Name: Pacinos, Angela Monique A.	**Date Performed:** 10-09-23
Course/Section: CPE232 - CPE31S4	**Date Submitted:** 10-10-23
Instructor: Dr. Jonathan V. Taylar	**Semester and SY:** 1st Sem '23 - '24

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**:

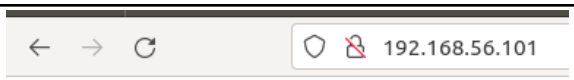
In this activity, we look at the concept of copying a file to a server. We are going to create a file into our git repository and use Ansible to grab that file and put it into a particular place so that we could do things like customize a default website, or maybe install a default configuration file.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

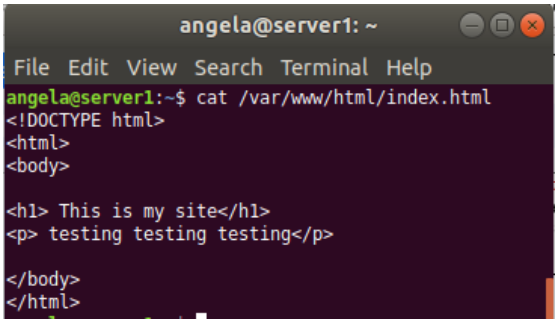
```
- name: copy default html file for site

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

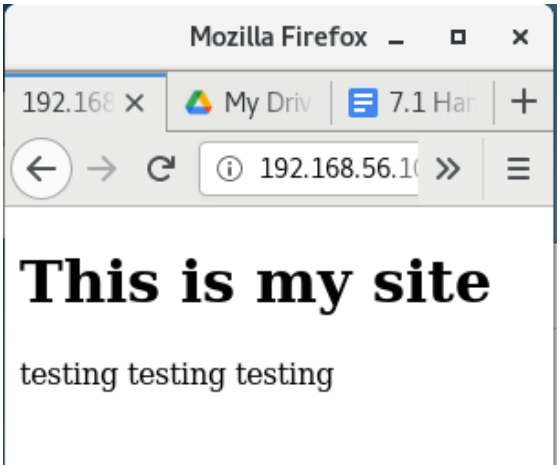
3. Run the playbook *site.yml*. Describe the changes.
4. Go to the remote servers (*web_servers*) listed in your inventory. Use cat command to check if the index.html is the same as the local repository file (*default_site.html*). Do both for Ubuntu and CentOS servers. On the CentOS server, go to the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

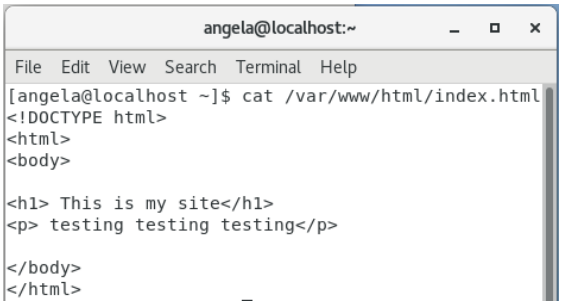


This is my site

testing testing testing





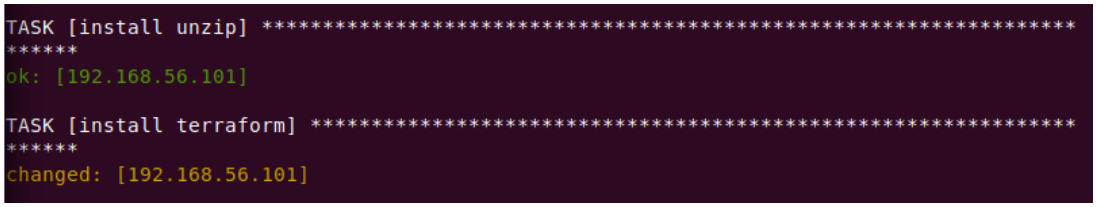


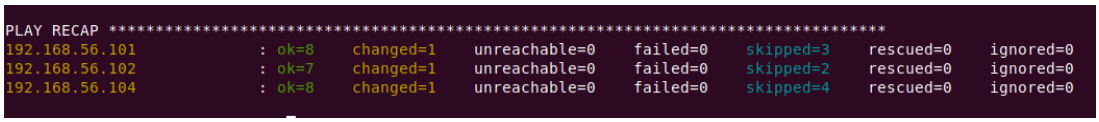
Task 2: Download a file and extract it to a remote server

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.
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.

```
angela@workstation:~/HOA7_S4/roles$ ls
base  db_servers  file_servers  web_servers  workstations
```

```
angela@workstation:~/HOA7_S4/roles$ tree
.
├── base
│   └── tasks
│       └── main.yml
├── db_servers
│   └── tasks
│       └── main.yml
├── file_servers
│   └── tasks
│       └── main.yml
├── web_servers
│   └── tasks
│       └── main.yml
└── workstations
    └── tasks
        └── main.yml

10 directories, 5 files
```

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.

web_servers

Open ▾ main.yml ~/HOA7_S4/roles/web_serve... Save

```
- 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
  yum:
    name:
      - httpd
      - php
    state: latest
  when: ansible_distribution == "CentOS"

- name: start httpd (CentOS)
  tags: apache,centos,httpd
  service:
    name: httpd
    state: started
  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: 0644
```

db_servers

main.yml [Read-Only]
~/HOA7_S4/roles/db_servers...

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

- name: "Mariadb- Restarting/Enabling"
  tags: db,mariadb,ubuntu
  service:
    name: mariadb
    state: restarted
    enabled: true

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

file_servers

main.yml ...
~/HOA7_S4/...

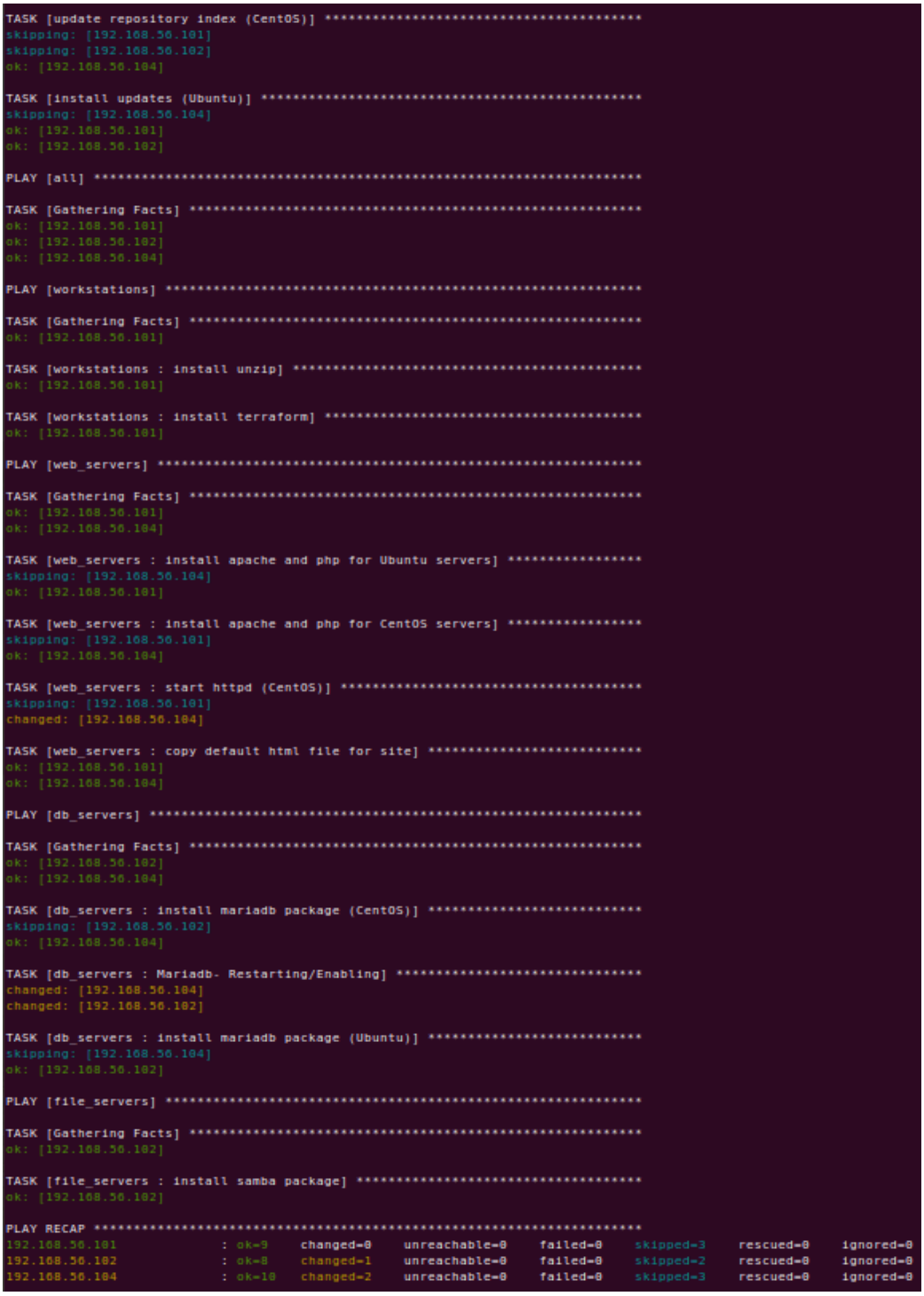
```yaml
- name: install samba package
  tags: samba
  package:
    name: samba
    state: latest
```

workstations

main.yml [Read-Only]
~/HOA7_S4/roles/workstations/tasks

```yaml
- 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.



```
TASK [update repository index (CentOS)] ************************************
skipping: [192.168.56.101]
skipping: [192.168.56.102]
ok: [192.168.56.104]

TASK [install updates (Ubuntu)] ********************************************
skipping: [192.168.56.104]
ok: [192.168.56.101]
ok: [192.168.56.102]

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

TASK [Gathering Facts] *****************************************************
ok: [192.168.56.101]
ok: [192.168.56.102]
ok: [192.168.56.104]

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

TASK [Gathering Facts] *****************************************************
ok: [192.168.56.101]

TASK [workstations : install unzip] ****************************************
ok: [192.168.56.101]

TASK [workstations : install terraform] ************************************
ok: [192.168.56.101]

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

TASK [Gathering Facts] *****************************************************
ok: [192.168.56.101]
ok: [192.168.56.104]

TASK [web_servers : install apache and php for Ubuntu servers] *************
skipping: [192.168.56.104]
ok: [192.168.56.101]

TASK [web_servers : install apache and php for CentOS servers] *************
skipping: [192.168.56.101]
ok: [192.168.56.104]

TASK [web_servers : start httpd (CentOS)] **********************************
skipping: [192.168.56.101]
changed: [192.168.56.104]

TASK [web_servers : copy default html file for site] ***********************
ok: [192.168.56.101]
ok: [192.168.56.104]

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

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

TASK [db_servers : install mariadb package (CentOS)] ***********************
skipping: [192.168.56.102]
ok: [192.168.56.104]

TASK [db_servers : Mariadb- Restarting/Enabling] ***************************
changed: [192.168.56.104]
changed: [192.168.56.102]

TASK [db_servers : install mariadb package (Ubuntu)] ***********************
skipping: [192.168.56.104]
ok: [192.168.56.102]

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

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

TASK [file_servers : install samba package] ********************************
ok: [192.168.56.102]

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

After running the site.yml and a little bit of working. as can be seen in the screenshot that the playbook runed and executed well and properly.

Reflections:

Answer the following:

1. **What is the importance of creating roles?**

   Creating roles when using or managing playbooks allows the administrator to separate the different services that they are doing to the system and remote servers. It allows for more organization because different services can be grouped and placed into the same playbook which will be easier to modify later on. Doing these roles allows us to specifically locate where, as the system tells us, the error is coming from thus making it easy to modify and change that part.

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

   First importance is Organization. Managing files properly allows for easier organization of the files within the system. It is easier to navigate or locate something around the system if the environment that we are working on is properly organized. Along with this is to avoid the accidental loss or erasure of files that we thought are not important. This is one of the cases that we are trying to avoid. That's why we organize and manage our files. We also do this to keep track of the changes and the versions that a single document had.