Name: Renier L. Lope	Date Performed: 10/07/2024
Course/Section: CPE212 - CPE31S2	Date Submitted: 10/09/2024
Instructor: Engr. Robin Valenzuela	Semester and SY: 1st Sem(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.

```
rnrlope@workstation:~/CPE212_LOPE_Act7$ mkdir files
rnrlope@workstation:~/CPE212_LOPE_Act7$ cd files
rnrlope@workstation:~/CPE212_LOPE_Act7/files$ nano default_site.html
```

```
rnrlope@workstation:~/CPE212_LOPE_Act7/files$ cat default_site.html
<html>
      <body>
            <header>Good Morning!</header>
      </body>
<html>
rnrlope@workstation:~/CPE212_LOPE_Act7/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

```
  - 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
```

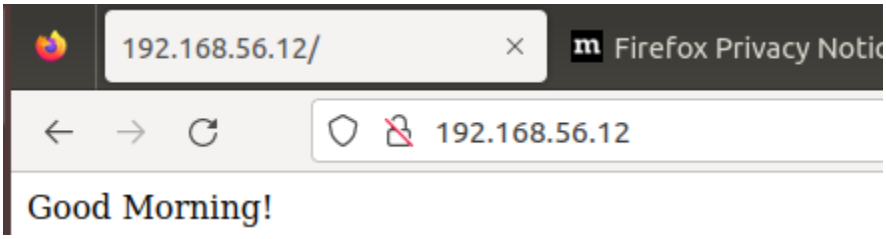
3. Run the playbook *site.yml*. Describe the changes.
    - **The playbook will copy the default_site.html with the destination /var/www/html/index.html to the assigned servers which is the web_servers.**

```
TASK [copy default html file for site] ************************************
*
changed: [server1]
changed: [server2]
changed: [centOS]
```

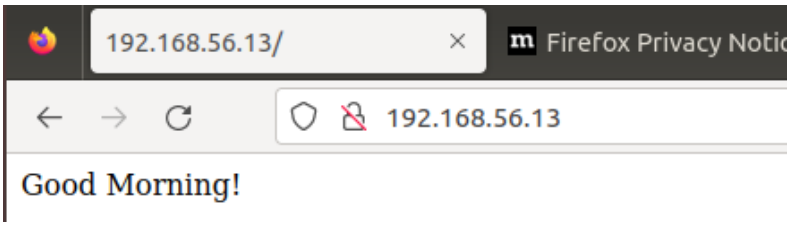
```
PLAY RECAP ********************************************************************
*
centOS                     : ok=9    changed=3    unreachable=0    failed=0
skipped=3    rescued=0    ignored=0
server1                    : ok=5    changed=1    unreachable=0    failed=0
skipped=3    rescued=0    ignored=0
server2                    : ok=5    changed=1    unreachable=0    failed=0
skipped=3    rescued=0    ignored=0
server3                    : ok=4    changed=0    unreachable=0    failed=0
skipped=1    rescued=0    ignored=0

rnrlope@workstation:~/CPE212_LOPE_Act7$
```

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.
    - **The contents of the defaut_file.html was copied into /var/www/html directory named index.html inside the remote servers using the playbook file, making it accessible by inputting the IP Address of the remote servers in the browser.**

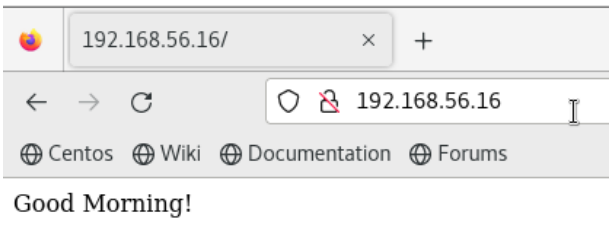
```
rnrlope@server1:~$ cat /var/www/html/index.html
<html>
      <body>
            <header>Good Morning!</header>
      </body>
<html>
rnrlope@server1:~$
```




```
rnrlope@server2:~$ cat /var/www/html/index.html
<html>
      <body>
            <header>Good Morning!</header>
      </body>
<html>
rnrlope@server2:~$
```




```
rnrlope@workstation:~/CPE212_LOPE_Act7$ ssh rnrlope@centOS
Last login: Sun Oct  6 21:29:37 2024 from 192.168.56.11
[rnrlope@localhost ~]$ cat /var/www/html/index.html
<html>
      <body>
            <header>Good Morning!</header>
      </body>
<html>
[rnrlope@localhost ~]$
```



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

```
rnrlope@workstation:~/CPE212_LOPE_Act7$ git add files
rnrlope@workstation:~/CPE212_LOPE_Act7$ git add site.yml
rnrlope@workstation:~/CPE212_LOPE_Act7$ git commit -m "Activiy 7 Task 1"
[main 73a578d] Activiy 7 Task 1
 2 files changed, 14 insertions(+)
 create mode 100644 files/default_site.html
rnrlope@workstation:~/CPE212_LOPE_Act7$ git push origin main
Counting objects: 5, done.
Delta compression using up to 2 threads.
Compressing objects: 100% (4/4), done.
Writing objects: 100% (5/5), 578 bytes | 578.00 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:RenierCode/CPE212_LOPE_Act7.git
   1204425..73a578d  main -> main
rnrlope@workstation:~/CPE212_LOPE_Act7$
```

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

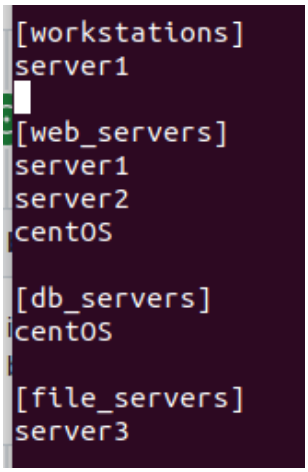
1. Edit the site.yml. Just before the web_servers play, create a new play:
   - hosts: workstations
     become: true
     tasks:

     - 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
```

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

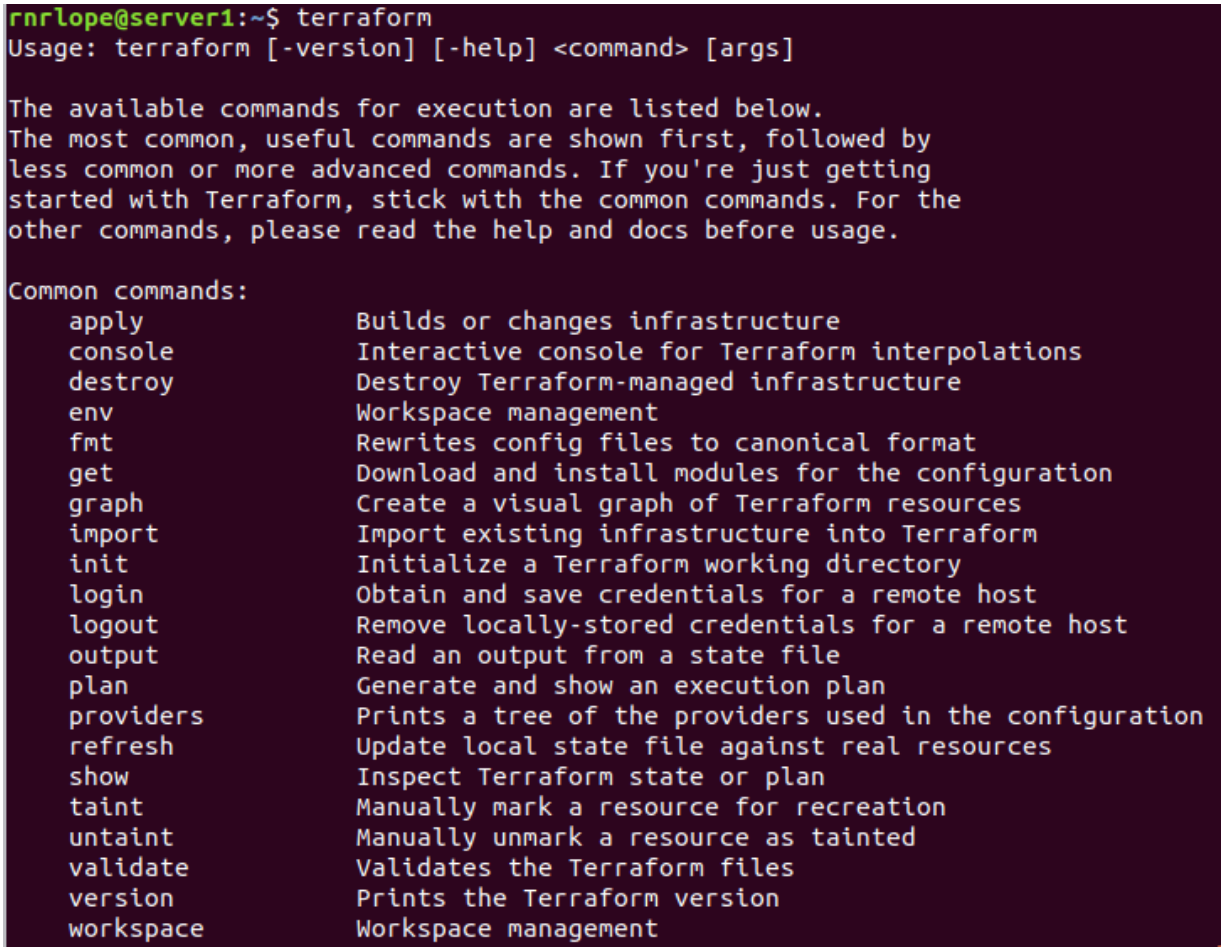


3. Run the playbook. Describe the output.
   - **The playbook will install terraform to the assigned hosts which is the workstations group.**

```
TASK [install terraform] ********************
************************
changed: [server1]
```

```
PLAY RECAP *********************************************************************
**********************
centOS                     : ok=9    changed=1    unreachable=0    failed=0
skipped=3    rescued=0    ignored=0
server1                    : ok=8    changed=1    unreachable=0    failed=0
skipped=3    rescued=0    ignored=0
server2                    : ok=5    changed=0    unreachable=0    failed=0
skipped=3    rescued=0    ignored=0
server3                    : ok=4    changed=0    unreachable=0    failed=0
skipped=1    rescued=0    ignored=0

rnrlope@workstation:~/CPE212_LOPE_Act7$
```

4. On the Ubuntu remote workstation, type terraform to verify installation of terraform. Describe the output.
    - **Issuing terraform will display its information including its common commands.**



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.

```
rnrlope@workstation:~/CPE212_LOPE_Act7$ mkdir roles
rnrlope@workstation:~/CPE212_LOPE_Act7$ cd roles
rnrlope@workstation:~/CPE212_LOPE_Act7/roles$ mkdir base web_servers file_servers
db_servers workstations
rnrlope@workstation:~/CPE212_LOPE_Act7/roles$ ls
base  db_servers  file_servers  web_servers  workstations
rnrlope@workstation:~/CPE212_LOPE_Act7/roles$ cd base
rnrlope@workstation:~/CPE212_LOPE_Act7/roles/base$ mkdir tasks
rnrlope@workstation:~/CPE212_LOPE_Act7/roles/base$ cd ..
rnrlope@workstation:~/CPE212_LOPE_Act7/roles$ cd db_servers
rnrlope@workstation:~/CPE212_LOPE_Act7/roles/db_servers$ mkdir tasks
rnrlope@workstation:~/CPE212_LOPE_Act7/roles/db_servers$ cd ..
rnrlope@workstation:~/CPE212_LOPE_Act7/roles$ cd file_servers
rnrlope@workstation:~/CPE212_LOPE_Act7/roles/file_servers$ mkdir tasks
rnrlope@workstation:~/CPE212_LOPE_Act7/roles/file_servers$ cd ..
rnrlope@workstation:~/CPE212_LOPE_Act7/roles$ cd web_servers
rnrlope@workstation:~/CPE212_LOPE_Act7/roles/web_servers$ mkdir tasks
rnrlope@workstation:~/CPE212_LOPE_Act7/roles/web_servers$ cd ..
rnrlope@workstation:~/CPE212_LOPE_Act7/roles$ cd workstations
rnrlope@workstation:~/CPE212_LOPE_Act7/roles/workstations$ mkdir tasks
rnrlope@workstation:~/CPE212_LOPE_Act7/roles/workstations$ cd ..
rnrlope@workstation:~/CPE212_LOPE_Act7/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.

```
rnrlope@workstation:~/CPE212_LOPE_Act7/roles$ cd base/tasks
rnrlope@workstation:~/CPE212_LOPE_Act7/roles/base/tasks$ nano main.yml
Use "fg" to return to nano.

[3]+  Stopped                 nano main.yml
rnrlope@workstation:~/CPE212_LOPE_Act7/roles/base/tasks$ cat 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"
rnrlope@workstation:~/CPE212_LOPE_Act7/roles/base/tasks$
```

BASE

```
rnrlope@workstation:~/CPE212_LOPE_Act7/roles$ cd db_servers/tasks
rnrlope@workstation:~/CPE212_LOPE_Act7/roles/db_servers/tasks$ nano main.yml
Use "fg" to return to nano.

[7]+  Stopped                 nano main.yml
rnrlope@workstation:~/CPE212_LOPE_Act7/roles/db_servers/tasks$ cat main.yml
---
- name: install mariadb package (CentOS)
  tags: centos, db,mariadb
  yum:
    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"
rnrlope@workstation:~/CPE212_LOPE_Act7/roles/db_servers/tasks$
```

DB_SERVERS

```
rnrlope@workstation:~/CPE212_LOPE_Act7/roles$ cd file_servers/tasks
rnrlope@workstation:~/CPE212_LOPE_Act7/roles/file_servers/tasks$ nano main.yml
Use "fg" to return to nano.

[8]+  Stopped                 nano main.yml
rnrlope@workstation:~/CPE212_LOPE_Act7/roles/file_servers/tasks$ cat main.yml
---
- name: install samba package
  tags: samba
  package:
    name: samba
    state: latest
rnrlope@workstation:~/CPE212_LOPE_Act7/roles/file_servers/tasks$
```

FILE_SERVERS

```
rnrlope@workstation:~/CPE212_LOPE_Act7/roles$ cd web_servers/tasks
rnrlope@workstation:~/CPE212_LOPE_Act7/roles/web_servers/tasks$ nano main.yml
Use "fg" to return to nano.

[9]+  Stopped                 nano main.yml
rnrlope@workstation:~/CPE212_LOPE_Act7/roles/web_servers/tasks$ cat main.yml
---
- 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
rnrlope@workstation:~/CPE212_LOPE_Act7/roles/web_servers/tasks$
```

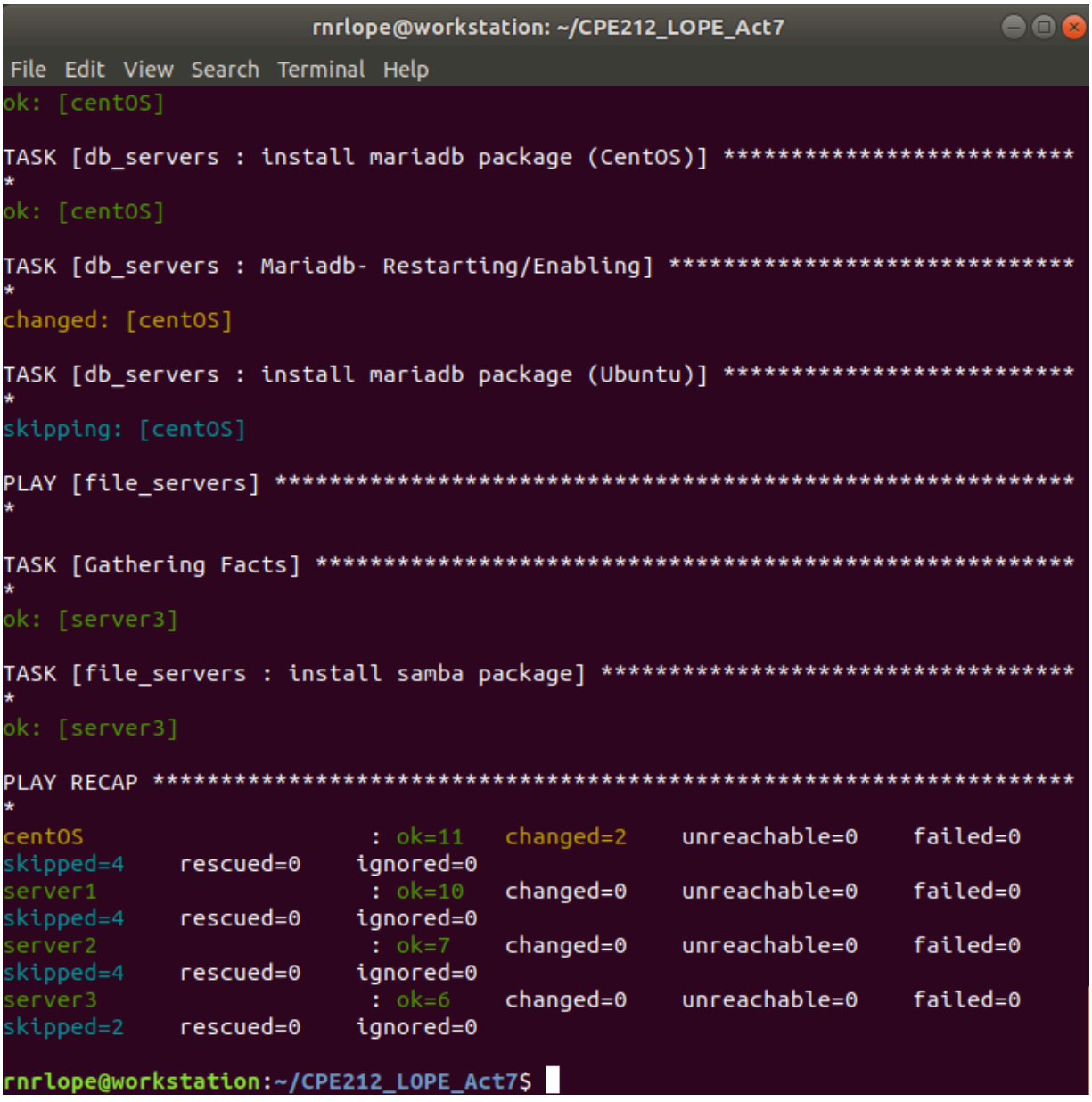
WEB_SERVERS

```
rnrlope@workstation:~/CPE212_LOPE_Act7/roles$ cd workstations/tasks
rnrlope@workstation:~/CPE212_LOPE_Act7/roles/workstations/tasks$ nano main.yml
rnrlope@workstation:~/CPE212_LOPE_Act7/roles/workstations/tasks$ cat 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_linu
x_amd64.zip
    dest: /usr/local/bin
    remote_src: yes
    mode: 0755
    owner: root
    group: root

rnrlope@workstation:~/CPE212_LOPE_Act7/roles/workstations/tasks$
```

WORKSTATIONS

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

   - **By assigning roles to each group of remote servers inside the site.yml and creating a roles directory, then creating a directory inside for each group which is base, web_servers, file_servers, db_servers and workstations. Inside the directories for each group creating a directory named tasks containing a playbook that contains each of the group Plays, the site.yml was executed perfectly.**



GIT PUSH:

```
rnrlope@workstation:~/CPE212_LOPE_Act7$ git add inventory
rnrlope@workstation:~/CPE212_LOPE_Act7$ git add roles
rnrlope@workstation:~/CPE212_LOPE_Act7$ git add site.yml
rnrlope@workstation:~/CPE212_LOPE_Act7$ git commint -m "Act7 done"
git: 'commint' is not a git command. See 'git --help'.

The most similar command is
        commit
rnrlope@workstation:~/CPE212_LOPE_Act7$ git commit -m "Act7 done"
[main 1d80a3b] Act7 done
 11 files changed, 92 insertions(+), 1 deletion(-)
 create mode 100644 roles/base/tasks/.main.yml.swp
 create mode 100644 roles/base/tasks/main.yml
 create mode 100644 roles/db_servers/tasks/.main.yml.swp
 create mode 100644 roles/db_servers/tasks/main.yml
 create mode 100644 roles/file_servers/tasks/.main.yml.swp
 create mode 100644 roles/file_servers/tasks/main.yml
 create mode 100644 roles/web_servers/tasks/.main.yml.swp
 create mode 100644 roles/web_servers/tasks/main.yml
 create mode 100644 roles/workstations/tasks/main.yml
rnrlope@workstation:~/CPE212_LOPE_Act7$ git push origin main
Counting objects: 24, done.
Delta compression using up to 2 threads.
Compressing objects: 100% (18/18), done.
Writing objects: 100% (24/24), 2.26 KiB | 2.26 MiB/s, done.
Total 24 (delta 5), reused 0 (delta 0)
remote: Resolving deltas: 100% (5/5), completed with 2 local objects.
To github.com:RenierCode/CPE212_LOPE_Act7.git
   33f1685..1d80a3b  main -> main
rnrlope@workstation:~/CPE212_LOPE_Act7$
```

GITHUB LINK:

https://github.com/RenierCode/CPE212_LOPE_Act7.git

Reflections:

Answer the following:

1. What is the importance of creating roles?
   - **The importance of roles is to simplify the creation of complex playbooks by providing a framework for completely self-contained or interdependent collections of variables, tasks, files, templates, and modules, making them more reusable. Roles also provide an easier way for maintenance.**
2. What is the importance of managing files?
   - **The importance of managing files is to create an organized system that improves efficiency by separating and grouping the files based on types or any other criteria keeping the files structured, making it easier for files to manage, locate, and also to create backups ensuring that data can be restored easily. File management also promotes security by ensuring the configuration files are stored securely and hidden inside only the specified groups or roles.**