Name: Andaya, Lyka C	Date Performed: October 9, 2023
Course/Section: CPE31S4	Date Submitted: October 10, 2023
Instructor: Dr. Taylar	Semester and SY: 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

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.

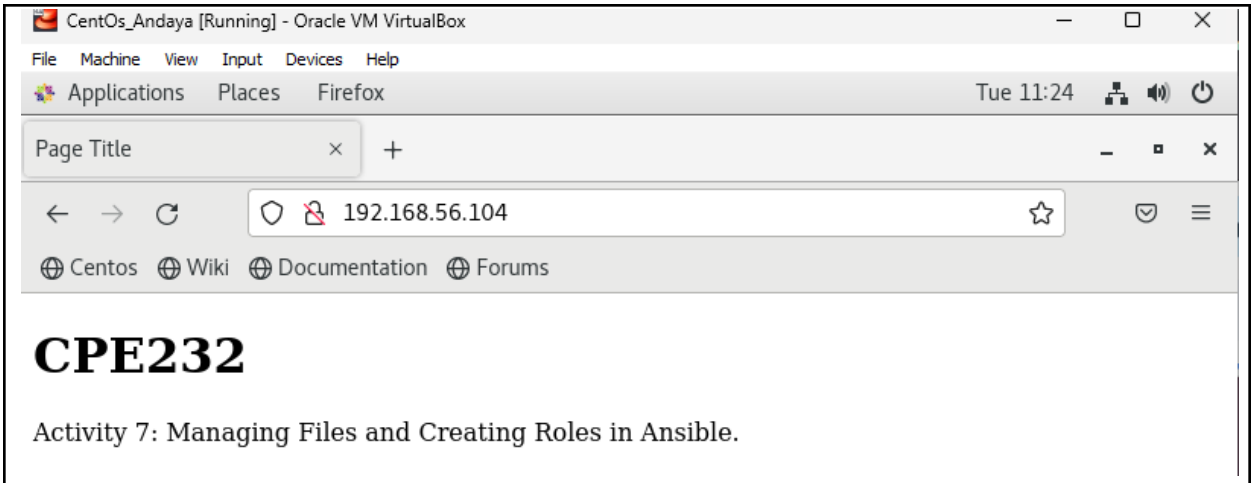
```
andayalyka@managenode:~/HOA7$ ansible-playbook --ask-become-pass site.yml
[DEPRECATION WARNING]: Ansible will require Python 3.8 or newer on the
controller starting with Ansible 2.12. Current version: 3.6.9 (default, Mar 10
2023, 16:46:00) [GCC 8.4.0]. This feature will be removed from ansible-core in
version 2.12. Deprecation warnings can be disabled by setting
deprecation_warnings=False in ansible.cfg.
BECOME password:

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

TASK [Gathering Facts] *********************************************************
ok: [192.168.56.103]
ok: [192.168.56.104]
ok: [192.168.56.102]

TASK [install updates (CentOS)] ************************************************
skipping: [192.168.56.102]
skipping: [192.168.56.103]
ok: [192.168.56.104]

TASK [install updates (Ubuntu)] ************************************************
skipping: [192.168.56.104]
ok: [192.168.56.102]
ok: [192.168.56.103]

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

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

TASK [install apache and php for Ubuntu servers] *******************************
skipping: [192.168.56.104]
ok: [192.168.56.102]
```

```
TASK [install apache and php for CentOS servers] ******************************
skipping: [192.168.56.102]
ok: [192.168.56.104]

TASK [start httpd (CentOS)] ***************************************************
skipping: [192.168.56.102]
ok: [192.168.56.104]

TASK [copy default html file for site] ****************************************
changed: [192.168.56.102]
changed: [192.168.56.104]

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

TASK [Gathering Facts] ********************************************************
ok: [192.168.56.104]
ok: [192.168.56.103]

TASK [install mariadb package (CentOS)] ***************************************
skipping: [192.168.56.103]
ok: [192.168.56.104]

TASK [Mariadb- Restarting/Enabling] *******************************************
changed: [192.168.56.103]
changed: [192.168.56.104]

TASK [install mariadb package (Ubuntu)] ***************************************
skipping: [192.168.56.104]
ok: [192.168.56.103]

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

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

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

Description: ***It verify that it copies the default_site.html for site***

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.



Description: ***It display the command that I created in the default_site.html using HTML syntax/code.***

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

```
andayalyka@managenode:~/HOA7$ git add *
andayalyka@managenode:~/HOA7$ git commit -m "ACT7"
[main 6f2bd70] ACT7
 2 files changed, 21 insertions(+), 1 deletion(-)
 create mode 100644 files/default_site.html
andayalyka@managenode:~/HOA7$ git push
Counting objects: 5, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (4/4), done.
Writing objects: 100% (5/5), 666 bytes | 666.00 KiB/s, done.
Total 5 (delta 2), reused 0 (delta 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To github.com:andayalyka/HOA7.git
   b910c64..6f2bd70  main -> main
```

Description: It shows that the two created files are push to the github

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.

```
lykaandaya@managenode:~/HOA7$ ansible-playbook --ask-become-pass sites.yml
BECOME password:

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

TASK [Gathering Facts] *********************************************************
ok: [192.168.56.108]

TASK [install unzip] ***********************************************************
ok: [192.168.56.108]

TASK [install terraform] *******************************************************
changed: [192.168.56.108]

PLAY RECAP *********************************************************************
192.168.56.108             : ok=3    changed=1    unreachable=0    failed=0    s
kipped=0    rescued=0    ignored=0
```

Description: ***It verify that the terraform is installed in Ubuntu remote workstation***

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

```
lykaandaya@controlnode2:~$ terraform --version
Terraform v0.12.28

Your version of Terraform is out of date! The latest version
is 1.6.0. You can update by downloading from https://www.terraform.io/downloads
.html
```

Description: ***It display of what version of terraform was installed in the Ubuntu remote workstation***

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,

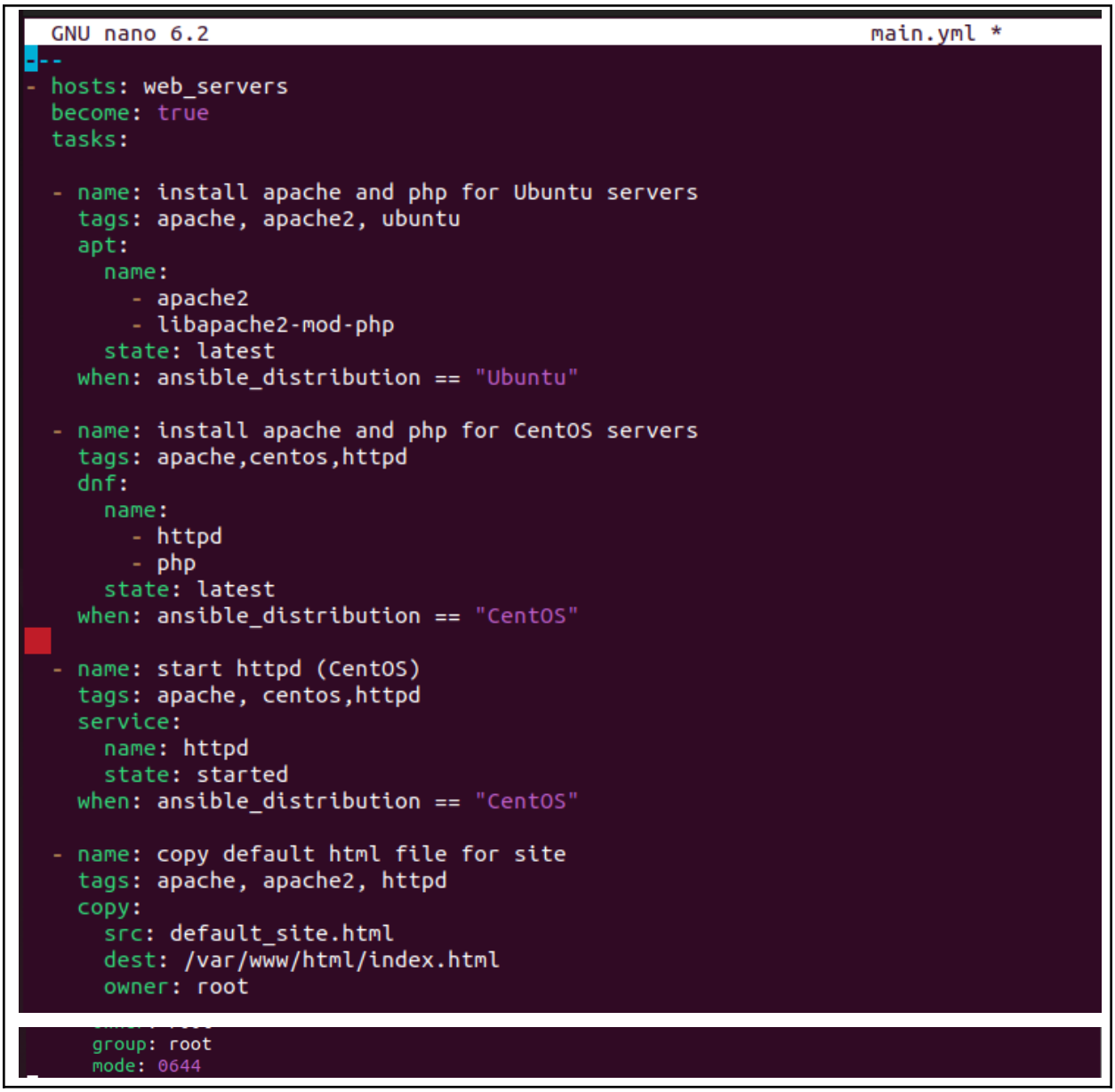
db_servers and workstations. For each directory, create a directory and name it tasks.

```
lykaandaya@managenode:~/HOA7/roles$ mkdir base
lykaandaya@managenode:~/HOA7/roles$ mkdir web_servers
lykaandaya@managenode:~/HOA7/roles$ mkdir file_servers
lykaandaya@managenode:~/HOA7/roles$ mkdir db_servers
lykaandaya@managenode:~/HOA7/roles$ mkdir workstations
lykaandaya@managenode:~/HOA7/roles$ ls
base  db_servers  file_servers  web_servers  workstations
```

```
lykaandaya@managenode:~/HOA7/roles$ cd base
lykaandaya@managenode:~/HOA7/roles/base$ mkdir tasks
lykaandaya@managenode:~/HOA7/roles/base$ cd roles
bash: cd: roles: No such file or directory
lykaandaya@managenode:~/HOA7/roles/base$ cd
lykaandaya@managenode:~$ cd HOA7
lykaandaya@managenode:~/HOA7$ cd roles
lykaandaya@managenode:~/HOA7/roles$ mkdir tasks
lykaandaya@managenode:~/HOA7/roles$ cd web_servers
lykaandaya@managenode:~/HOA7/roles/web_servers$ cd tasks
bash: cd: tasks: No such file or directory
lykaandaya@managenode:~/HOA7/roles/web_servers$ mkdir tasks
lykaandaya@managenode:~/HOA7/roles/web_servers$ cd
lykaandaya@managenode:~$ cd HOA7
lykaandaya@managenode:~/HOA7$ cd roles
lykaandaya@managenode:~/HOA7/roles$ cd file_servers
lykaandaya@managenode:~/HOA7/roles/file_servers$ mkdir tasks
lykaandaya@managenode:~/HOA7/roles/file_servers$ cd
lykaandaya@managenode:~$ cd HOA7
lykaandaya@managenode:~/HOA7$ cd roles
lykaandaya@managenode:~/HOA7/roles$ cd db_servers
lykaandaya@managenode:~/HOA7/roles/db_servers$ mkdir tasks
lykaandaya@managenode:~/HOA7/roles/db_servers$ cd
lykaandaya@managenode:~$ cd HOA7
```

```
lykaandaya@managenode:~/HOA7$ cd roles
lykaandaya@managenode:~/HOA7/roles$ cd workstations
```

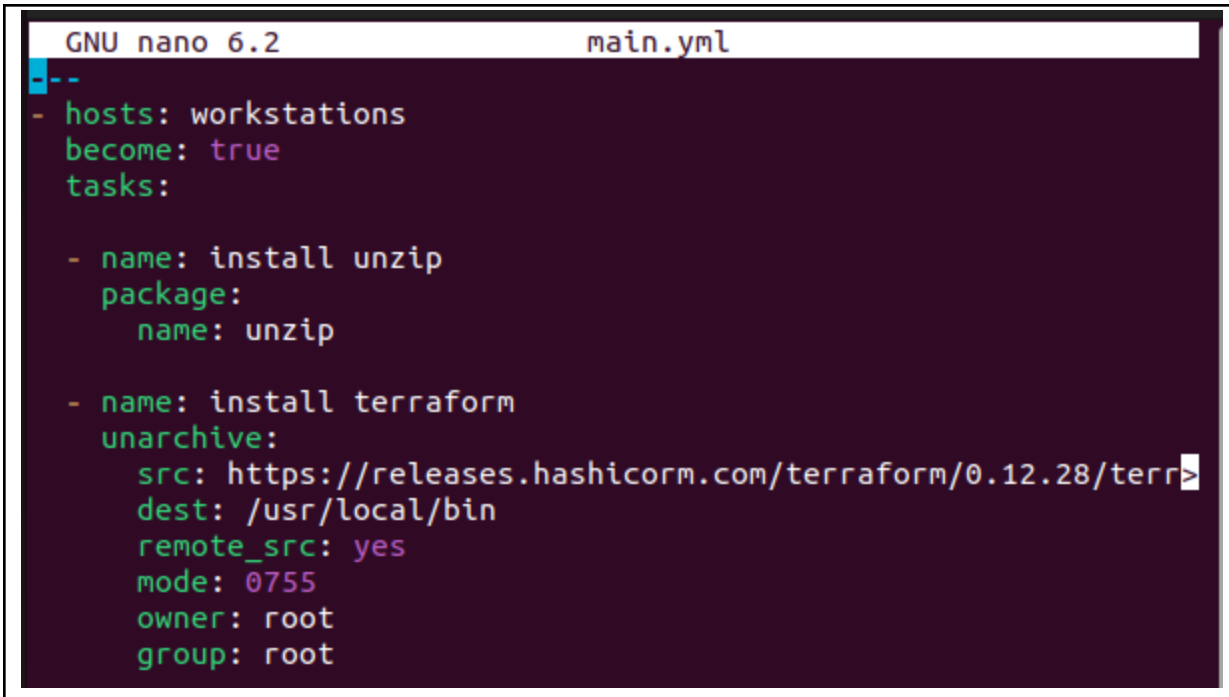
3. Go to tasks for all directory and create a file. Name it main.yml. In each of the tasks for all directories, copy and paste the code from the old site.yml file. Show all contents of main.yml files for all tasks.



```
  GNU nano 6.2                                                main.yml *
---
- hosts: web_servers
  become: true
  tasks:

  - 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
    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
```

```
  GNU nano 6.2                    main.yml
---
- hosts: db_servers
  become: true
  tasks:

  - 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"
```

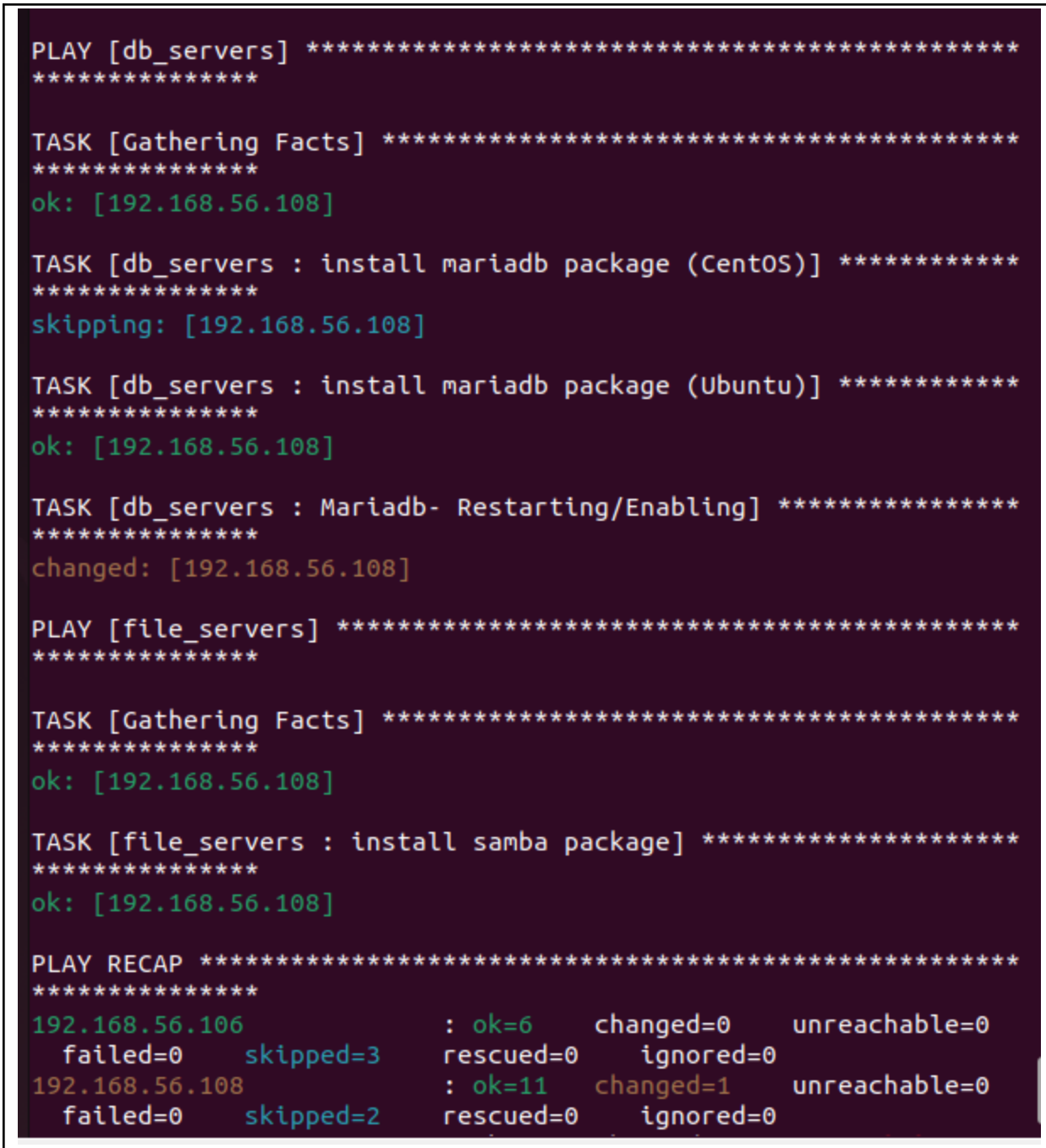
```
  GNU nano 6.2                    main.yml
---
- hosts: file_servers
  become: true
  tasks:

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



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

```
lykaandaya@managenode:~/HOA7$ ansible-playbook --ask-become-pass
sitess.yml
BECOME password:

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

TASK [Gathering Facts] *********************************************
***************
fatal: [192.168.56.110]: UNREACHABLE! => {"changed": false, "msg"
: "Failed to connect to the host via ssh: lykaandaya@192.168.56.1
10: Permission denied (publickey,gssapi-keyex,gssapi-with-mic,pas
sword).", "unreachable": true}
ok: [192.168.56.108]
ok: [192.168.56.106]

TASK [update repository index (CentOS)] ****************************
***************
skipping: [192.168.56.108]
skipping: [192.168.56.106]

TASK [install updates (Ubuntu)] ************************************
***************
ok: [192.168.56.106]
ok: [192.168.56.108]

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

TASK [Gathering Facts] *********************************************
***************
ok: [192.168.56.106]
ok: [192.168.56.108]

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

```
TASK [Gathering Facts] *********************************************
***************
ok: [192.168.56.108]

TASK [workstations : install unzip] ********************************
***************
ok: [192.168.56.108]

TASK [workstations : install terraform] ****************************
***************
ok: [192.168.56.108]

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

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

TASK [web_servers : install apache and php for Ubuntu servers] **
***************
ok: [192.168.56.106]

TASK [web_servers : install apache and php for CentOS servers] **
***************
skipping: [192.168.56.106]

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

TASK [web_servers : copy default html file for site] ************
***************
ok: [192.168.56.106]
```



Description: ***It verifies all the codes that is in the file***

Reflections:

Answer the following:

1. What is the importance of creating roles?
   - **Roles are the linchpin in upholding a fortified, well-structured, and optimally controlled Linux ecosystem. Their significance is especially**

pronounced in environments teeming with numerous users necessitating access to and collaboration with system resources. They act as the cornerstone, ensuring that permissions are judiciously allocated, security is robustly upheld, and tasks are streamlined with precision and orderliness. In essence, roles form the backbone of a resilient and harmonized Linux environment, vital for seamless collaborative work and safeguarding sensitive information.

2. What is the importance of managing files?

- **Effectively managing files in Linux is the bedrock of a secure, streamlined, and dependable computing environment. It serves as the cornerstone for optimal resource allocation, safeguarding critical data, and ensuring seamless accessibility. This practice not only fortifies the system's integrity but also contributes to heightened efficiency and trustworthiness, making it indispensable for any robust Linux operation.**

CONCLUSION:

I have learned that roles are the linchpin in upholding a fortified, well-structured, and optimally controlled Linux ecosystem. Their significance is especially pronounced in environments teeming with numerous users necessitating access to and collaboration with system resources. They act as the cornerstone, ensuring that permissions are judiciously allocated, security is robustly upheld, and tasks are streamlined with precision and orderliness. In essence, roles form the backbone of a resilient and harmonized Linux environment, vital for seamless collaborative work and safeguarding sensitive information. Also I have learned that effectively managing files in Linux is the bedrock of a secure, streamlined, and dependable computing environment. It serves as the cornerstone for optimal resource allocation, safeguarding critical data, and ensuring seamless accessibility. This practice not only fortifies the system's integrity but also contributes to heightened efficiency and trustworthiness, making it indispensable for any robust Linux operation.