Name: Yuri P. Nollan	Date Performed: 10/09/2023
Course/Section: CPE31S6	Date Submitted: 10/09/2023
Instructor: Dr. Jonathan Taylar	Semester and SY: 1st Sem 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.

```
workstation@workstation:~/sysAds6$ mkdir files
workstation@workstation:~/sysAds6$ ls
ansible.cfg  install_apache.yml  inventory.yaml  site.yml
files        inventory           README.md
workstation@workstation:~/sysAds6$ cd files
workstation@workstation:~/sysAds6/files$ sudo nano default_site.html
workstation@workstation:~/sysAds6/files$ cat default_site.html
<!DOCTYPE html>
<html>
<body>
        <p>Yuri P. Nollan</p>
</body>
</html>
workstation@workstation:~/sysAds6/files$
```

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

```
- 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.

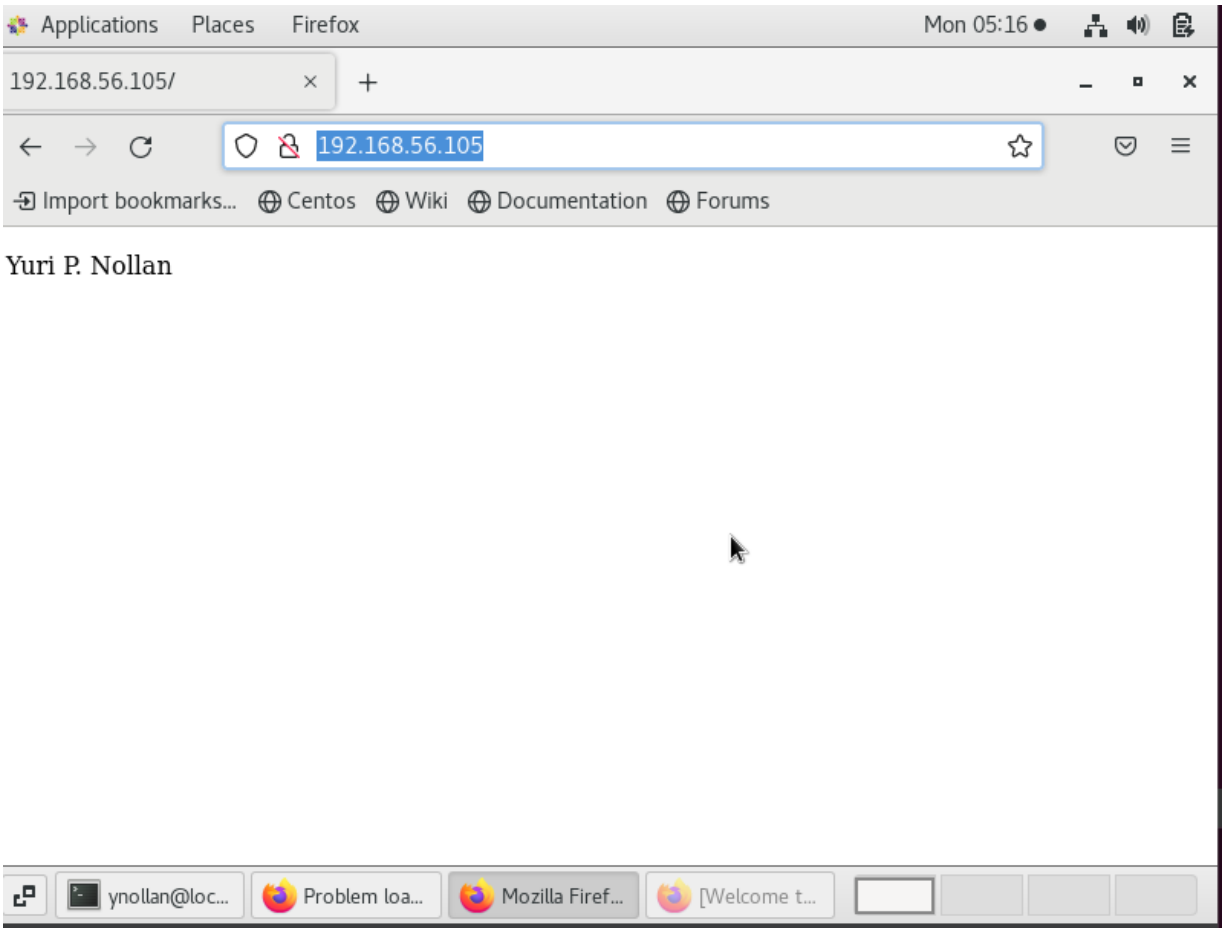
```
BECOME password:

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

TASK [Gathering Facts] *********************************************************
ok: [192.168.56.102]
ok: [192.168.56.104]
ok: [ynollan@192.168.56.105]

TASK [install updates (CentOS)] ************************************************
skipping: [192.168.56.102]
skipping: [192.168.56.104]
ok: [ynollan@192.168.56.105]

TASK [install updates (Ubuntu)] ************************************************
skipping: [ynollan@192.168.56.105]
ok: [192.168.56.102]
ok: [192.168.56.104]

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

TASK [Gathering Facts] *********************************************************
ok: [192.168.56.102]
ok: [ynollan@192.168.56.105]

TASK [copy default html file for site] *****************************************
changed: [192.168.56.102]
changed: [ynollan@192.168.56.105]

TASK [install apache and php for Ubuntu servers] *******************************
skipping: [ynollan@192.168.56.105]
ok: [192.168.56.102]

TASK [install apache and php for CentOS servers] *******************************
skipping: [192.168.56.102]
ok: [ynollan@192.168.56.105]
```

```
ok: [ynollan@192.168.56.105]

TASK [start httpd (CentOS)] ****************************************************
skipping: [192.168.56.102]
ok: [ynollan@192.168.56.105]

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

TASK [Gathering Facts] *********************************************************
ok: [192.168.56.104]
ok: [ynollan@192.168.56.105]

TASK [install mariadb package (CentOS)] ****************************************
skipping: [192.168.56.104]
ok: [ynollan@192.168.56.105]

TASK [install mariadb package (Ubuntu)] ****************************************
skipping: [ynollan@192.168.56.105]
ok: [192.168.56.104]

TASK [Mariadb- Restarting/Enabling] ********************************************
changed: [192.168.56.104]
changed: [ynollan@192.168.56.105]

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

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

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

PLAY RECAP *********************************************************************
192.168.56.102             : ok=7    changed=1    unreachable=0    failed=0    skipped=3    rescued=0    ignored=0
192.168.56.104             : ok=5    changed=1    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
ynollan@192.168.56.105     : ok=9    changed=2    unreachable=0    failed=0    skipped=3    rescued=0    ignored=0

workstation@workstation:~/sysAds6$
```

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.

```
workstation@server1:~$ cd /var/www/html
workstation@server1:/var/www/html$ cat index.html
<!DOCTYPE html>
<html>
<body>
        <p>Yuri P. Nollan</p>
</body>
</html>
workstation@server1:/var/www/html$
```

```
[ynollan@localhost ~]$ cd /var/www/html
[ynollan@localhost html]$ cat index.html
<!DOCTYPE html>
<html>
<body>
        <p>Yuri P. Nollan</p>
</body>
</html>
[ynollan@localhost html]$
```



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

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

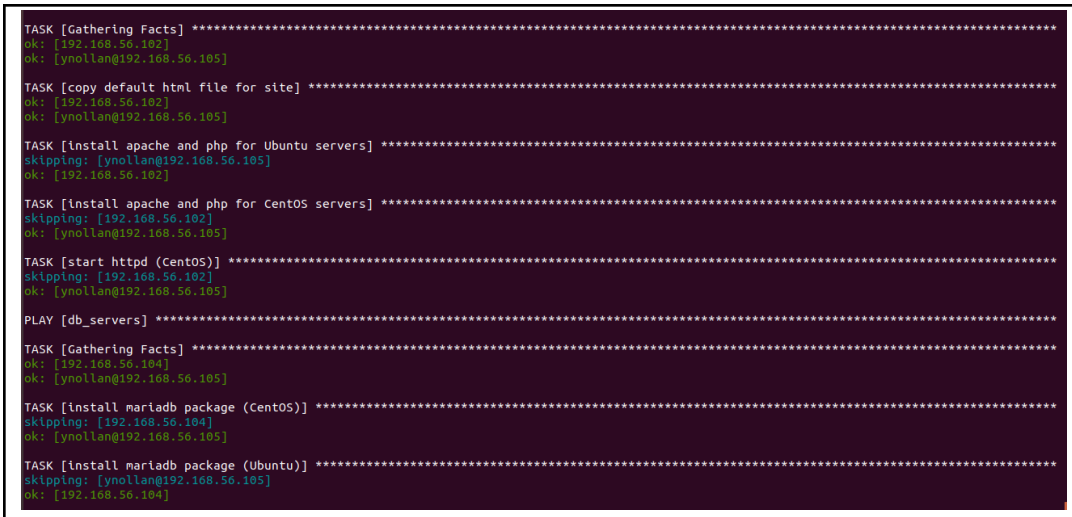
```
- hosts: workstations
  become: true
  tasks:

  - name: install unzip
    package:
      name: unzip

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

- hosts: web_servers
  become: true
  tasks:
```

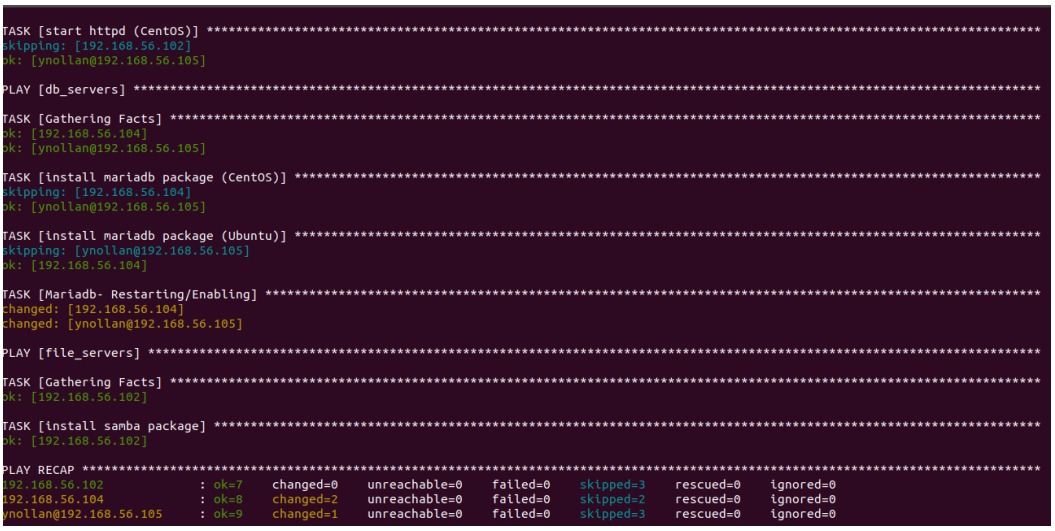
2. Edit the inventory file and add workstations group. Add any Ubuntu remote server. Make sure to remember the IP address.

```
[web_servers]
192.168.56.102
ynollan@192.168.56.105

[db_servers]
192.168.56.104
ynollan@192.168.56.105

[file_servers]
192.168.56.102


[workstations]
192.168.56.104
```

3. Run the playbook. Describe the output.

```
workstation@workstation:~/sysAds6$ ansible-playbook --ask-become-pass site.yml
BECOME password:

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

TASK [Gathering Facts] *********************************************************************************************************************************
ok: [192.168.56.102]
ok: [192.168.56.104]
ok: [ynollan@192.168.56.105]

TASK [install updates (CentOS)] ************************************************************************************************************************
skipping: [192.168.56.102]
skipping: [192.168.56.104]
ok: [ynollan@192.168.56.105]

TASK [install updates (Ubuntu)] ************************************************************************************************************************
skipping: [ynollan@192.168.56.105]
ok: [192.168.56.102]
ok: [192.168.56.104]

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

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

TASK [install unzip] ***********************************************************************************************************************************
ok: [192.168.56.104]

TASK [install terraform] *******************************************************************************************************************************
changed: [192.168.56.104]

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

TASK [Gathering Facts] *********************************************************************************************************************************
ok: [192.168.56.102]
ok: [ynollan@192.168.56.105]
```





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

```
workstation@server2:~$ terraform
Usage: terraform [-version] [-help] <command> [args]

The available commands for execution are listed below.
The most common, useful commands are shown first, followed by
less common or more advanced commands. If you're just getting
started with Terraform, stick with the common commands. For the
other commands, please read the help and docs before usage.

Common commands:
    apply              Builds or changes infrastructure
    console            Interactive console for Terraform interpolations
    destroy            Destroy Terraform-managed infrastructure
    env                Workspace management
    fmt                Rewrites config files to canonical format
    get                Download and install modules for the configuration
    graph              Create a visual graph of Terraform resources
    import             Import existing infrastructure into Terraform
    init               Initialize a Terraform working directory
    login              Obtain and save credentials for a remote host
    logout             Remove locally-stored credentials for a remote host
    output             Read an output from a state file
    plan               Generate and show an execution plan
    providers          Prints a tree of the providers used in the configuration
```

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.

```
workstation@workstation:~/sysAds6$ sudo nano site.yml
workstation@workstation:~/sysAds6$ mkdir roles
workstation@workstation:~/sysAds6$ cd roles
workstation@workstation:~/sysAds6/roles$ mkdir base
workstation@workstation:~/sysAds6/roles$ mkdir web_servers
workstation@workstation:~/sysAds6/roles$ mkdir file_servers
workstation@workstation:~/sysAds6/roles$ mkdir db_servers
workstation@workstation:~/sysAds6/roles$ mkdir workstations
workstation@workstation:~/sysAds6/roles$ ls
base  db_servers  file_servers  web_servers  workstations
workstation@workstation:~/sysAds6/roles$
```

```
workstation@workstation:~/sysAds6/roles$ cd base
workstation@workstation:~/sysAds6/roles/base$ mkdir tasks
workstation@workstation:~/sysAds6/roles/base$ cd ..
workstation@workstation:~/sysAds6/roles$ cd db_servers
workstation@workstation:~/sysAds6/roles/db_servers$ mkdir tasks
workstation@workstation:~/sysAds6/roles/db_servers$ cd ..
workstation@workstation:~/sysAds6/roles$ cd file_servers
workstation@workstation:~/sysAds6/roles/file_servers$ mkdir tasks
workstation@workstation:~/sysAds6/roles/file_servers$ cd ..
workstation@workstation:~/sysAds6/roles$ cd web_servers
workstation@workstation:~/sysAds6/roles/web_servers$ mkdir tasks
workstation@workstation:~/sysAds6/roles/web_servers$ cd ..
workstation@workstation:~/sysAds6/roles$ cd workstations
workstation@workstation:~/sysAds6/roles/workstations$ mkdir tasks
workstation@workstation:~/sysAds6/roles/workstations$ cd ..
workstation@workstation:~/sysAds6/roles$
```

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 2.9.3                          main.yml

---

- 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_linu$
      dest: /usr/local/bin
      remote_src: yes
      mode: 0755
      owner: root
      group: root
```

```
 GNU nano 2.9.3                          main.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  - 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
```

```
    tags: apache,apache2,ubuntu
    apt:
      name:
        - apache2
        - libapache2-mod-php
      state: latest
    when: ansible_distribution == "Ubuntu"

LibreOffice Writer 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
      enabled: true
    when: ansible_distribution == "CentOS"
```

```
 GNU nano 2.9.3                                  main.yml

---

- hosts: db_servers
  become: true
  tasks:

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

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

```
 GNU nano 2.9.3                                  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.

```
workstation@workstation:~/sysAds6$ ansible-playbook --ask-become-pass site.yml
BECOME password:

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

TASK [Gathering Facts] *********************************************************************************************************************************
ok: [192.168.56.102]
ok: [192.168.56.104]
ok: [ynollan@192.168.56.105]

TASK [update repository index (CentOS)] ****************************************************************************************************************
skipping: [192.168.56.102]
skipping: [192.168.56.104]
changed: [ynollan@192.168.56.105]

TASK [install updates (Ubuntu)] ************************************************************************************************************************
skipping: [ynollan@192.168.56.105]
ok: [192.168.56.102]
ok: [192.168.56.104]

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

TASK [Gathering Facts] *********************************************************************************************************************************
ok: [192.168.56.102]
ok: [ynollan@192.168.56.105]
ok: [192.168.56.104]

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

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

```
TASK [workstations : install unzip] ********************************************************************************************************************
ok: [192.168.56.104]

TASK [workstations : install terraform] ****************************************************************************************************************
ok: [192.168.56.104]

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

TASK [Gathering Facts] *********************************************************************************************************************************
ok: [192.168.56.102]
ok: [ynollan@192.168.56.105]

TASK [web_servers : copy default html file for site] ***************************************************************************************************
ok: [192.168.56.102]
ok: [ynollan@192.168.56.105]

TASK [web_servers : install apache and php for Ubuntu servers] *****************************************************************************************
skipping: [ynollan@192.168.56.105]
ok: [192.168.56.102]

TASK [web_servers : install apache and php for CentOS servers] *****************************************************************************************
skipping: [192.168.56.102]
ok: [ynollan@192.168.56.105]

TASK [web_servers : start httpd (CentOS)] **************************************************************************************************************
skipping: [192.168.56.102]
ok: [ynollan@192.168.56.105]

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

```
TASK [Gathering Facts] *********************************************************************************************************************************
ok: [192.168.56.104]
ok: [ynollan@192.168.56.105]

TASK [db_servers : install mariadb package (CentOS)] ***************************************************************************************************
skipping: [192.168.56.104]
ok: [ynollan@192.168.56.105]

TASK [db_servers : Mariadb- Restarting/Enabling] ********************************************************************************************************
changed: [192.168.56.104]
changed: [ynollan@192.168.56.105]

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

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

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

PLAY RECAP *********************************************************************************************************************************************
192.168.56.102             : ok=8    changed=0    unreachable=0    failed=0    skipped=3    rescued=0    ignored=0
192.168.56.104             : ok=8    changed=1    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
ynollan@192.168.56.105     : ok=10   changed=2    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0

workstation@workstation:~/sysAds6$
```

Reflections:

Answer the following:

1. What is the importance of creating roles?
    - For an organization to function effectively, clearly, and productively, roles must be established. Roles offer clarity and concentration, enabling people to specialize, assume responsibility for, and be held accountable for, particular duties. Workflows are streamlined, conflicts are avoided, resources are better allocated, communication is improved, and organizational growth and adaptation are made possible. In the end, responsibilities are essential to effectively and efficiently fulfilling organizational goals.
2. What is the importance of managing files?
    - For effective organization, easy access, and data security, managing files is essential. Redundancy is decreased, communication is aided, legal compliance is ensured, and data backup is made easier. Effective file management boosts output, reduces storage expenses, and facilitates smooth updates and migrations.

Conclusion:

Overall, this activity has helped me enhance my knowledge in navigating through the linux ubuntu and CentOS. It also helped me manage the files in remote servers through a centralized server and implement roles in ansible.