Name: Buenvenida, Ken Benedict D.	**Date Performed: 10/09/23**
Course/Section:CpE-232 - CpE31S4	**Date Submitted: 10/09/23**
Instructor: Dr. Jonathan Taylar	**Semester and SY: 1ST Semester 2023 - 2024**

Activity 7: Managing Files and Creating Roles in Ansible

1. **Objectives:**

1.1 Manage files in remote servers

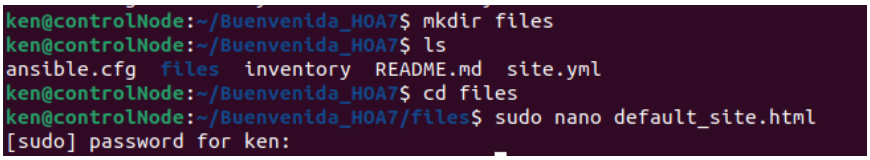
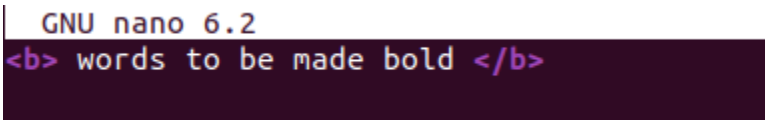
1.2 Implement roles in ansible

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.

INPUT	
OUTPUT	

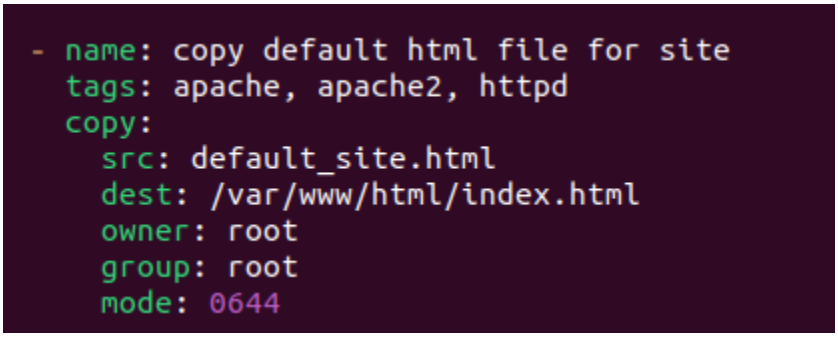
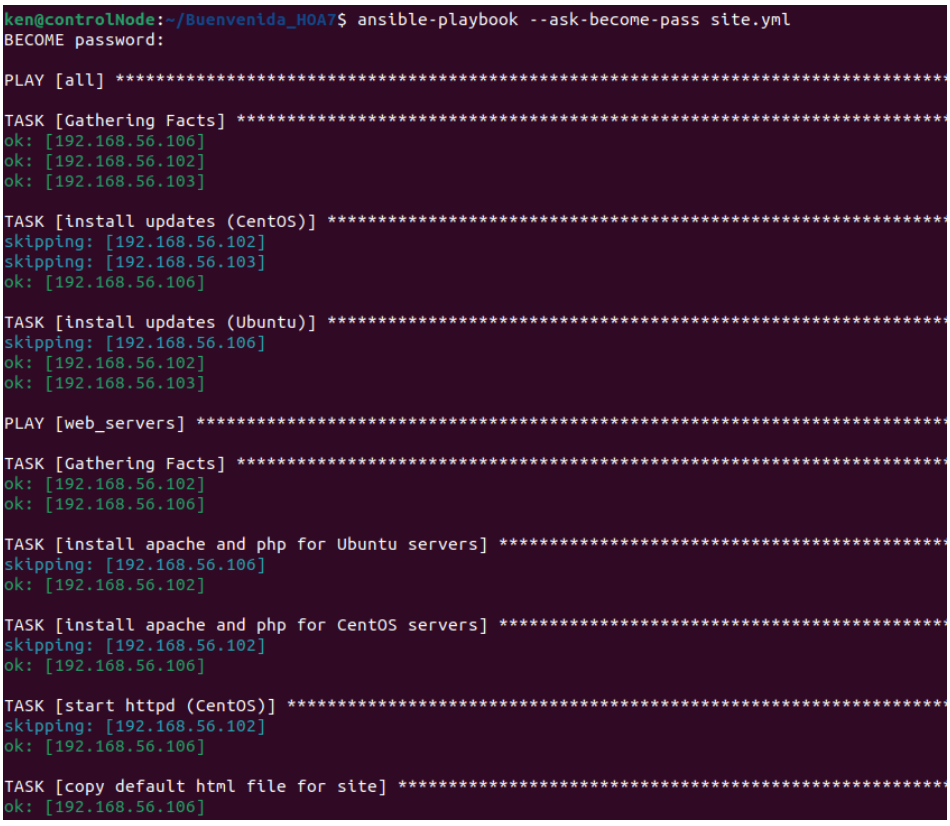
- I created a directory called files and created a .html file inside the directory that contains an html syntax that makes the text bold.

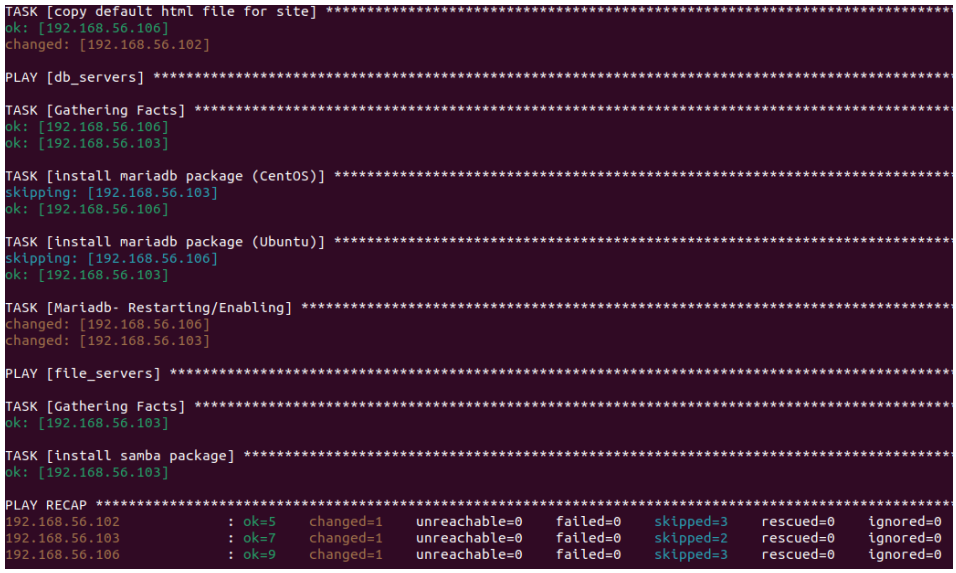
2. Edit the *site.yml* file and just below the *web_servers* play, create a new file to copy the default html file for site:
   - name: copy default html file for site

     tags: apache, apache2, httpd
     copy:

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

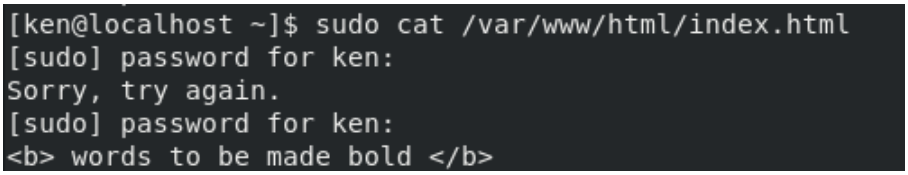
3. Run the playbook *site.yml*. Describe the changes.

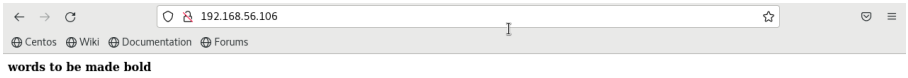
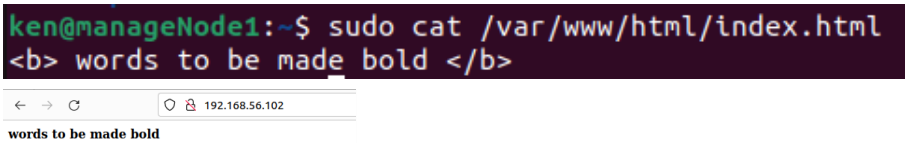
INPUT	
PROCESS	
OUTPUT	



- I edited the site.yml for it to copy the .html file that I have inputted in the site.yml, It had successfully copied the .html files into the remote servers as shown above.

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.

CENTOS	

	
Ubuntu Server 1	

- I was able to display the output properly that I have added in the html file that I had copied from the local machine to the remote servers.

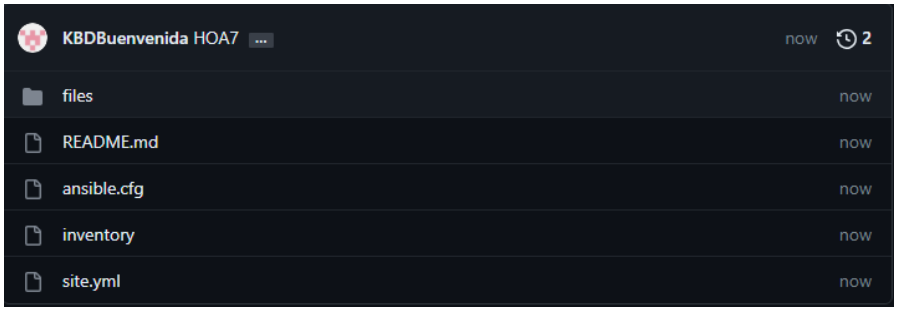
5. Sync your local repository with GitHub and describe the changes.

INPUT

```
ken@controlNode:~/Buenvenida_HOA7$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   README.md

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        ansible.cfg
        files/
        inventory
        site.yml
```

```
ken@controlNode:~/Buenvenida_HOA7$ git add *
ken@controlNode:~/Buenvenida_HOA7$ git commit -m "HOA7"
[main 740644e] HOA7
 5 files changed, 111 insertions(+), 1 deletion(-)
 create mode 100644 ansible.cfg
 create mode 100644 files/default_site.html
 create mode 100644 inventory
 create mode 100644 site.yml
ken@controlNode:~/Buenvenida_HOA7$ git push origin
Enumerating objects: 10, done.
Counting objects: 100% (10/10), done.
Compressing objects: 100% (5/5), done.
Writing objects: 100% (8/8), 1.16 KiB | 1.16 MiB/s, done.
Total 8 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:KBDBuenvenida/Buenvenida_HOA7.git
   194ff2d..740644e  main -> main
```

OUTPUT



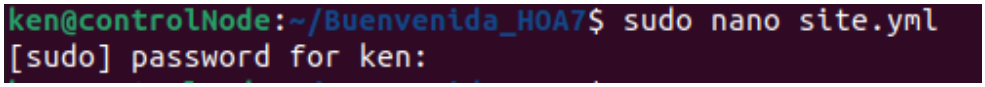
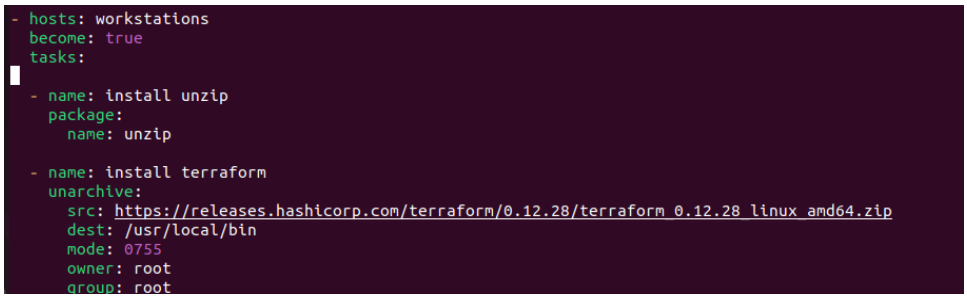
- I was able to update the repository earlier to the new changes in the repository.

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

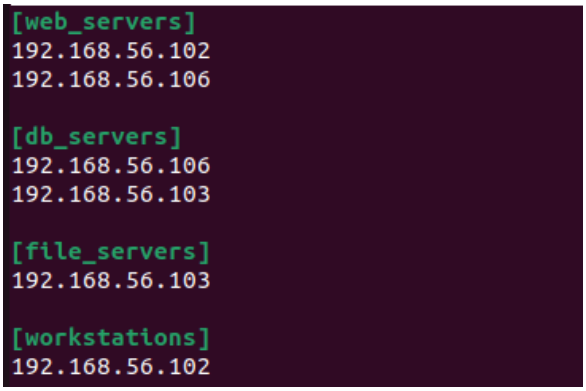
1. Edit the site.yml. Just before the web_servers play, create a new play:
   - hosts: workstations
     become: true
     tasks:

     - 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

INPUT	
OUTPUT	

2. Edit the inventory file and add workstations group. Add any Ubuntu remote server. Make sure to remember the IP address.

INPUT	
OUTPUT	

3. Run the playbook. Describe the output.

INPUT

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

OUTPUT

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

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

TASK [install unzip] ***********************************************************************************************************
ok: [192.168.56.102]

TASK [install terraform] *******************************************************************************************************
changed: [192.168.56.102]

PLAY RECAP *********************************************************************************************************************
192.168.56.102             : ok=8    changed=1    unreachable=0    failed=0    skipped=3    rescued=0    ignored=0
192.168.56.103             : ok=7    changed=1    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
192.168.56.106             : ok=9    changed=1    unreachable=0    failed=0    skipped=3    rescued=0    ignored=0
```

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

OUTPUT

```
ken@manageNode1:~$ terraform -v
Terraform v1.6.0-dev
on linux_amd64
```

- I was able to unzip properly and install the terraform properly to my S1 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.

```
---

- 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: workstations
  become: true
  roles:
    - workstations

- hosts: web_servers
  become: true
  roles:
    - web_servers

- hosts: db_servers
  become: true
  roles:
    - db_servers

- hosts: file_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.

```
ken@controlNode:~/Buenvenida_HOA7$ mkdir roles
ken@controlNode:~/Buenvenida_HOA7$ cd roles
ken@controlNode:~/Buenvenida_HOA7/roles$ mkdir base
ken@controlNode:~/Buenvenida_HOA7/roles$ mkdir web_servers
ken@controlNode:~/Buenvenida_HOA7/roles$ mkdir file_servers
ken@controlNode:~/Buenvenida_HOA7/roles$ mkdir db_servers
ken@controlNode:~/Buenvenida_HOA7/roles$ mkdir workstations
ken@controlNode:~/Buenvenida_HOA7/roles$ ls
base  db_servers  file_servers  web_servers  workstations
```

```
ken@controlNode:~/Buenvenida_HOA7/roles$ cd base
ken@controlNode:~/Buenvenida_HOA7/roles/base$ mkdir tasks
ken@controlNode:~/Buenvenida_HOA7/roles/base$ cd db_servers
bash: cd: db_servers: No such file or directory
ken@controlNode:~/Buenvenida_HOA7/roles/base$ cd ..
ken@controlNode:~/Buenvenida_HOA7/roles$ cd db_servers
ken@controlNode:~/Buenvenida_HOA7/roles/db_servers$ mkdir tasks
ken@controlNode:~/Buenvenida_HOA7/roles/db_servers$ cd ..
ken@controlNode:~/Buenvenida_HOA7/roles$ cd file_servers
ken@controlNode:~/Buenvenida_HOA7/roles/file_servers$ mkdir tasks
ken@controlNode:~/Buenvenida_HOA7/roles/file_servers$ cd ..
ken@controlNode:~/Buenvenida_HOA7/roles$ cd web_servers
ken@controlNode:~/Buenvenida_HOA7/roles/web_servers$ mkdir tasks
ken@controlNode:~/Buenvenida_HOA7/roles/web_servers$ cd ..
ken@controlNode:~/Buenvenida_HOA7/roles$ cd workstations
ken@controlNode:~/Buenvenida_HOA7/roles/workstations$ mkdir tasks
ken@controlNode:~/Buenvenida_HOA7/roles/workstations$ ls
tasks
```

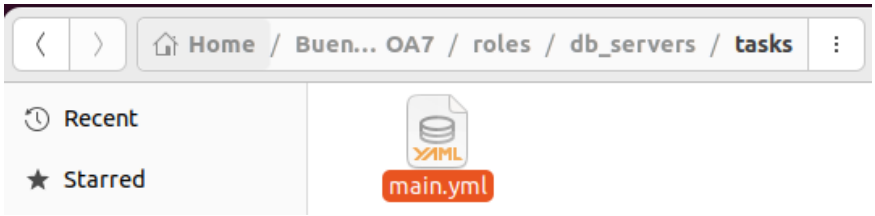
3. Go to tasks for all directory and create a file. Name it main.yml. In each of the tasks for all directories, copy and paste the code from the old site.yml file. Show all contents of main.yml files for all tasks.

```
ken@controlNode:~/Buenvenida_HOA7/roles/workstations$ touch main.yml
ken@controlNode:~/Buenvenida_HOA7/roles/workstations$ sudo nano main.yml
ken@controlNode:~/Buenvenida_HOA7/roles/workstations$ cd ..
ken@controlNode:~/Buenvenida_HOA7/roles$ cd base
ken@controlNode:~/Buenvenida_HOA7/roles/base$ touch main.yml
ken@controlNode:~/Buenvenida_HOA7/roles/base$ sudo nano main.yml
ken@controlNode:~/Buenvenida_HOA7/roles/base$ cd ..
ken@controlNode:~/Buenvenida_HOA7/roles$ cd db_servers
ken@controlNode:~/Buenvenida_HOA7/roles/db_servers$ touch main.yml
ken@controlNode:~/Buenvenida_HOA7/roles/db_servers$ sudo nano main.yml
ken@controlNode:~/Buenvenida_HOA7/roles/db_servers$ cd ..
ken@controlNode:~/Buenvenida_HOA7/roles$ cd file_servers
ken@controlNode:~/Buenvenida_HOA7/roles/file_servers$ touch main.yml
ken@controlNode:~/Buenvenida_HOA7/roles/file_servers$ sudo nano main.yml
ken@controlNode:~/Buenvenida_HOA7/roles/file_servers$ cd ..
ken@controlNode:~/Buenvenida_HOA7/roles$ cd web_servers
ken@controlNode:~/Buenvenida_HOA7/roles/web_servers$ touch main.yml
ken@controlNode:~/Buenvenida_HOA7/roles/web_servers$ sudo nano main.yml
ken@controlNode:~/Buenvenida_HOA7/roles/web_servers$ cd ..
ken@controlNode:~/Buenvenida_HOA7/roles$ cd ..
```

BASE

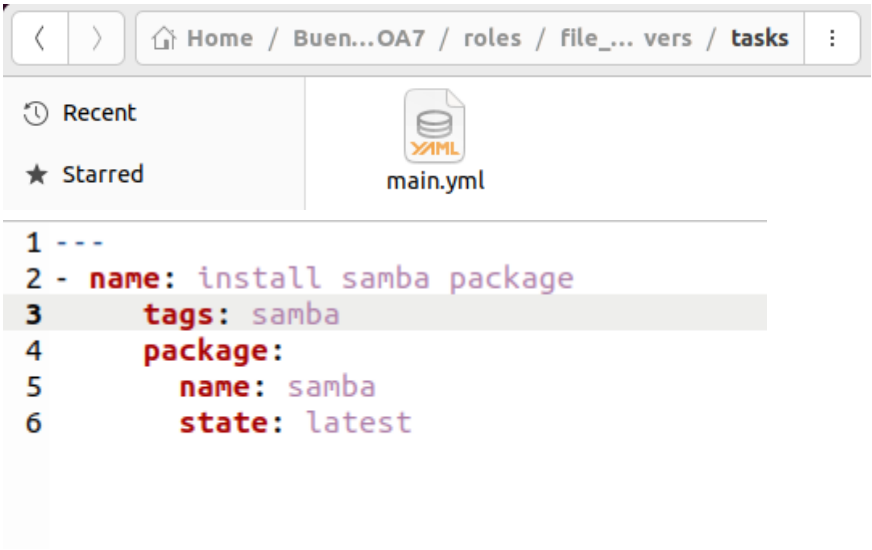
```yaml
---
- name: install updates (CentOS)
  tags: always
  dnf:
    update_only: yes
    update_cache: yes
    use_backend: dnf4
  when: ansible_distribution == "CentOS"

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

```

DB_SERVERS



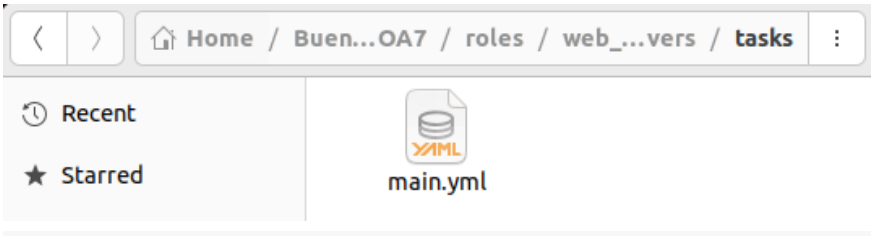

```yaml
---
- name: install mariadb package (CentOS)
  tags: centos,db,mariadb
  dnf:
    name: mariadb-serer
    state: latest
    use_backend: dnf4
  when: ansible_distribution == "CentOS"

- 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

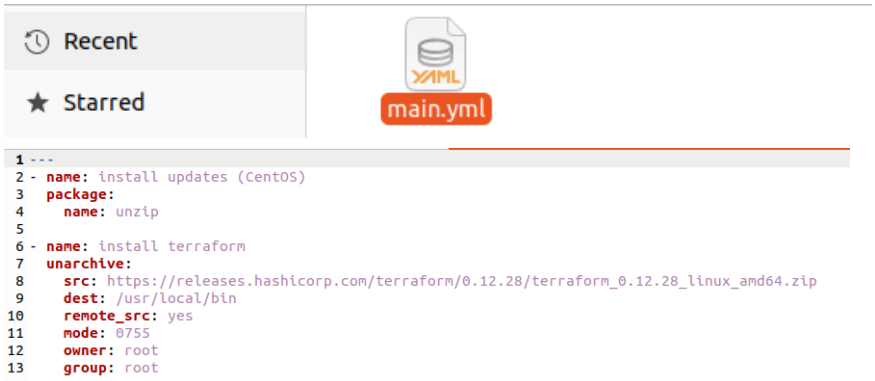



```yaml
---
- name: install samba package
    tags: samba
    package:
      name: samba
      state: latest
```

WEB_SERVERS




```yaml
---
- 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"
```

WORKSTATIONS	

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

INPUT	`ken@controlNode:~/Buenvenida_HOA7$ ansible-playbook --ask-become-pass site.yml` `BECOME password:`
PROCESS	

OUTPUT	`PLAY [all] ***********`
	
	`TASK [Gathering Facts] ***********`
	`ok: [192.168.56.103]`
	`ok: [192.168.56.102]`
	`ok: [192.168.56.106]`
	
	`TASK [update repository index (CentOS)] ***********`
	`skipping: [192.168.56.102]`
	`skipping: [192.168.56.103]`
	`ok: [192.168.56.106]`
	
	`TASK [install updates (Ubuntu)] ***********`
	`skipping: [192.168.56.106]`
	`ok: [192.168.56.102]`
	`ok: [192.168.56.103]`
	
	`PLAY [all] ***********`
	
	`TASK [Gathering Facts] ***********`
	`ok: [192.168.56.103]`
	`ok: [192.168.56.102]`
	`ok: [192.168.56.106]`
	
	`TASK [base : install updates (CentOS)] ***********`
	`skipping: [192.168.56.102]`
	`skipping: [192.168.56.103]`
	`ok: [192.168.56.106]`
	
	`TASK [base : install updates (Ubuntu)] ***********`
	`skipping: [192.168.56.106]`
	`ok: [192.168.56.103]`
	`ok: [192.168.56.102]`
	
	`PLAY [workstations] ***********`
	
	`TASK [Gathering Facts] ***********`
	`ok: [192.168.56.102]`

```
TASK [workstations : install updates (CentOS)] *********************************
ok: [192.168.56.102]

TASK [workstations : install terraform] ****************************************
ok: [192.168.56.102]

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

TASK [Gathering Facts] *********************************************************
ok: [192.168.56.102]
ok: [192.168.56.106]

TASK [web_servers : install apache and php for Ubuntu servers] *****************
skipping: [192.168.56.106]
ok: [192.168.56.102]

TASK [web_servers : install apache and php for CentOS servers] *****************
skipping: [192.168.56.102]
ok: [192.168.56.106]

TASK [web_servers : start httpd (CentOS)] **************************************
skipping: [192.168.56.102]
ok: [192.168.56.106]

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

TASK [Gathering Facts] *********************************************************
ok: [192.168.56.103]
ok: [192.168.56.106]

TASK [db_servers : install mariadb package (CentOS)] ***************************
skipping: [192.168.56.103]
ok: [192.168.56.106]

TASK [db_servers : install mariadb package (Ubuntu)] ***************************
skipping: [192.168.56.106]
ok: [192.168.56.103]

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

```
TASK [db_servers : Mariadb- Restarting/Enabling] ********************************************************************
changed: [192.168.56.103]
changed: [192.168.56.106]

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

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

TASK [file_servers : install samba package] *************************************************************************
ok: [192.168.56.103]

PLAY RECAP **********************************************************************************************************
192.168.56.102             : ok=9    changed=0    unreachable=0    failed=0    skipped=4    rescued=0    ignored=0
192.168.56.103             : ok=9    changed=1    unreachable=0    failed=0    skipped=3    rescued=0    ignored=0
192.168.56.106             : ok=10   changed=1    unreachable=0    failed=0    skipped=4    rescued=0    ignored=0
```

- I was able to run the modified version of “site.yml” by giving roles to the playbook. As you can see in the output it played role by role according to the playbook.

Reflections:

Answer the following:

1. What is the importance of creating roles?
    - The importance of creating roles is to make the playbook run smoothly and easier since it is much easier to share and modify them. It also reduces the time complexity of the playbook and also minimizes the syntax needed for a playbook. Roles play a big part in organizing and managing task automatically and effectively.

2. What is the importance of managing files?

   - Managing files in Ansible is very important for automatic configuration tasks that ensures consistency across servers and it also enhances its security and reduces the risk of errors occuring. It also provides the flexibility to distribute files and resources efficiently.

Conclusion

- In conclusion, I was able to accomplish the objectives of this activity which are managing files and creating roles in Ansible where I can use it in a remote or local machine to implement roles. I was able to copy the basic .html file that I had created and display its output on the remote servers. I was also able to install Terraform properly.