Name: Jozshua Amiel Alonzo	**Date Performed: October 13, 2022**
Course/Section: CPE31S23	**Date Submitted: October 13, 2022**
Instructor: Dr. Jonathan Taylar	**Semester and SY: 2022-2023**

Activity 7: Managing Files and Creating Roles in Ansible

1. Objectives:

1.1 Manage files in remote servers

1.2 Implement roles in ansible

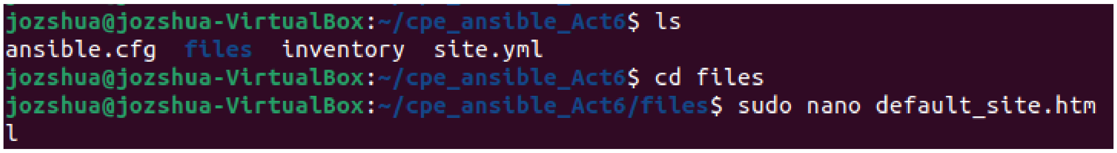
2. Discussion:

Github Link: https://github.com/jozshua/HOA7_Alonzo.git

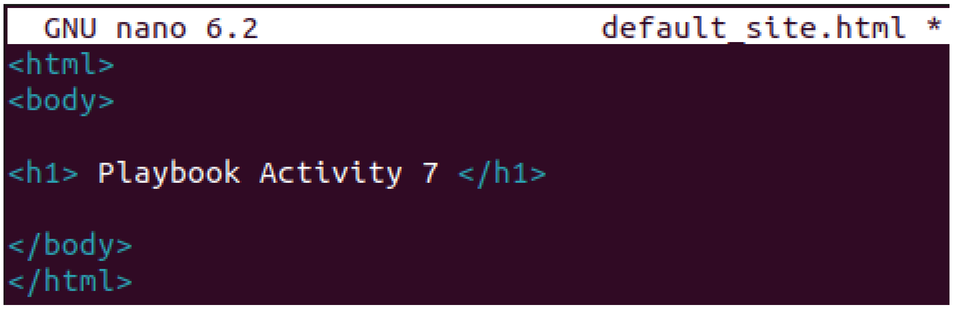
In this activity, we look at the concept of copying a file to a server. We are going to create a file into our git repository and use Ansible to grab that file and put it into a particular place so that we could do things like customize a default website, or maybe install a default configuration file.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



```
jozshua@jozshua-VirtualBox:~/cpe_ansible_Act6$ ls
ansible.cfg  files  inventory  site.yml
jozshua@jozshua-VirtualBox:~/cpe_ansible_Act6$ cd files
jozshua@jozshua-VirtualBox:~/cpe_ansible_Act6/files$ sudo nano default_site.htm
l
```

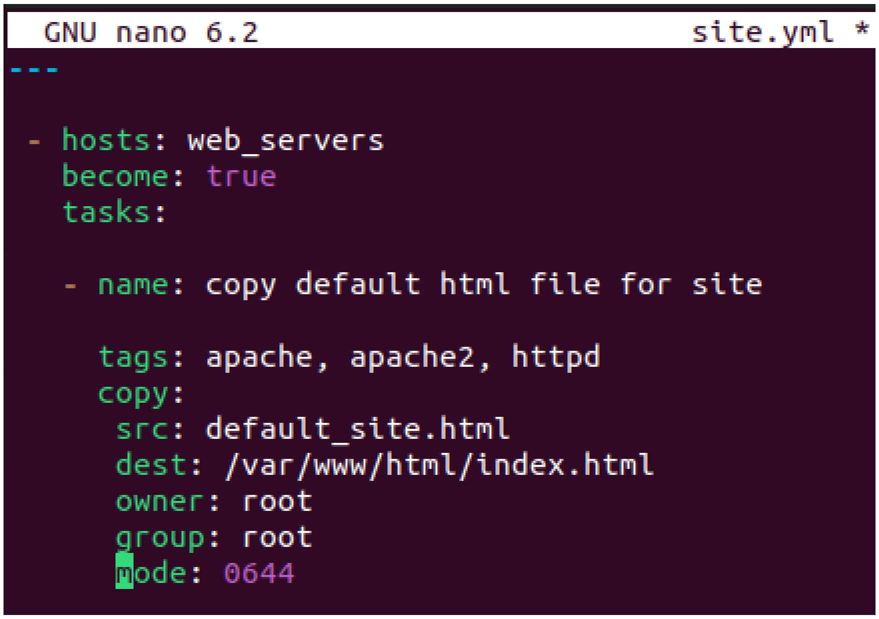


```
  GNU nano 6.2                    default_site.html *
<html>
<body>

<h1> Playbook Activity 7 </h1>

</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

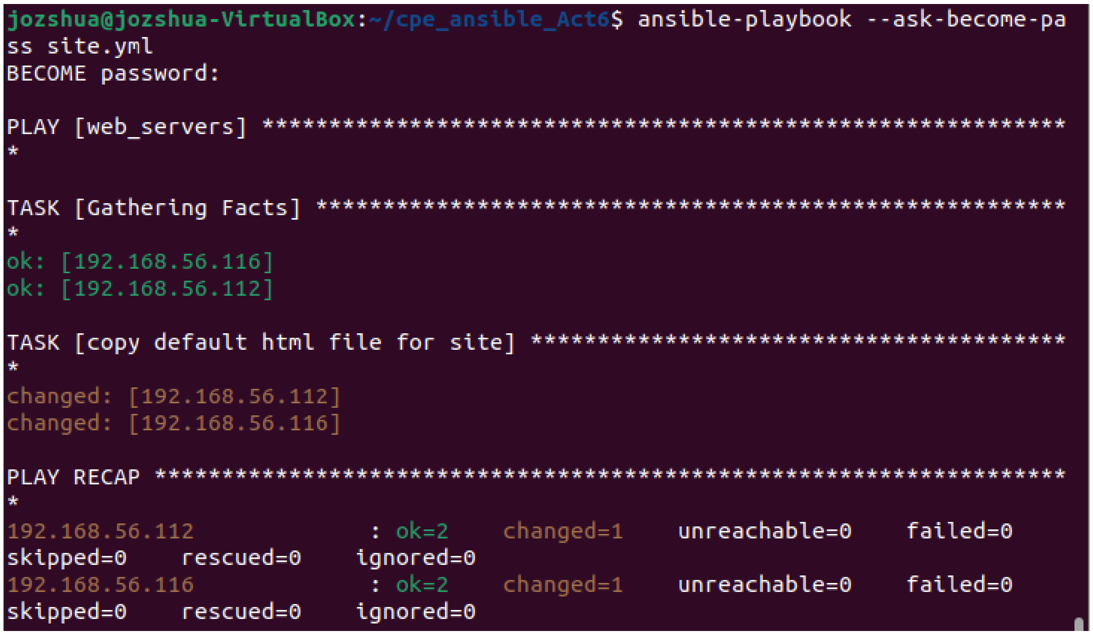
     tags: apache, apache2, httpd

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



```
  GNU nano 6.2                                     site.yml *
---

 - hosts: web_servers
   become: true
   tasks:

   - 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
```

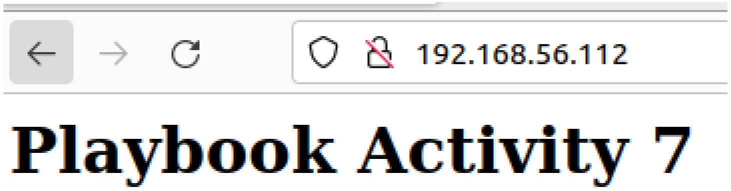
3. Run the playbook *site.yml*. Describe the changes.



```
jozshua@jozshua-VirtualBox:~/cpe_ansible_Act6$ ansible-playbook --ask-become-pa
ss site.yml
BECOME password:

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

TASK [Gathering Facts] *********************************************************
*
ok: [192.168.56.116]
ok: [192.168.56.112]

TASK [copy default html file for site] *****************************************
*
changed: [192.168.56.112]
changed: [192.168.56.116]

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

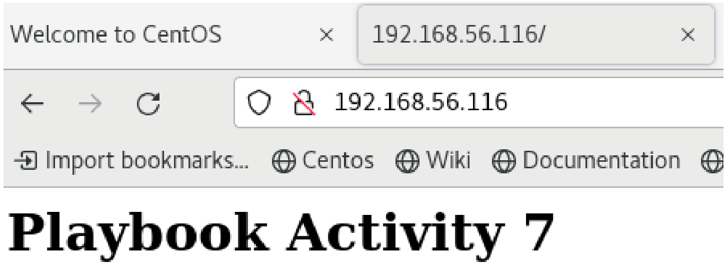
There is 1 change for each remote server from the task executed copied default html file for site.

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.



192.168.56.112

Playbook Activity 7

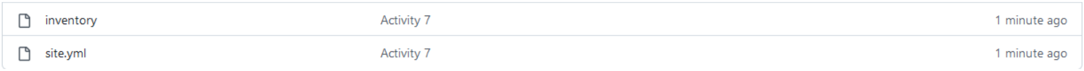


192.168.56.116

Playbook Activity 7

On each remote server it displays the text from the browser from the basic html syntax.

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



inventory	Activity 7	1 minute ago
site.yml	Activity 7	1 minute ago

The two files were added to my personal repository.

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

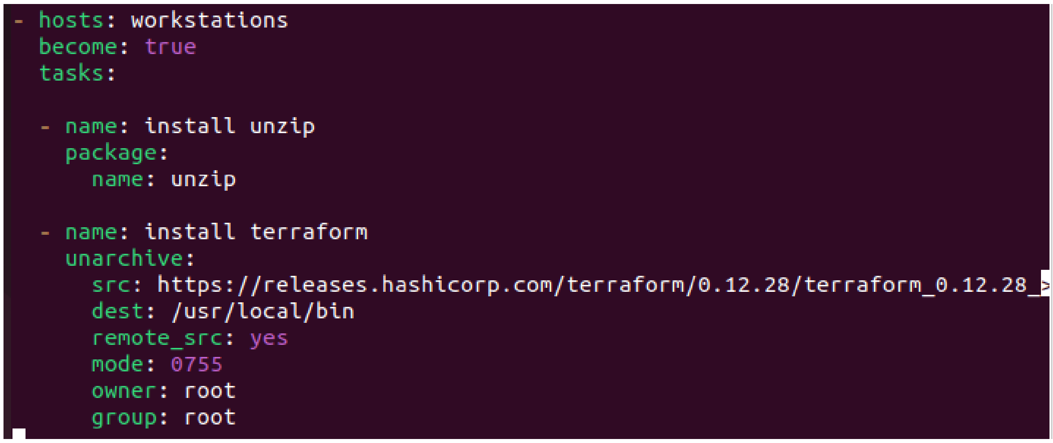
dest: /usr/local/bin

remote_src: yes

mode: 0755

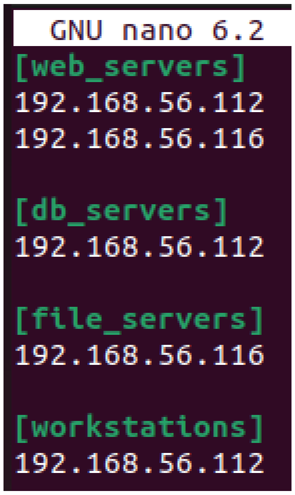
owner: root

group: root



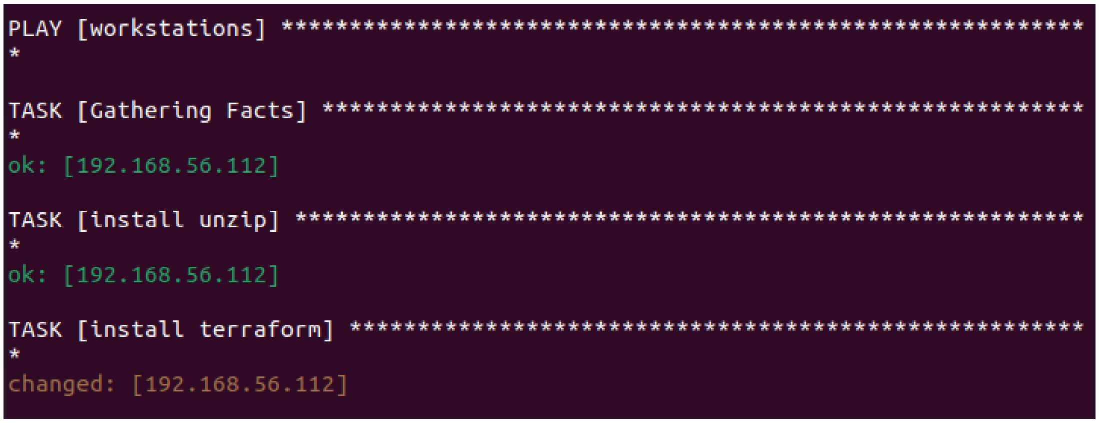
```
- hosts: workstations
  become: true
  tasks:

  - name: install unzip
    package:
      name: unzip

  - name: install terraform
    unarchive:
      src: https://releases.hashicorp.com/terraform/0.12.28/terraform_0.12.28_>
      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 6.2
[web_servers]
192.168.56.112
192.168.56.116

[db_servers]
192.168.56.112

[file_servers]
192.168.56.116

[workstations]
192.168.56.112
```

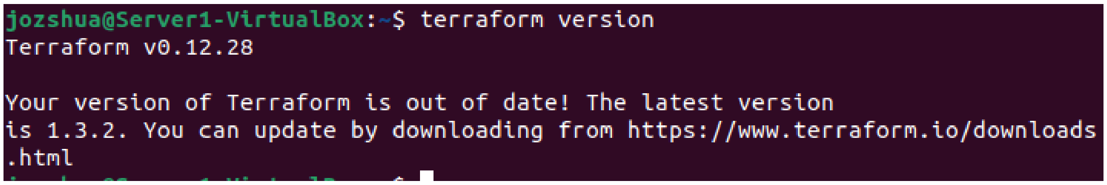
3. Run the playbook. Describe the output.



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

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

TASK [install unzip] **********************************************************
*
ok: [192.168.56.112]

TASK [install terraform] ******************************************************
*
changed: [192.168.56.112]
```

I applied it on the Ubuntu server and the 2 tasks were successfully executed and one was changed from the installation of terraform.

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

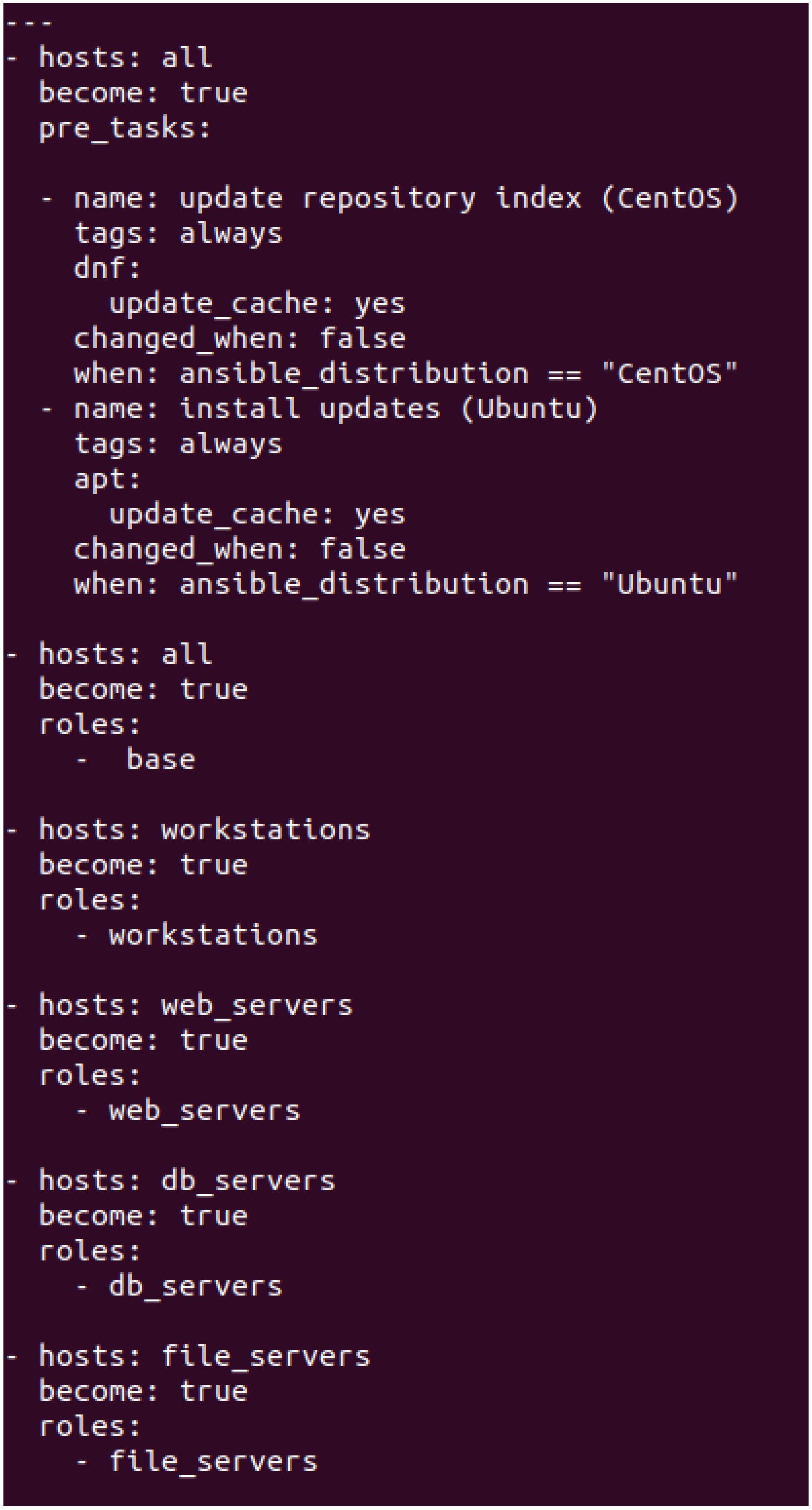


```
jozshua@Server1-VirtualBox:~$ terraform version
Terraform v0.12.28

Your version of Terraform is out of date! The latest version
is 1.3.2. You can update by downloading from https://www.terraform.io/downloads
.html
```

It showed the terraform version and the version is not the latest version.

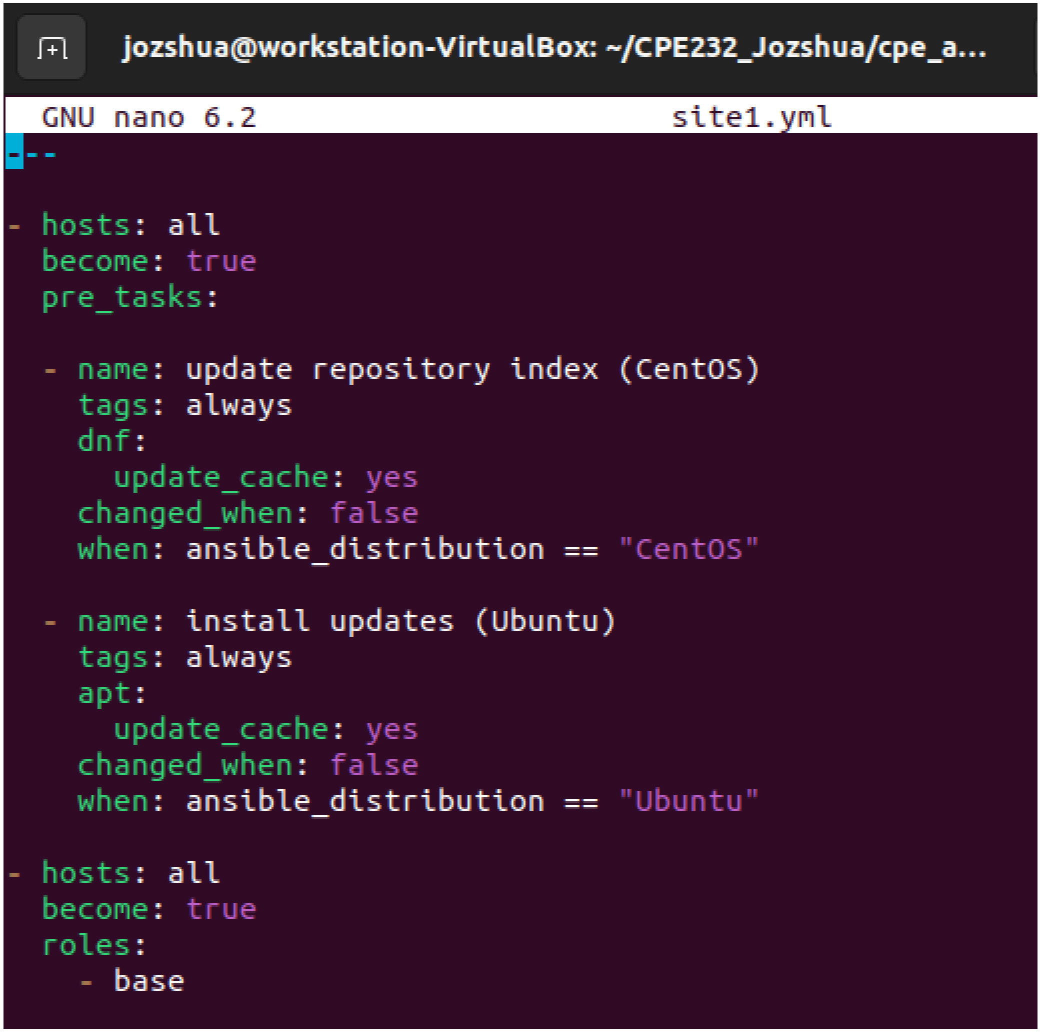
Task 3: Create roles

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)



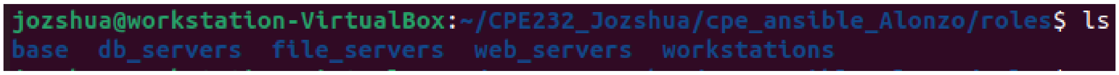
```
---
- hosts: all
  become: true
  pre_tasks:

  - 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.

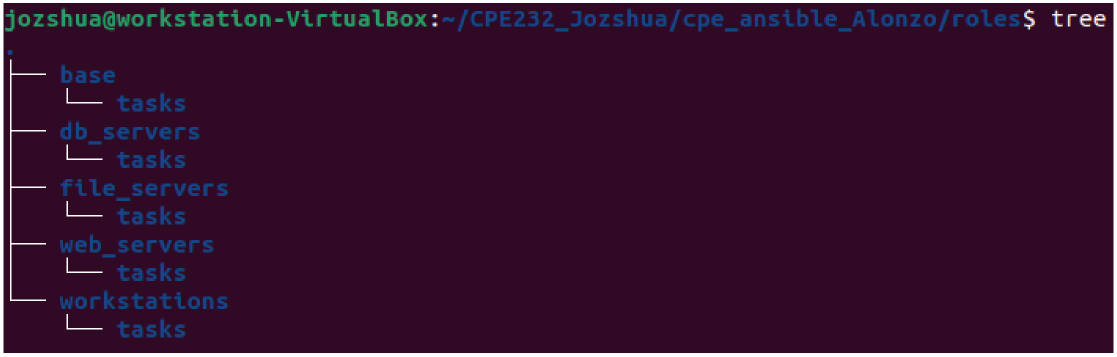


```
  GNU nano 6.2                          site1.yml
---

- 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
```

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.



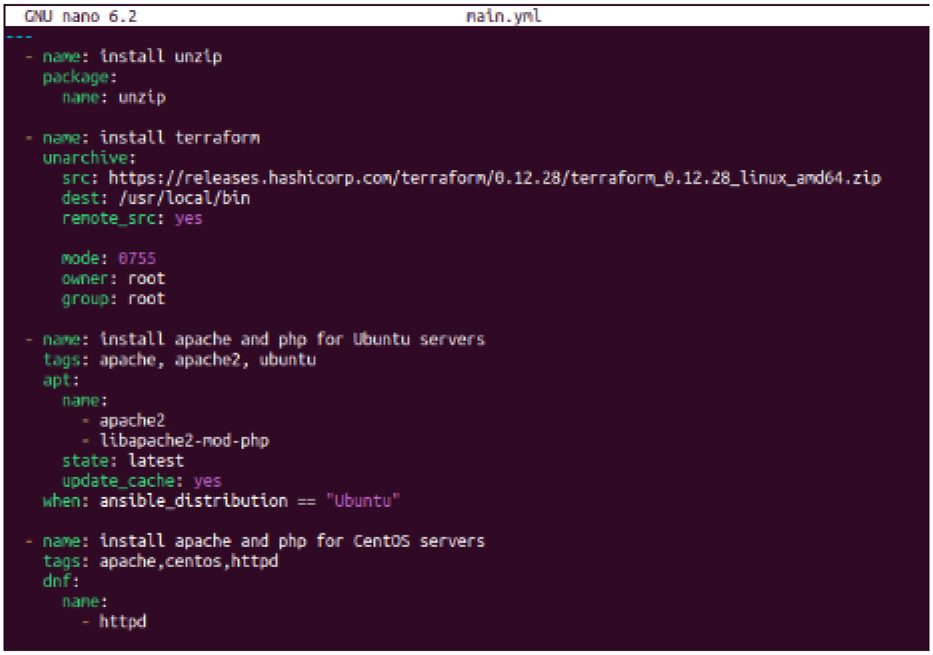
```
jozshua@workstation-VirtualBox:~/CPE232_Jozshua/cpe_ansible_Alonzo/roles$ ls
base  db_servers  file_servers  web_servers  workstations
```



```
jozshua@workstation-VirtualBox:~/CPE232_Jozshua/cpe_ansible_Alonzo/roles$ tree
.
├── base
│   └── tasks
├── db_servers
│   └── tasks
├── file_servers
│   └── tasks
├── web_servers
│   └── tasks
└── workstations
    └── tasks
```

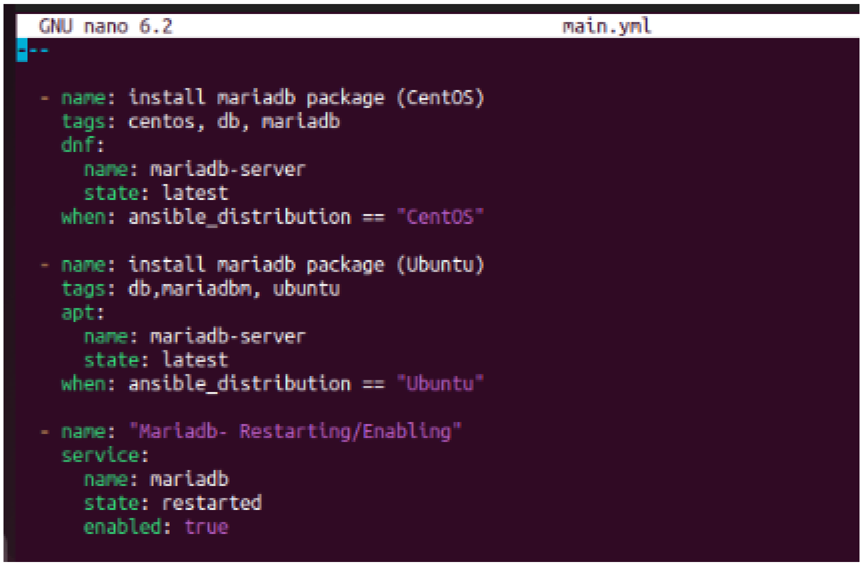
3. Go to tasks for all directory and create a file. Name it main.yml. In each of the tasks for all directories, copy and paste the code from the old site.yml file. Show all contents of main.yml files for all tasks.

Base:



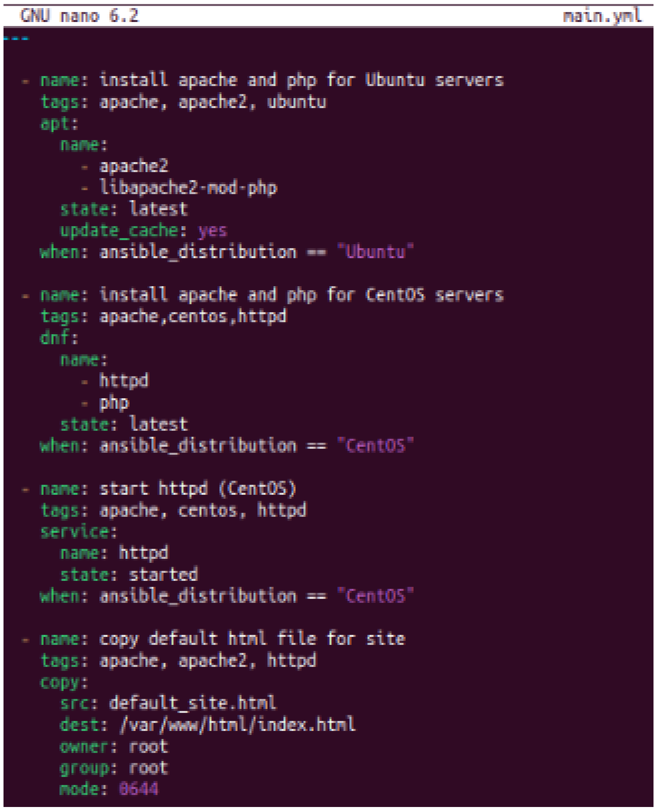
```
  GNU nano 6.2                                main.yml
---
  - name: install unzip
    package:
      name: unzip

  - 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

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

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

db_servers:



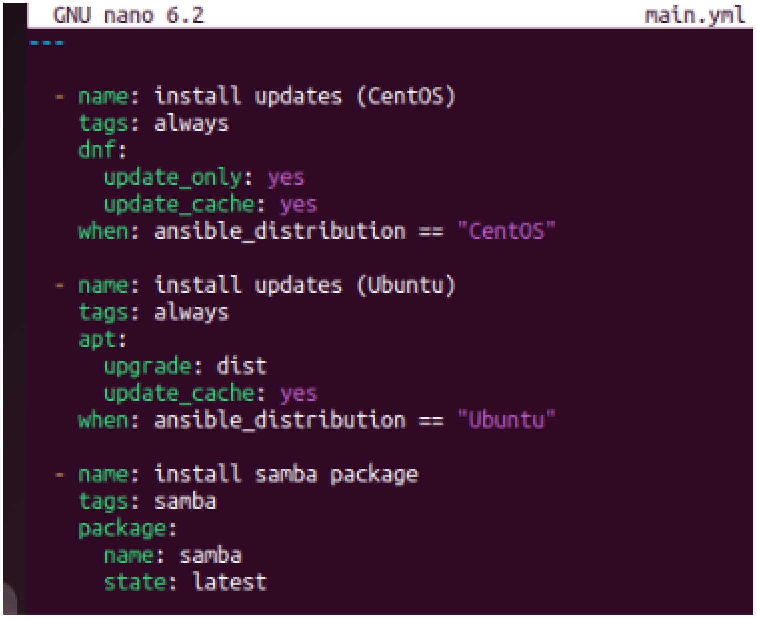
```
GNU nano 6.2                                          main.yml
---

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

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

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

web_servers:



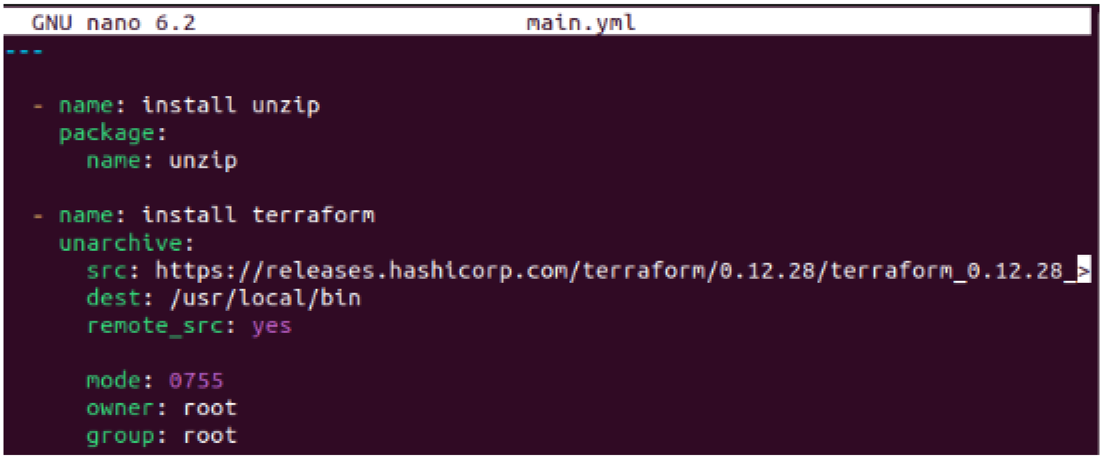
```
GNU nano 6.2                                               main.yml
---

  - name: install apache and php for Ubuntu servers
    tags: apache, apache2, ubuntu
    apt:
      name:
        - apache2
        - libapache2-mod-php
      state: latest
      update_cache: yes
    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
```

file_servers:



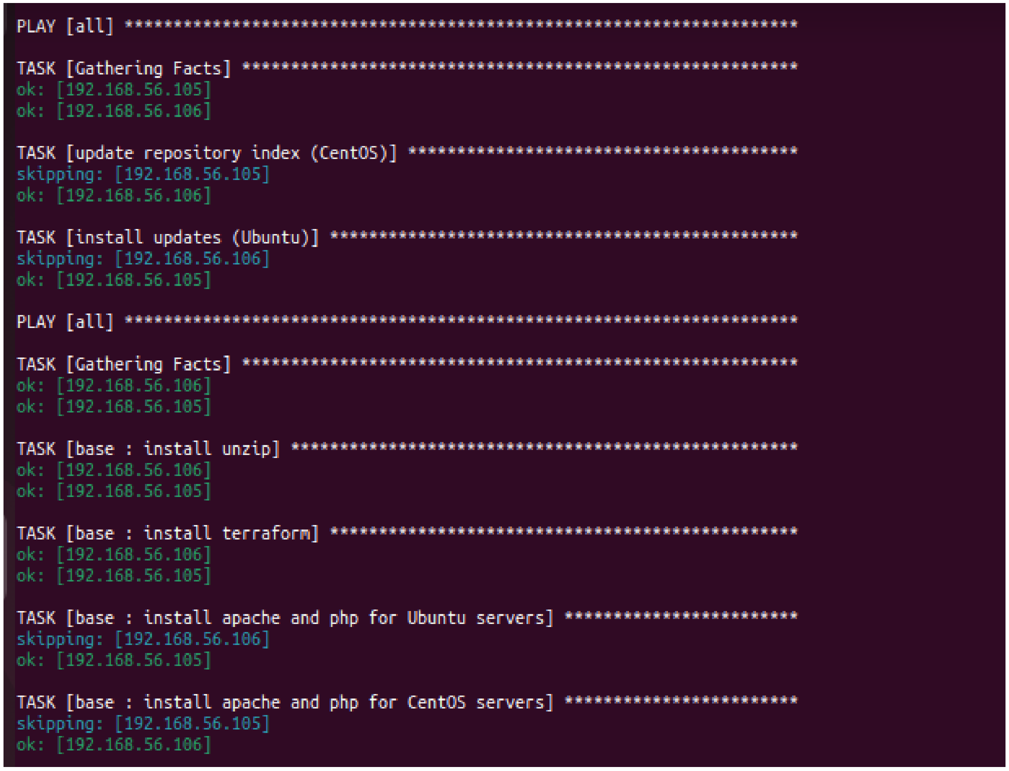
```
 GNU nano 6.2                               main.yml
---

 - name: install updates (CentOS)
   tags: always
   dnf:
     update_only: yes
     update_cache: yes
   when: ansible_distribution == "CentOS"

 - name: install updates (Ubuntu)
   tags: always
   apt:
     upgrade: dist
     update_cache: yes
   when: ansible_distribution == "Ubuntu"

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

workstations:

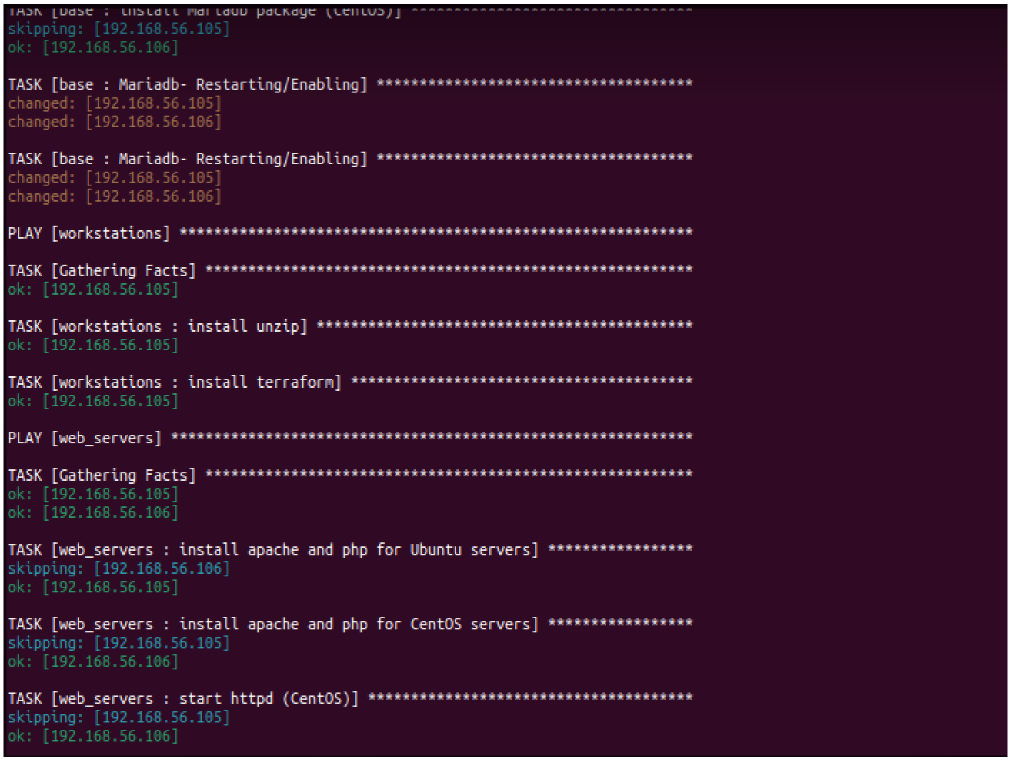


```
 GNU nano 6.2                              main.yml
---

 - name: install unzip
   package:
     name: unzip

 - name: install terraform
   unarchive:
     src: https://releases.hashicorp.com/terraform/0.12.28/terraform_0.12.28_>
     dest: /usr/local/bin
     remote_src: yes

     mode: 0755
     owner: root
     group: root
```

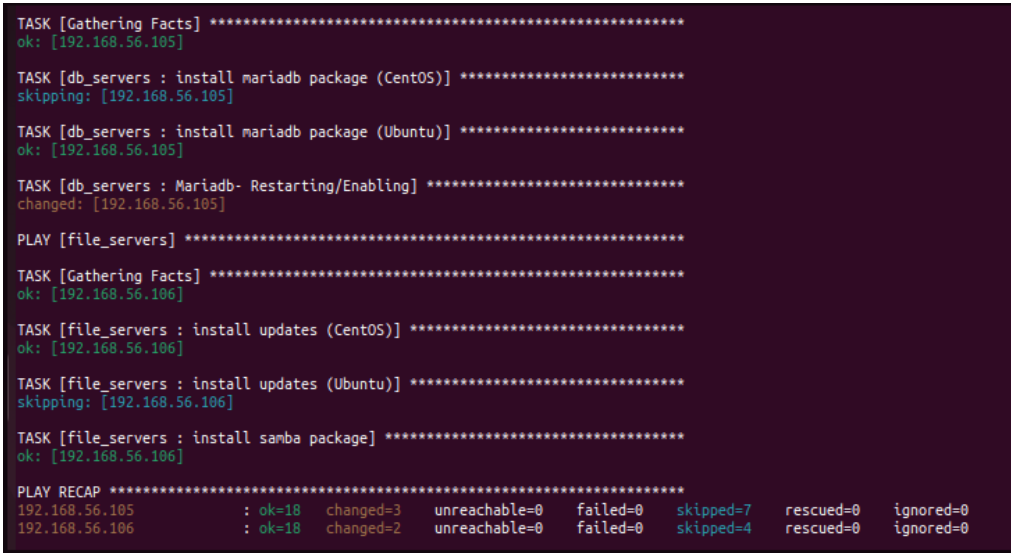
4. Run the site.yml playbook and describe the output.



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

TASK [Gathering Facts] **********************************************************
ok: [192.168.56.105]
ok: [192.168.56.106]

TASK [update repository index (CentOS)] *****************************************
skipping: [192.168.56.105]
ok: [192.168.56.106]

TASK [install updates (Ubuntu)] *************************************************
skipping: [192.168.56.106]
ok: [192.168.56.105]

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

TASK [Gathering Facts] **********************************************************
ok: [192.168.56.106]
ok: [192.168.56.105]

TASK [base : install unzip] *****************************************************
ok: [192.168.56.106]
ok: [192.168.56.105]

TASK [base : install terraform] *************************************************
ok: [192.168.56.106]
ok: [192.168.56.105]

TASK [base : install apache and php for Ubuntu servers] *************************
skipping: [192.168.56.106]
ok: [192.168.56.105]

TASK [base : install apache and php for CentOS servers] *************************
skipping: [192.168.56.105]
ok: [192.168.56.106]
```



```
TASK [base : install mariadb package (CentOS)] **********************************
skipping: [192.168.56.105]
ok: [192.168.56.106]

TASK [base : Mariadb- Restarting/Enabling] **************************************
changed: [192.168.56.105]
changed: [192.168.56.106]

TASK [base : Mariadb- Restarting/Enabling] **************************************
changed: [192.168.56.105]
changed: [192.168.56.106]

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

TASK [Gathering Facts] **********************************************************
ok: [192.168.56.105]

TASK [workstations : install unzip] *********************************************
ok: [192.168.56.105]

TASK [workstations : install terraform] *****************************************
ok: [192.168.56.105]

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

TASK [Gathering Facts] **********************************************************
ok: [192.168.56.105]
ok: [192.168.56.106]

TASK [web_servers : install apache and php for Ubuntu servers] ******************
skipping: [192.168.56.106]
ok: [192.168.56.105]

TASK [web_servers : install apache and php for CentOS servers] ******************
skipping: [192.168.56.105]
ok: [192.168.56.106]

TASK [web_servers : start httpd (CentOS)] ***************************************
skipping: [192.168.56.105]
ok: [192.168.56.106]
```



```
TASK [Gathering Facts] *********************************************************
ok: [192.168.56.105]

TASK [db_servers : install mariadb package (CentOS)] ***************************
skipping: [192.168.56.105]

TASK [db_servers : install mariadb package (Ubuntu)] ***************************
ok: [192.168.56.105]

TASK [db_servers : Mariadb- Restarting/Enabling] *******************************
changed: [192.168.56.105]

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

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

TASK [file_servers : install updates (CentOS)] *********************************
ok: [192.168.56.106]

TASK [file_servers : install updates (Ubuntu)] *********************************
skipping: [192.168.56.106]

TASK [file_servers : install samba package] ************************************
ok: [192.168.56.106]

PLAY RECAP *********************************************************************
192.168.56.105             : ok=18   changed=3    unreachable=0    failed=0    skipped=7    rescued=0    ignored=0
192.168.56.106             : ok=18   changed=2    unreachable=0    failed=0    skipped=4    rescued=0    ignored=0
```

For running this playbook, each task was successfully executed because it has no error in it. From these 2 servers with different tasks from the site.yml there are 3 changed states from the Ubuntu server and 2 changed states from the Centos server. The skipped state is from the different tasks for the only server that is designated.

Reflections:

Answer the following:

1. What is the importance of creating roles? It is important because it gives a framework for the collections of variables, tasks, files, templates, and modules. Also, it could distribute a playbook into multiple files.

2. What is the importance of managing files? I think it is important because it is a practice for storing and managing files in an organized manner.