Name: Vincent Bryan Bose	Date Performed: 10/07/2024
Course/Section: CPE31S2	Date Submitted: 10/07/2024
Instructor: Engr. Robin Valenzuela	Semester and SY: 2024-2025

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.

```
  GNU nano 2.9.3                    default_site.html

<!DOCTYPE html>
<html>

<head>
  <title>I LOVE</title>
</head>

<body>
 <h2>U</h2>
</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

```
- 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.

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

<head>
  <title>I LOVE</title>
</head>

<body>
 <h2>U</h2>
</body>

</html>
```

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

```
vbbose@workstation:~/BOSE-HOA-7.1$ ls
ansible.cfg  files  inventory  README.md  site.yml
vbbose@workstation:~/BOSE-HOA-7.1$ git add ansible.cfg files inventory site.yml
vbbose@workstation:~/BOSE-HOA-7.1$ git commit -m "DONE"
[main 3152578] DONE
 4 files changed, 115 insertions(+)
 create mode 100644 ansible.cfg
 create mode 100644 files/default_site.html
 create mode 100644 inventory
 create mode 100644 site.yml
vbbose@workstation:~/BOSE-HOA-7.1$ git push
Counting objects: 7, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (6/6), done.
Writing objects: 100% (7/7), 1.16 KiB | 1.16 MiB/s, done.
Total 7 (delta 0), reused 0 (delta 0)
To github.com:BOSE-13/BOSE-HOA-7.1.git
   0a0b9d4..3152578  main -> main
```

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

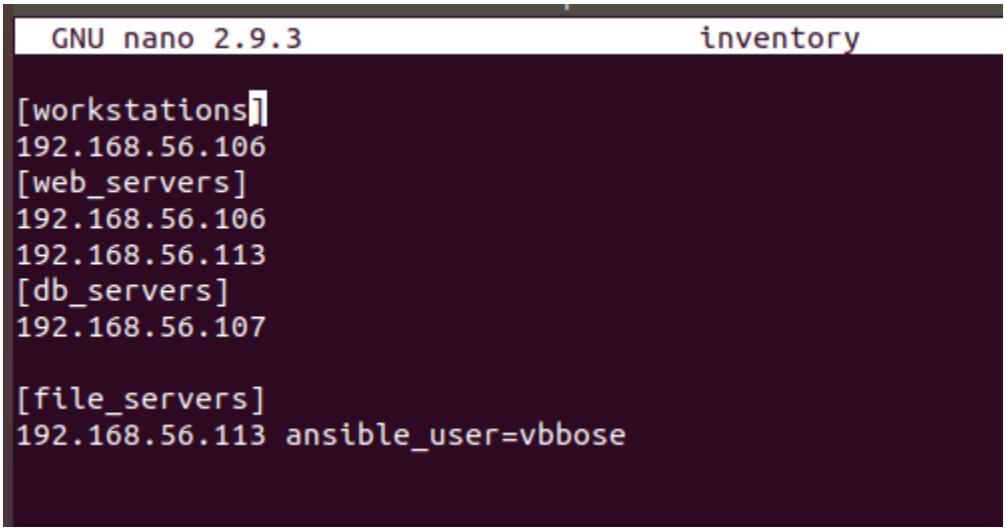
1. Edit the site.yml. Just before the web_servers play, create a new play:
   - hosts: workstations
     become: true
     tasks:

     - name: install unzip
       package:
         name: unzip

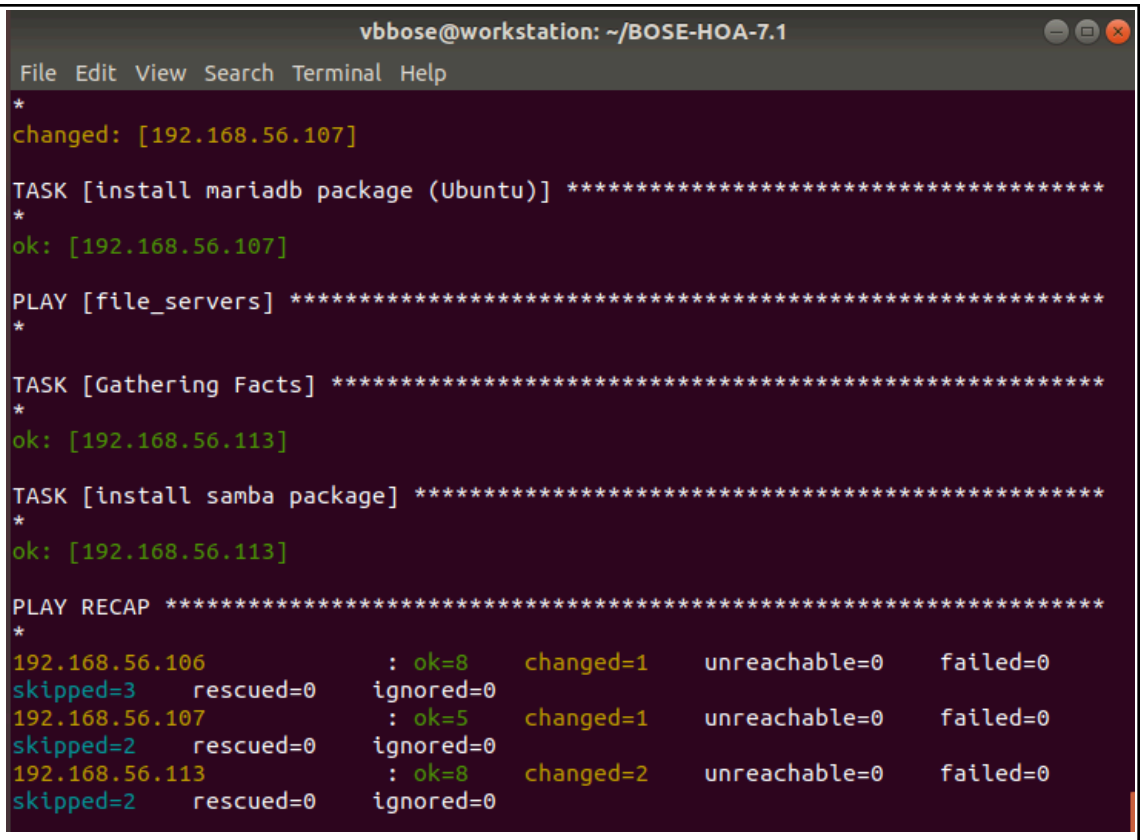
     - name: install terraform
       unarchive:
         src: https://releases.hashicorp.com/terraform/0.12.28/terraform_0.12.28_linux_amd64.zip
         dest: /usr/local/bin
         remote_src: yes
         mode: 0755
         owner: root
         group: root

```
- 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_$
      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.

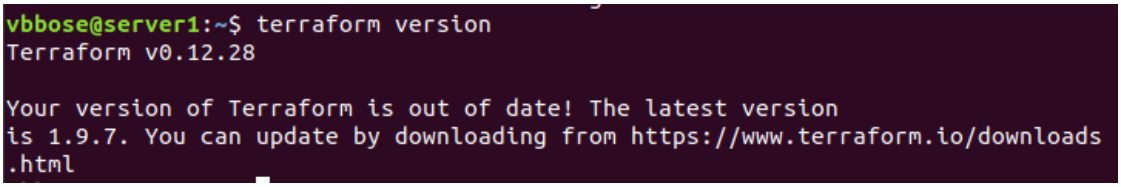


3. Run the playbook. Describe the output.



- The modifications are successful.

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



- The terraform is installed in the server successfully.

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.

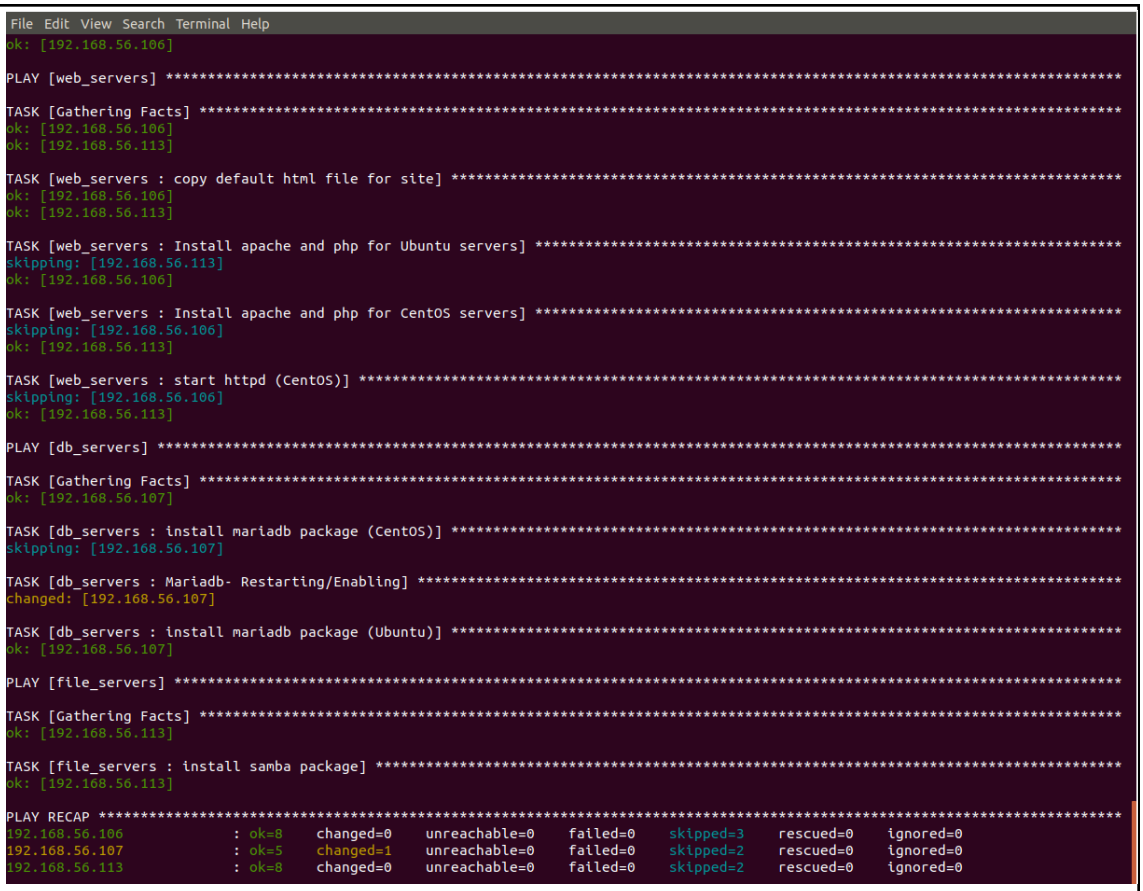
2. Under the same directory, create a new directory and name it roles. Enter the roles directory and create new directories: base, web_servers, file_servers, db_servers and workstations. For each directory, create a directory and name it tasks.

```
_servers workstations
vbbose@workstation:~/BOSE-HOA-7.1/roles$ ls
base  db_servers  file_servers  web_servers  workstations
vbbose@workstation:~/BOSE-HOA-7.1/roles$ cd base
vbbose@workstation:~/BOSE-HOA-7.1/roles/base$ mkdir tasks
vbbose@workstation:~/BOSE-HOA-7.1/roles/base$ ls
tasks
vbbose@workstation:~/BOSE-HOA-7.1/roles/base$ cd ..
vbbose@workstation:~/BOSE-HOA-7.1/roles$ cd web_servers
vbbose@workstation:~/BOSE-HOA-7.1/roles/web_servers$ mkdir tasks
vbbose@workstation:~/BOSE-HOA-7.1/roles/web_servers$ ls
tasks
vbbose@workstation:~/BOSE-HOA-7.1/roles/web_servers$ cd ..
vbbose@workstation:~/BOSE-HOA-7.1/roles$ cd file_servers
vbbose@workstation:~/BOSE-HOA-7.1/roles/file_servers$ mkdir tasks
vbbose@workstation:~/BOSE-HOA-7.1/roles/file_servers$ ls
tasks
vbbose@workstation:~/BOSE-HOA-7.1/roles/file_servers$ cd ..
vbbose@workstation:~/BOSE-HOA-7.1/roles$ cd db_servers
vbbose@workstation:~/BOSE-HOA-7.1/roles/db_servers$ mkdir tasks
vbbose@workstation:~/BOSE-HOA-7.1/roles/db_servers$ ls
tasks
vbbose@workstation:~/BOSE-HOA-7.1/roles/db_servers$ cd ..
vbbose@workstation:~/BOSE-HOA-7.1/roles$ cd workstations
vbbose@workstation:~/BOSE-HOA-7.1/roles/workstations$ mkdir tasks
vbbose@workstation:~/BOSE-HOA-7.1/roles/workstations$ ls
tasks
vbbose@workstation:~/BOSE-HOA-7.1/roles/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.

```
File  Edit  View  Search  Terminal  Help
ok: [192.168.56.106]

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

TASK [Gathering Facts] *********************************************************************************************************
ok: [192.168.56.106]
ok: [192.168.56.113]

TASK [web_servers : copy default html file for site] ***************************************************************************
ok: [192.168.56.106]
ok: [192.168.56.113]

TASK [web_servers : Install apache and php for Ubuntu servers] *****************************************************************
skipping: [192.168.56.113]
ok: [192.168.56.106]

TASK [web_servers : Install apache and php for CentOS servers] *****************************************************************
skipping: [192.168.56.106]
ok: [192.168.56.113]

TASK [web_servers : start httpd (CentOS)] **************************************************************************************
skipping: [192.168.56.106]
ok: [192.168.56.113]

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

TASK [Gathering Facts] *********************************************************************************************************
ok: [192.168.56.107]

TASK [db_servers : install mariadb package (CentOS)] ***************************************************************************
skipping: [192.168.56.107]

TASK [db_servers : Mariadb- Restarting/Enabling] *******************************************************************************
changed: [192.168.56.107]

TASK [db_servers : install mariadb package (Ubuntu)] ***************************************************************************
ok: [192.168.56.107]

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

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

TASK [file_servers : install samba package] ************************************************************************************
ok: [192.168.56.113]

PLAY RECAP *********************************************************************************************************************
192.168.56.106             : ok=8    changed=0    unreachable=0    failed=0    skipped=3    rescued=0    ignored=0
192.168.56.107             : ok=5    changed=1    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
192.168.56.113             : ok=8    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
```

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



- The site.yml is modified in order for this to run successfully.

Reflections:

Answer the following:

1. What is the importance of creating roles?
   - Creating roles in Ubuntu is essential for us as students learning about system administration and user management. By defining specific roles, we can ensure that everyone has the appropriate level of access, which enhances security and reduces the risk of accidental changes. Organizing user accounts by roles simplifies permission management, especially in group projects, and allows us to easily track who has access to what resources. This structure fosters better collaboration, as we can form teams based on projects and streamline the onboarding process for new members. Ultimately, utilizing roles in Ubuntu helps us build a secure and efficient learning environment, preparing us for real-world scenarios in system management.
2. What is the importance of managing files?
   - Effective file management in Ubuntu is essential for maintaining organization, efficiency, and security. A well-structured file system allows for quick access to documents, saving time and supporting collaboration. It also safeguards

sensitive information through proper permissions and backups, reducing the risk of data loss. Moreover, good file organization enhances system performance by minimizing clutter and optimizing storage. Understanding file types and permissions further empowers users, promoting better computer literacy. Overall, effective file management fosters a productive and secure working environment.