Name: Daniela Marie D. Rabang	**Date Performed:** 10/09/2023
Course/Section: CPE232/CPE31S4	**Date Submitted:** 10/10/2023
Instructor: Dr. Jonathan V. Taylar	**Semester and SY:** 1st Sem 2023-2024

Activity 7: Managing Files and Creating Roles in Ansible

1. Objectives:

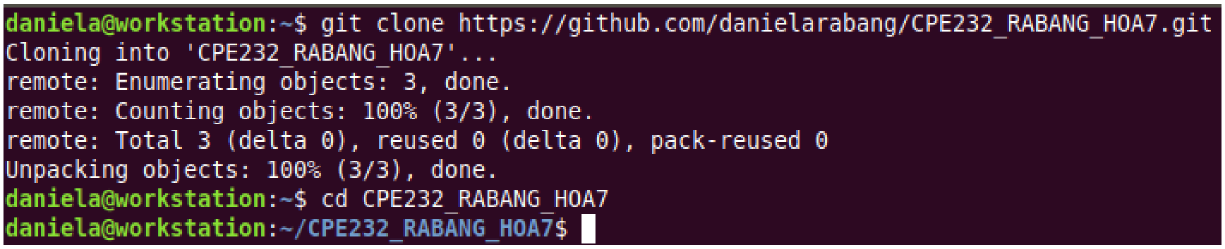
1.1 Manage files in remote servers

1.2 Implement roles in ansible

2. Discussion:

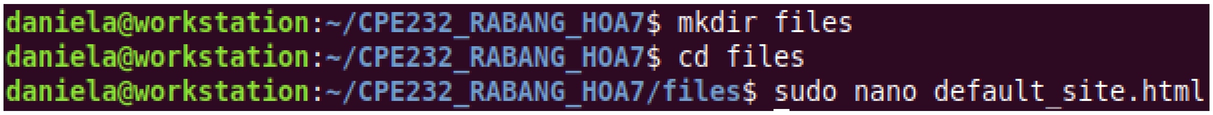
In this activity, we look at the concept of copying a file to a server. We are going to create a file into our git repository and use Ansible to grab that file and put it into a particular place so that we could do things like customize a default website, or maybe install a default configuration file.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

Task 1: Create a file and copy it to remote servers

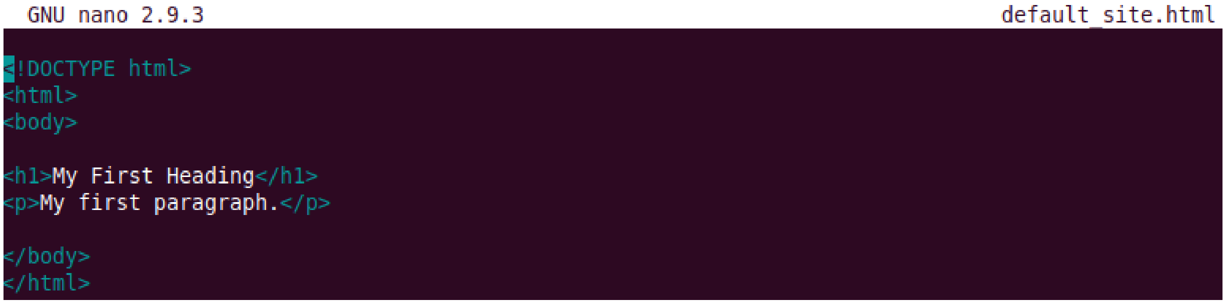


```
daniela@workstation:~$ git clone https://github.com/danielarabang/CPE232_RABANG_HOA7.git
Cloning into 'CPE232_RABANG_HOA7'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
daniela@workstation:~$ cd CPE232_RABANG_HOA7
daniela@workstation:~/CPE232_RABANG_HOA7$
```

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.



```
daniela@workstation:~/CPE232_RABANG_HOA7$ mkdir files
daniela@workstation:~/CPE232_RABANG_HOA7$ cd files
daniela@workstation:~/CPE232_RABANG_HOA7/files$ sudo nano default_site.html
```

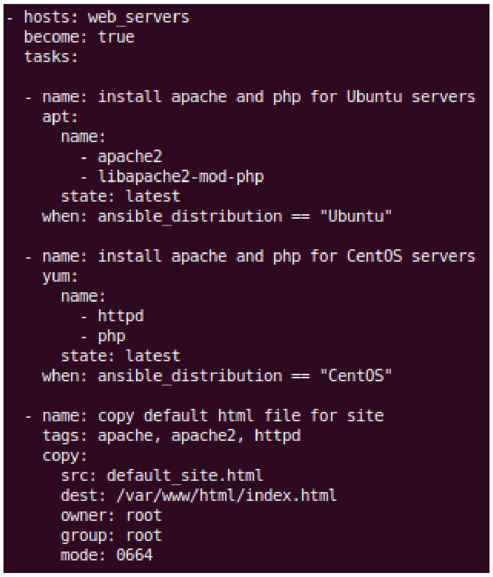


```
  GNU nano 2.9.3                                                default_site.html

<!DOCTYPE html>
<html>
<body>

<h1>My First Heading</h1>
<p>My first paragraph.</p>

</body>
</html>
```

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

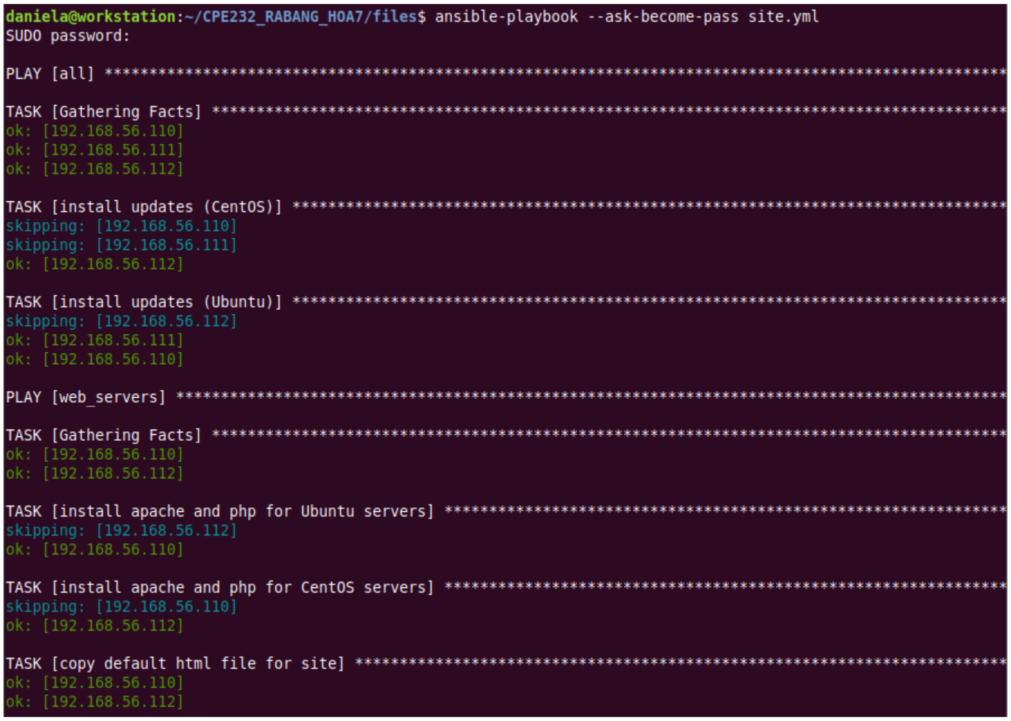
     tags: apache, apache2, httpd

```
copy:
  src: default_site.html
  dest: /var/www/html/index.html
  owner: root
  group: root
  mode: 0644
```



```
- hosts: web_servers
  become: true
  tasks:

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

  - name: install apache and php for CentOS servers
    yum:
      name:
        - httpd
        - php
      state: latest
    when: ansible_distribution == "CentOS"

  - 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: 0664
```

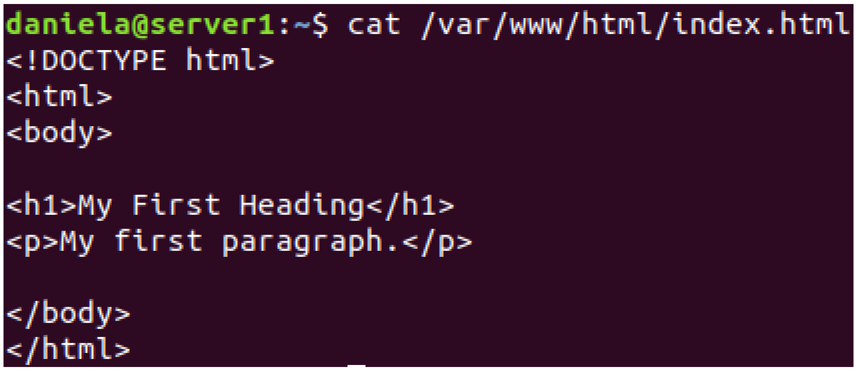
3. Run the playbook *site.yml*. Describe the changes.



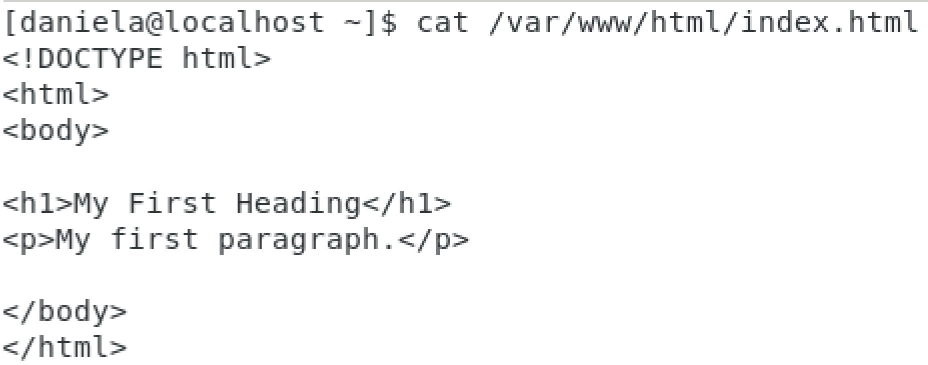
```
daniela@workstation:~/CPE232_RABANG_HOA7/files$ ansible-playbook --ask-become-pass site.yml
SUDO password:

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

TASK [Gathering Facts] ***************************************************************************
ok: [192.168.56.110]
ok: [192.168.56.111]
ok: [192.168.56.112]

TASK [install updates (CentOS)] ******************************************************************
skipping: [192.168.56.110]
skipping: [192.168.56.111]
ok: [192.168.56.112]

TASK [install updates (Ubuntu)] ******************************************************************
skipping: [192.168.56.112]
ok: [192.168.56.111]
ok: [192.168.56.110]

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

TASK [Gathering Facts] ***************************************************************************
ok: [192.168.56.110]
ok: [192.168.56.112]

TASK [install apache and php for Ubuntu servers] *************************************************
skipping: [192.168.56.112]
ok: [192.168.56.110]

TASK [install apache and php for CentOS servers] *************************************************
skipping: [192.168.56.110]
ok: [192.168.56.112]

TASK [copy default html file for site] ***********************************************************
ok: [192.168.56.110]
ok: [192.168.56.112]
```

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.

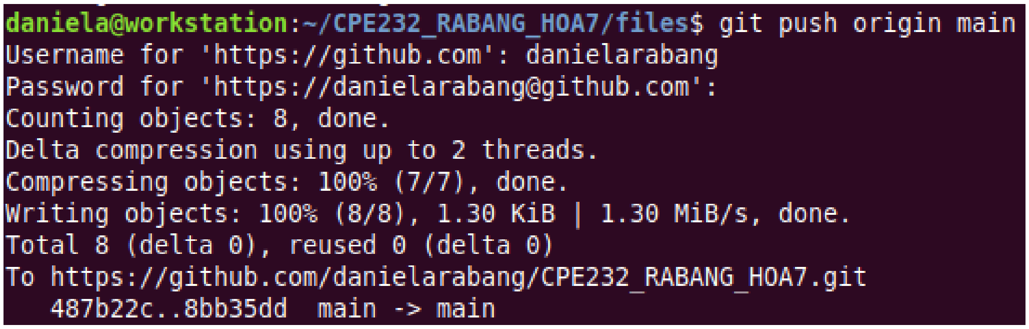


```
daniela@server1:~$ cat /var/www/html/index.html
<!DOCTYPE html>
<html>
<body>

<h1>My First Heading</h1>
<p>My first paragraph.</p>

</body>
</html>
```



```
[daniela@localhost ~]$ cat /var/www/html/index.html
<!DOCTYPE html>
<html>
<body>

<h1>My First Heading</h1>
<p>My first paragraph.</p>

</body>
</html>
```

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



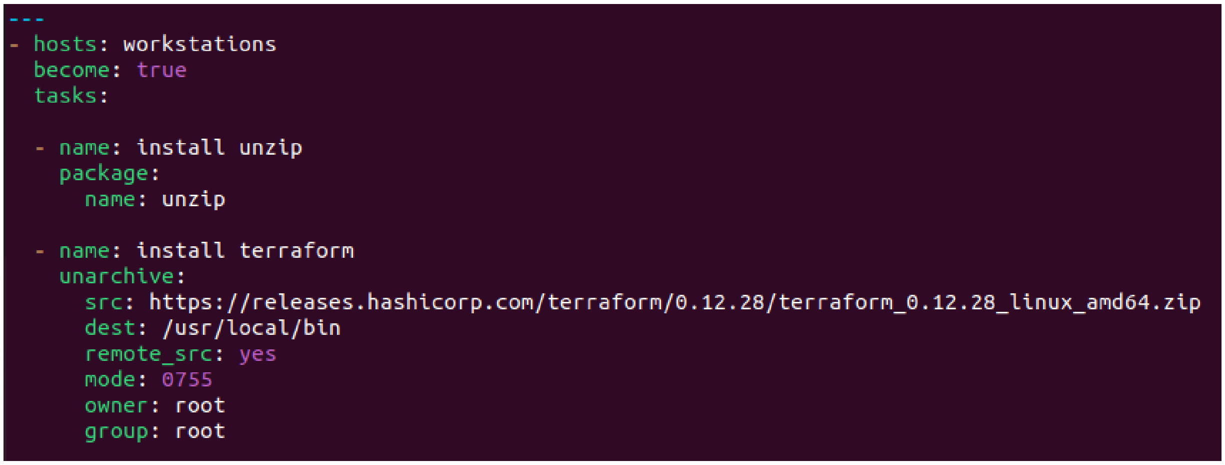
```
daniela@workstation:~/CPE232_RABANG_HOA7/files$ git push origin main
Username for 'https://github.com': danielarabang
Password for 'https://danielarabang@github.com':
Counting objects: 8, done.
Delta compression using up to 2 threads.
Compressing objects: 100% (7/7), done.
Writing objects: 100% (8/8), 1.30 KiB | 1.30 MiB/s, done.
Total 8 (delta 0), reused 0 (delta 0)
To https://github.com/danielarabang/CPE232_RABANG_HOA7.git
   487b22c..8bb35dd  main -> main
```

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:

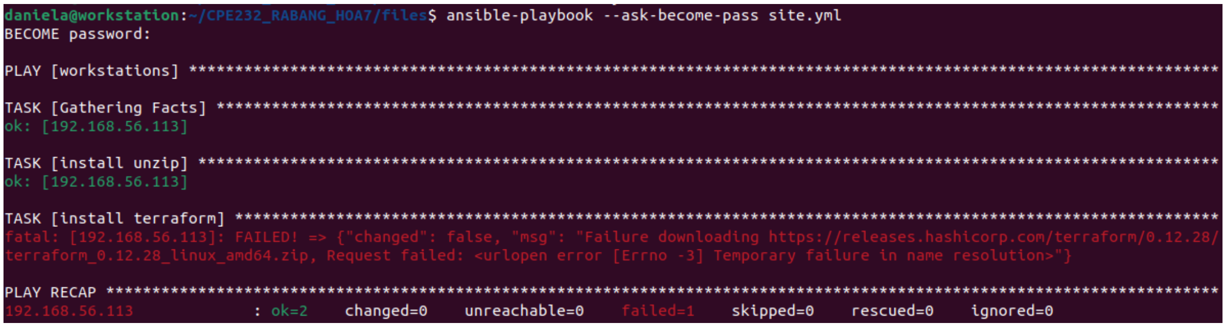
```
- hosts: workstations
  become: true
  tasks:

  - 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_0.12.28_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.

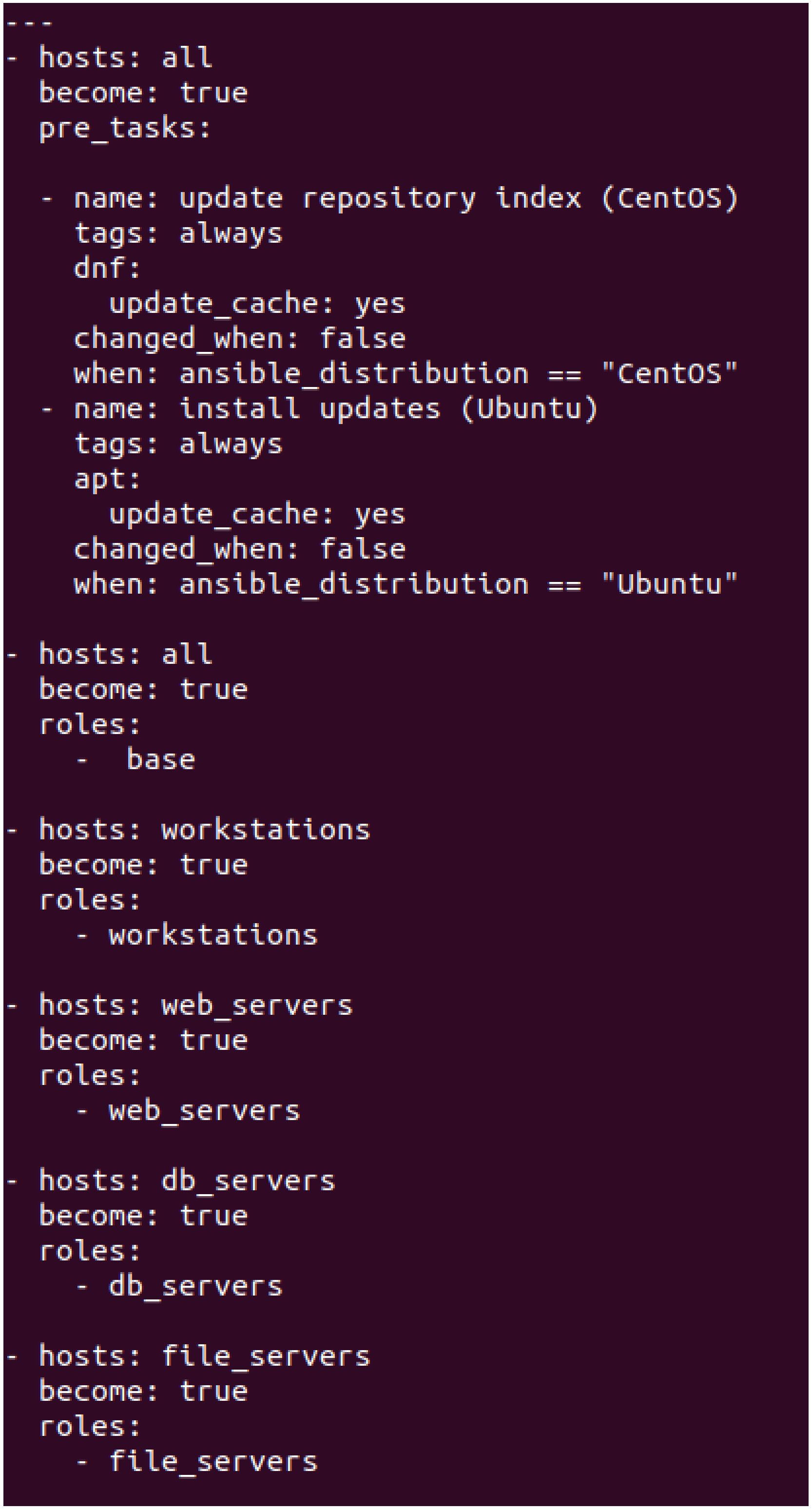


```
daniela@workstation:~/CPE232_RABANG_HOA7/files$ ansible-playbook --ask-become-pass site.yml
BECOME password:

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

TASK [Gathering Facts] ********************************************************************************************
ok: [192.168.56.113]

TASK [install unzip] **********************************************************************************************
ok: [192.168.56.113]

TASK [install terraform] ******************************************************************************************
fatal: [192.168.56.113]: FAILED! => {"changed": false, "msg": "Failure downloading https://releases.hashicorp.com/terraform/0.12.28/
terraform_0.12.28_linux_amd64.zip, Request failed: <urlopen error [Errno -3] Temporary failure in name resolution>"}

PLAY RECAP ********************************************************************************************************
192.168.56.113             : ok=2    changed=0    unreachable=0    failed=1    skipped=0    rescued=0    ignored=0
```

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

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)

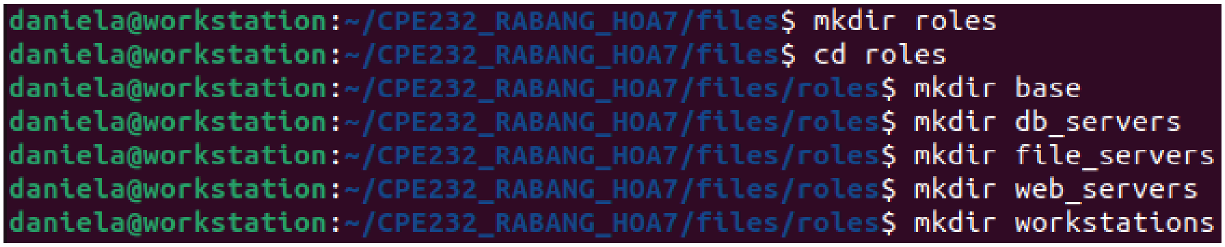


```
---
- hosts: all
  become: true
  pre_tasks:

  - 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
```

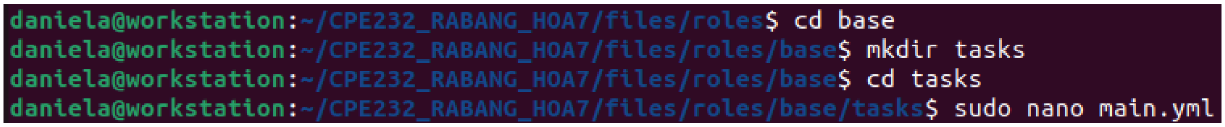
Save the file and exit.

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,

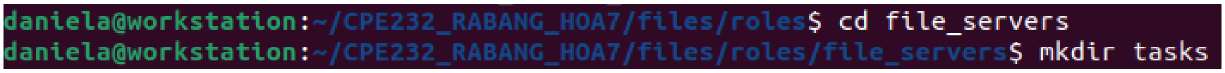
db_servers and workstations. For each directory, create a directory and name it tasks.



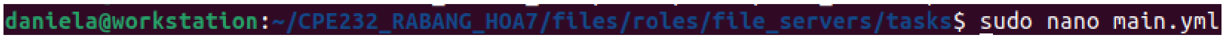
```
daniela@workstation:~/CPE232_RABANG_HOA7/files$ mkdir roles
daniela@workstation:~/CPE232_RABANG_HOA7/files$ cd roles
daniela@workstation:~/CPE232_RABANG_HOA7/files/roles$ mkdir base
daniela@workstation:~/CPE232_RABANG_HOA7/files/roles$ mkdir db_servers
daniela@workstation:~/CPE232_RABANG_HOA7/files/roles$ mkdir file_servers
daniela@workstation:~/CPE232_RABANG_HOA7/files/roles$ mkdir web_servers
daniela@workstation:~/CPE232_RABANG_HOA7/files/roles$ mkdir workstations
```



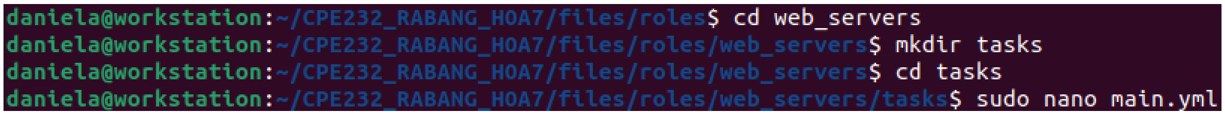
```
daniela@workstation:~/CPE232_RABANG_HOA7/files/roles$ cd base
daniela@workstation:~/CPE232_RABANG_HOA7/files/roles/base$ mkdir tasks
daniela@workstation:~/CPE232_RABANG_HOA7/files/roles/base$ cd tasks
daniela@workstation:~/CPE232_RABANG_HOA7/files/roles/base/tasks$ sudo nano main.yml
```



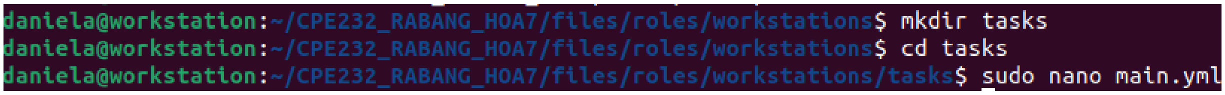
```
daniela@workstation:~/CPE232_RABANG_HOA7/files/roles$ cd file_servers
daniela@workstation:~/CPE232_RABANG_HOA7/files/roles/file_servers$ mkdir tasks
```



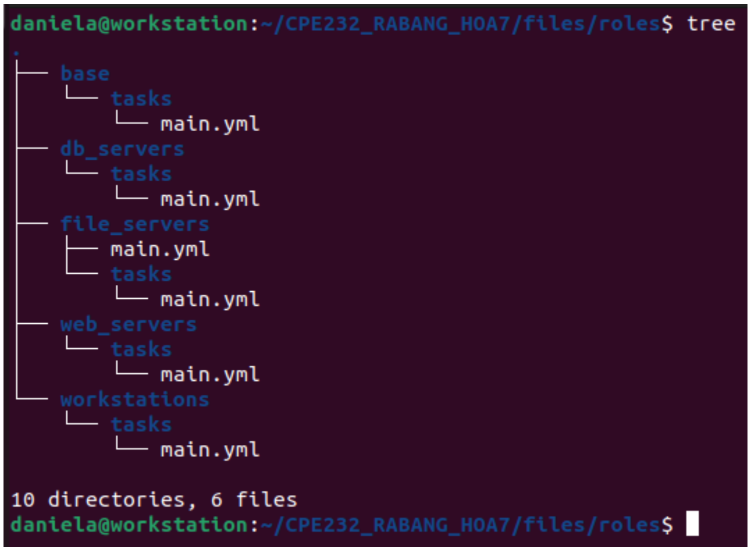
```
daniela@workstation:~/CPE232_RABANG_HOA7/files/roles/file_servers/tasks$ sudo nano main.yml
```



```
daniela@workstation:~/CPE232_RABANG_HOA7/files/roles$ cd web_servers
daniela@workstation:~/CPE232_RABANG_HOA7/files/roles/web_servers$ mkdir tasks
daniela@workstation:~/CPE232_RABANG_HOA7/files/roles/web_servers$ cd tasks
daniela@workstation:~/CPE232_RABANG_HOA7/files/roles/web_servers/tasks$ sudo nano main.yml
```

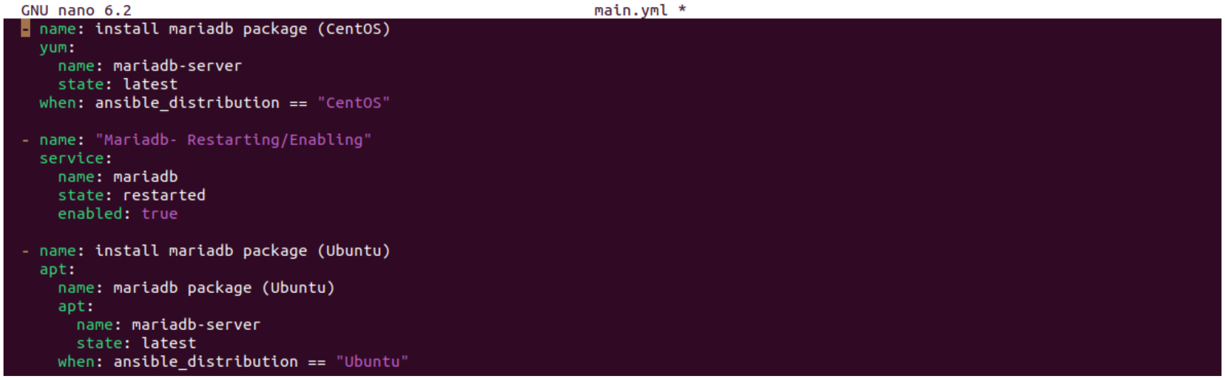


```
daniela@workstation:~/CPE232_RABANG_HOA7/files/roles/workstations$ mkdir tasks
daniela@workstation:~/CPE232_RABANG_HOA7/files/roles/workstations$ cd tasks
daniela@workstation:~/CPE232_RABANG_HOA7/files/roles/workstations/tasks$ sudo nano main.yml
```

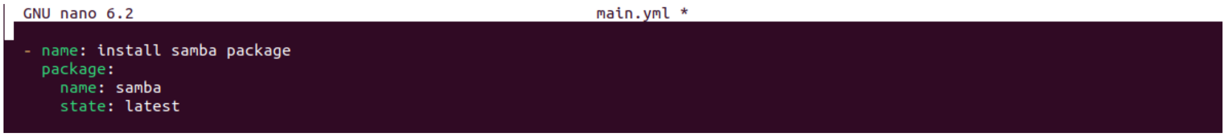


```
daniela@workstation:~/CPE232_RABANG_HOA7/files/roles$ tree
.
├── base
│   └── tasks
│       └── main.yml
├── db_servers
│   └── tasks
│       └── main.yml
├── file_servers
│   ├── main.yml
│   └── tasks
│       └── main.yml
├── web_servers
│   └── tasks
│       └── main.yml
└── workstations
    └── tasks
        └── main.yml

10 directories, 6 files
daniela@workstation:~/CPE232_RABANG_HOA7/files/roles$
```

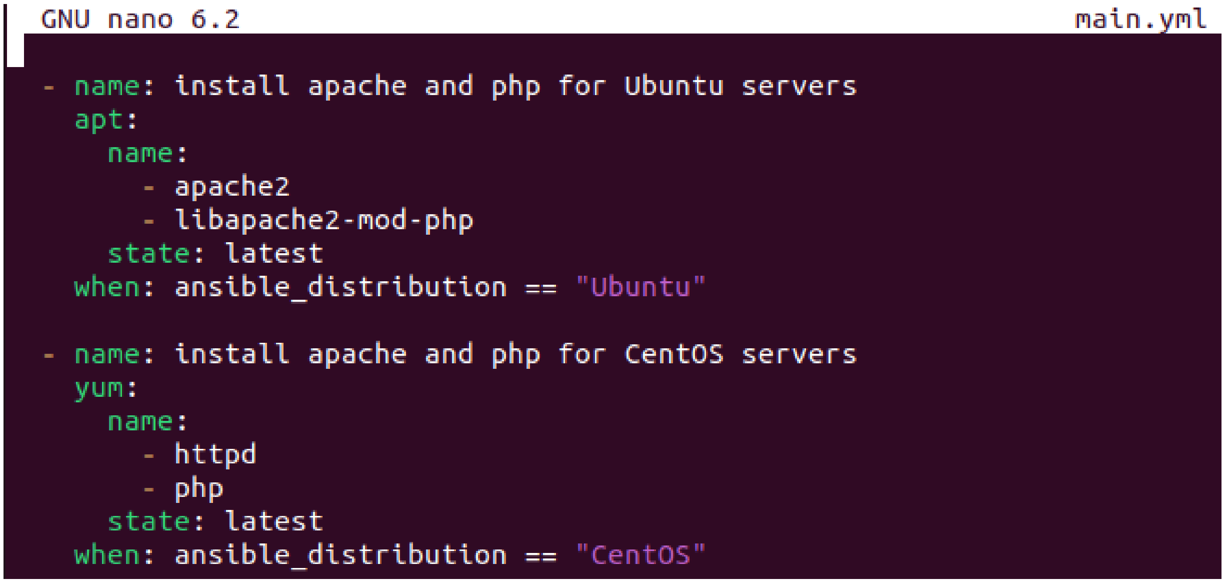
3. Go to tasks for all directory and create a file. Name it main.yml. In each of the tasks for all directories, copy and paste the code from the old site.yml file. Show all contents of main.yml files for all tasks.



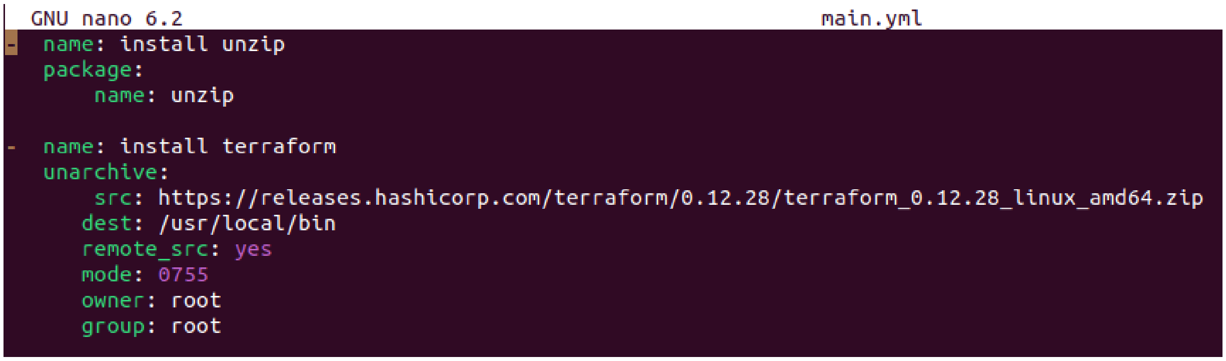
```
GNU nano 6.2                                   main.yml *
- name: install mariadb package (CentOS)
  yum:
    name: mariadb-server
    state: latest
  when: ansible_distribution == "CentOS"

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

- name: install mariadb package (Ubuntu)
  apt:
    name: mariadb package (Ubuntu)
    apt:
      name: mariadb-server
      state: latest
    when: ansible_distribution == "Ubuntu"
```



```
GNU nano 6.2                                   main.yml *

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



```
GNU nano 6.2                                   main.yml

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

- name: install apache and php for CentOS servers
  yum:
    name:
      - httpd
      - php
    state: latest
  when: ansible_distribution == "CentOS"
```



```
GNU nano 6.2                                   main.yml
- 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
```

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

Reflections:

Answer the following:

1. What is the importance of creating roles?

- It is important to create roles for specific tasks, this is because it helps the system to know where the tasks need to be done and executed. By this the process will be more light and easier not just for the user.

2. **What is the importance of managing files?**

- Managing files is important so that the user can have a hold on the files that are in the systems. This is also important for the user, so that they can have access from all the files, information, and documents.

Conclusion:

In this hands-on activity which tackles managing files and also creating roles in ansible, From the start of the activity, I had created a new repository for this specific hands-on activity. Then I used the command git clone so that I can access the repository through my workstation. In task 1, I had to create a directory inside my hoa repository directory. After that I had created a default_site.html file that I will use on the next task. Then next I had edited my site.yml using all the commands on my playbook that I had used last hands-on activity. I also created a site.yml and ran it. after running all the tasks that are in the playbook. Then I tested if the html file was there, after that I had pushed and committed all the changes that I had done for task number one through my repository. For task 2, which is the file to a remote server, I had modified the site.yml where I can run a task that has a specific host to install unzip, and the terraform. Then after that I had encountered a lot of errors saying that it had failed downloading. Then on the tasks 3, I had created roles on where I had created directory roles then in that I created a directory tasks and inside that is the main.yml file where I had put all the commands that I had used in the previous activities that is for that role.