servers. On the CentOS server, go to the browser and type its IP address. Describe the output.



HELLO WORLD I'M RAY LAN!

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

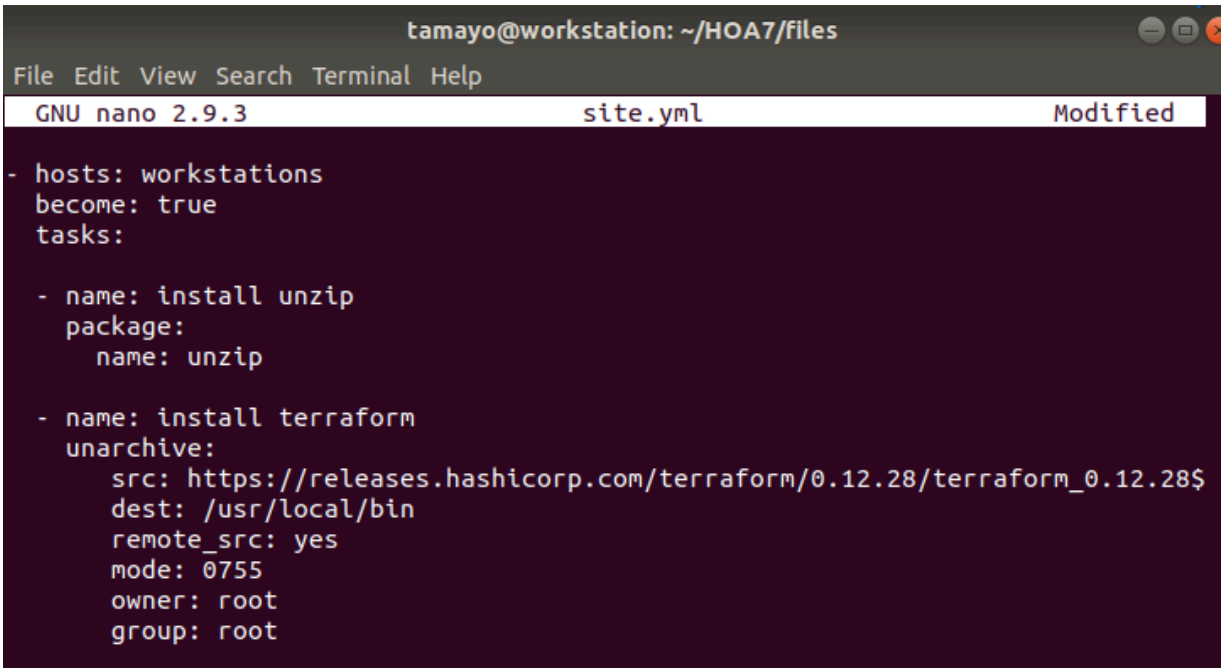


Task 2: Download a file and extract it to a remote server

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.

```
tamayo@workstation: ~/HOA7/files
File  Edit  View  Search  Terminal  Help
  GNU nano 2.9.3                          inventory

[web_servers]
192.168.56.128 ansible_ssh_private_key_file=~/.ssh/id_rsa
192.168.56.129 ansible_ssh_private_key_file=~/.ssh/id_rsa

[db_servers]
192.168.56.129 ansible_ssh_private_key_file=~/.ssh/id_rsa
192.168.56.130 ansible_ssh_private_key_file=~/.ssh/id_rsa

[file_servers]
192.168.56.130 ansible_ssh_private_key_file=~/.ssh/id_rsa

[workstations]
192.168.56.128 ansible_ssh_private_key_file=~/.ssh/id_rsa
```

3. Run the playbook. Describe the output.

```
tamayo@workstation: ~/HOA7/files
File  Edit  View  Search  Terminal  Help

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

TASK [Gathering Facts] *********************************
*
ok: [192.168.56.128]

TASK [install unzip] ***********************************
*
ok: [192.168.56.128]

TASK [install terraform] *******************************
*
changed: [192.168.56.128]

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

TASK [Gathering Facts] *********************************
*
ok: [192.168.56.129]
ok: [192.168.56.128]
```

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

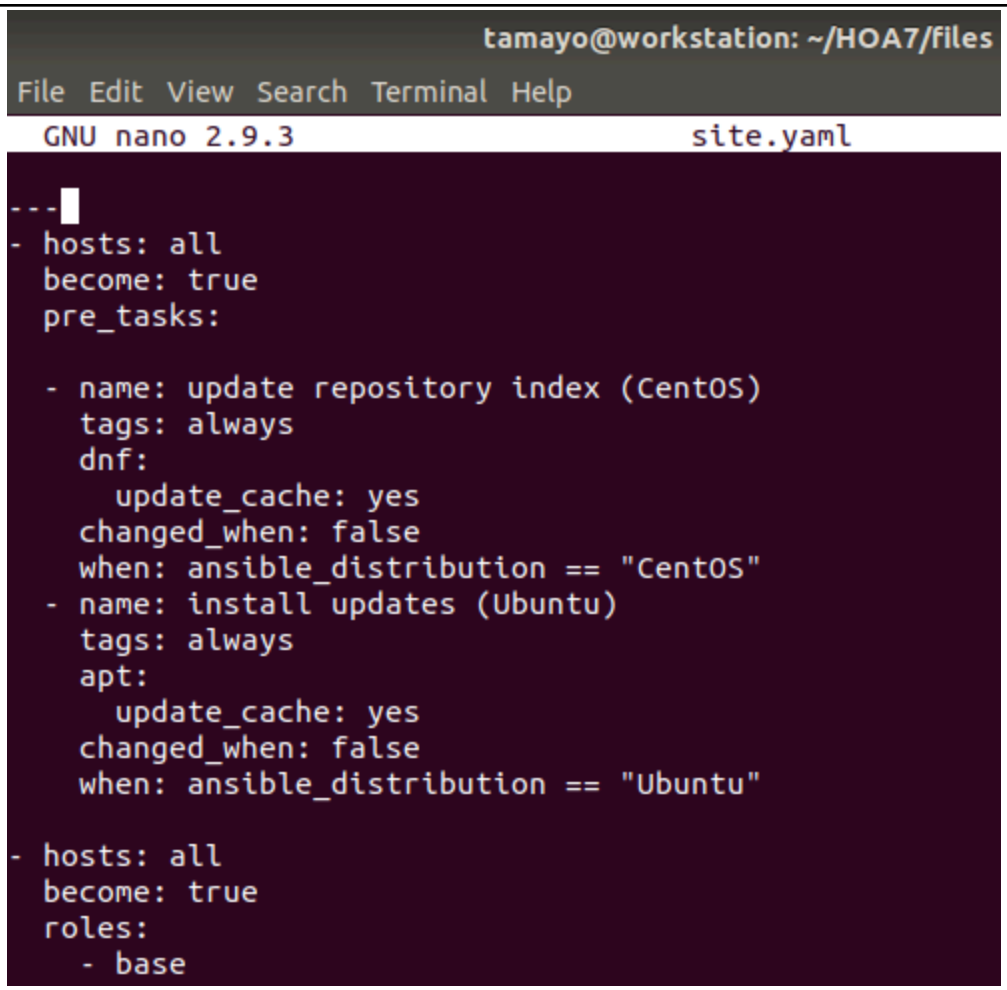
```
tamayo@workstation:~/HOA7/files$ terraform -v
Terraform v0.12.28

Your version of Terraform is out of date! The latest version
is 1.9.7. You can update by downloading from https://www.terraform.io/downloads
.html
tamayo@workstation:~/HOA7/files$
```

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.




```
- 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
```

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.

BASE

```
tamayo@workstation:~/HOA7$ mkdir roles
tamayo@workstation:~/HOA7$ cd roles
tamayo@workstation:~/HOA7/roles$ mkdir base
tamayo@workstation:~/HOA7/roles$ cd base
tamayo@workstation:~/HOA7/roles/base$ mkdir tasks
tamayo@workstation:~/HOA7/roles/base$ cd tasks
tamayo@workstation:~/HOA7/roles/base/tasks$ touch main.yml
tamayo@workstation:~/HOA7/roles/base/tasks$ sudo nano main.yml
```

WEB_SERVERS

```
tamayo@workstation:~/HOA7/roles$ mkdir web_servers
tamayo@workstation:~/HOA7/roles$ cd web_servers
tamayo@workstation:~/HOA7/roles/web_servers$ mkdir tasks
tamayo@workstation:~/HOA7/roles/web_servers$ cd tasks
tamayo@workstation:~/HOA7/roles/web_servers/tasks$ touch main.yml
tamayo@workstation:~/HOA7/roles/web_servers/tasks$ sudo nano main.yml
```

DB_SERVERS

```
tamayo@workstation:~/HOA7/roles$ mkdir db_servers
tamayo@workstation:~/HOA7/roles$ cd db_servers
tamayo@workstation:~/HOA7/roles/db_servers$ mkdir tasks
tamayo@workstation:~/HOA7/roles/db_servers$ touch main.yml
tamayo@workstation:~/HOA7/roles/db_servers$ sudo nano main.yml
```

FILE_SERVERS

```
tamayo@workstation:~/HOA7/roles$ mkdir file_servers
tamayo@workstation:~/HOA7/roles$ cd file_servers
tamayo@workstation:~/HOA7/roles/file_servers$ mkdir tasks
tamayo@workstation:~/HOA7/roles/file_servers$ cd tasks
tamayo@workstation:~/HOA7/roles/file_servers/tasks$ touch main.yml
tamayo@workstation:~/HOA7/roles/file_servers/tasks$ sudo nano main.yml
```

WORKSTATIONS

```
tamayo@workstation:~/HOA7/roles$ mkdir workstations
tamayo@workstation:~/HOA7/roles$ cd workstations
tamayo@workstation:~/HOA7/roles/workstations$ mkdir tasks
tamayo@workstation:~/HOA7/roles/workstations$ cd tasks
tamayo@workstation:~/HOA7/roles/workstations/tasks$ touch main.yml
tamayo@workstation:~/HOA7/roles/workstations/tasks$ sudo nano main.yml
```

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.

BASE

```
                                tamayo@workstation: ~/HOA7/roles/base/tasks
File  Edit  View  Search  Terminal  Help
  GNU nano 2.9.3                          main.yml
```

WEB_SERVERS

```
                                tamayo@workstation: ~/HOA7/roles/web_servers/tasks
File  Edit  View  Search  Terminal  Help
  GNU nano 2.9.3                          main.yml

- 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: 0064

- name: install apache and php for Ubuntu servers
  tags: apache,apache2,ubuntu
  apt:
    name:
      - apache2
      - libapache2-mod-php
    state: latest
  when: ansible_distribution == "Ubuntu"

- name: install apache and php for CentOS servers
  tags: apache,centos,httpd
  dnf:
    name:
      - httpd
```

```
      - php
    state: latest
  when: ansible_distribution == "CentOS"

- name: start httpd (CentOS)
  tags: apache, centos,httpd
  service:
    name: httpd
    state: started
    enabled: true
  when: ansible_distribution == "CentOS"
```

DB_SERVERS

tamayo@workstation: ~/HOA7/roles/db_servers

File Edit View Search Terminal Help

GNU nano 2.9.3 main.yml

```
- name: install mariadb package (CentOS)
  tags: centos, db,mariadb
  dnf:
    name: mariadb-server
    state: latest
  when: ansible_distribution == "CentOS"

- name: "Mariadb - Restarting/Enabling"
  service:
    name: mariadb
    state: restarted
    enabled: true

- name: install mariadb package (Ubuntu)
  tags: db, mariadb,ubuntu
  apt:
    name: mariadb-server
    state: latest
  when: ansible_distribution == "Ubuntu"

- name: "Mariadb - Restarting/Enabling"
  service:
    name: mariadb
```

```
    state: restarted
    enabled: true
```

FILE_SERVERS

```
tamayo@workstation: ~/HOA7/roles/file_servers/tasks
File  Edit  View  Search  Terminal  Help
 GNU nano 2.9.3                    main.yml

- name: install samba package
  tags: samba
  package:
    name: samba
    state: latest
```

WORKSTATIONS

```
tamayo@workstation: ~/HOA7/roles/workstations/tasks
File  Edit  View  Search  Terminal  Help
 GNU nano 2.9.3                  main.yml                  Modified


- name: install unzip
  package:
    name: unzip

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

```
tamayo@workstation:~/HOA7/roles$ tree
.
├── base
│   └── tasks
│       └── main.yml
├── db_servers
│   ├── main.yml
│   └── tasks
├── file_servers
│   ├── main.yml
│   └── tasks
│       └── main.yml
├── web_servers
│   └── tasks
│       └── main.yml
└── workstations
    └── tasks
        └── main.yml

10 directories, 6 files
```

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

It produces the same results as the old site.yml; the only change is that roles have been assigned now.

GITHUB LINK: https://github.com/raylantamayo/HOA7

Reflections:

Answer the following:

1. What is the importance of creating roles?

   To automate tasks smoothly in an Ubuntu playbook, you need to create roles. Roles help keep tasks organized and make playbooks easier to manage, which means you can reuse code more effectively. This approach makes it easier for teams to work together and maintain the system, ensuring that server setups are reliable and consistent. In the end, this saves time and reduces mistakes.

2. What is the importance of managing files?

   In Ubuntu, managing files is important for keeping your data organized, ensuring the system runs smoothly, and making the best use of storage. It helps you find, use, and arrange documents, programs, and settings easily. Good file management also boosts security by controlling who can access what. Overall, it keeps your Ubuntu system efficient, secure, and well-organized.

CONCLUSION

After doing this activity, I got to work with roles. While testing the playbook, I found that using roles makes complicated tasks much easier in an Ubuntu playbook. They let me group related tasks into reusable modules, which helps keep things consistent and reduces repetitive code. This makes it easier to maintain the playbook and ensures that managing configurations is more efficient. Overall, using roles simplifies the creation of playbooks, making them easier to manage. I really enjoyed this activity because it will be very helpful for future projects we'll work on.