| | |
|---|---|
| **Name:** Ruud Van G. Apostol | **Date Performed:** 10/03/25 |
| **Course/Section:** CPE212 / CPE31S4 | **Date Submitted:** 10/03/25 |
| **Instructor:** Engr. Robin Valenzuela | **Semester and SY:**1st.sem / 2025-2026 |

# Activity 7: Managing Files and Creating Roles in Ansible

1. **Objectives:**

1.1 Manage files in remote servers

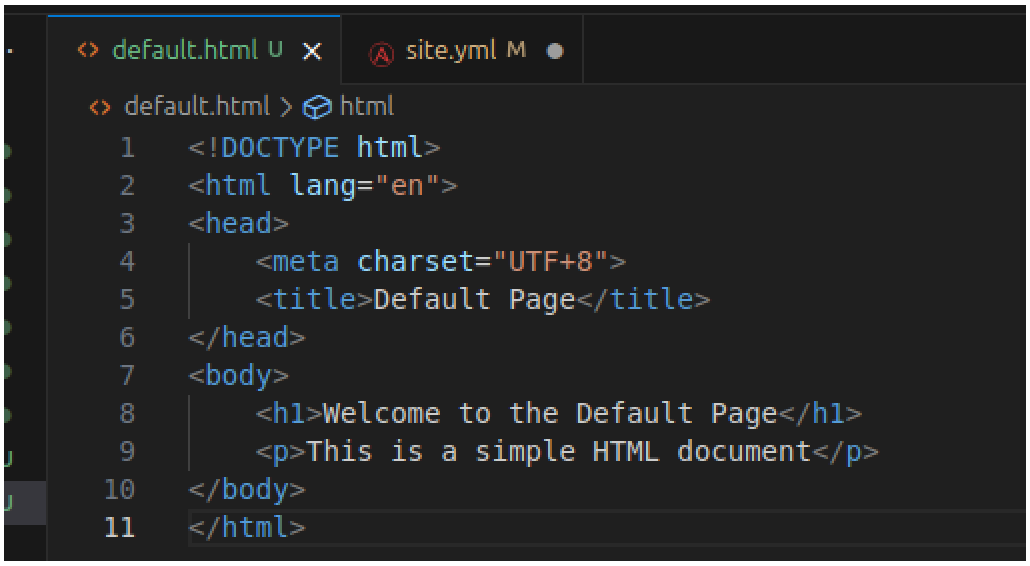
1.2 Implement roles in ansible

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.

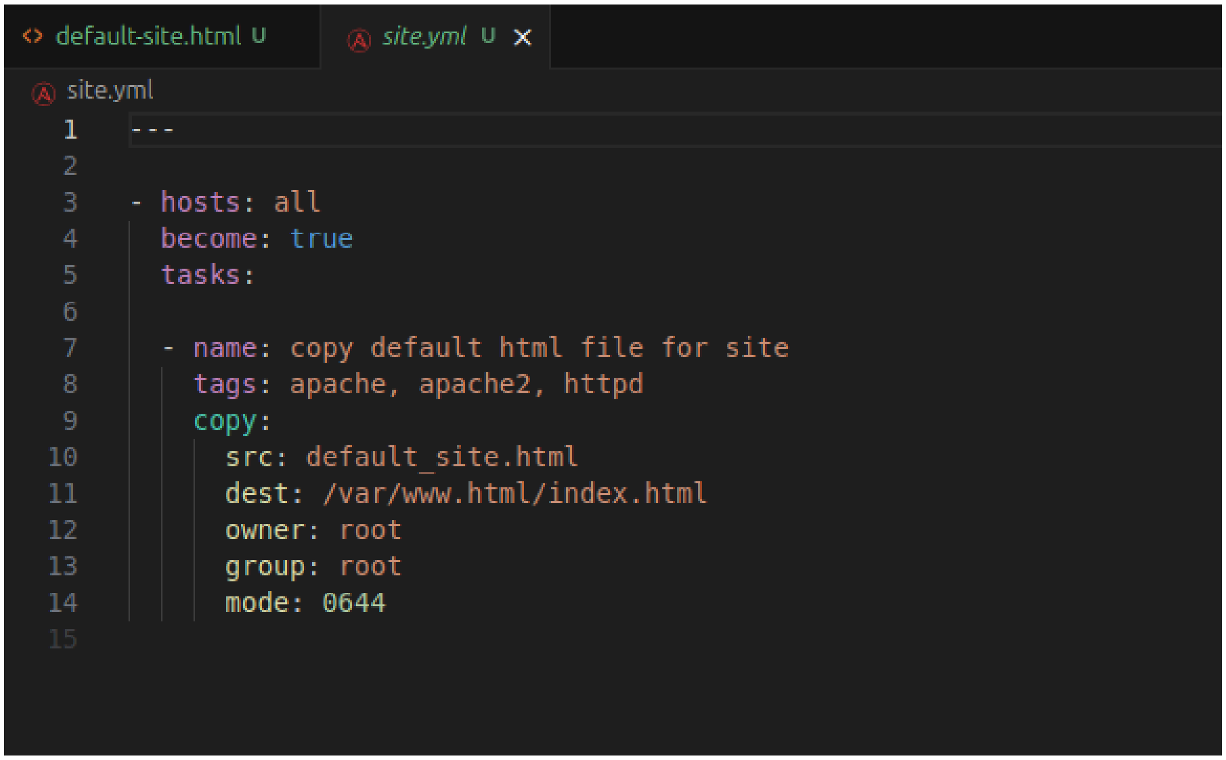


```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF+8">
    <title>Default Page</title>
</head>
<body>
    <h1>Welcome to the Default Page</h1>
    <p>This is a simple HTML document</p>
</body>
</html>
```

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:

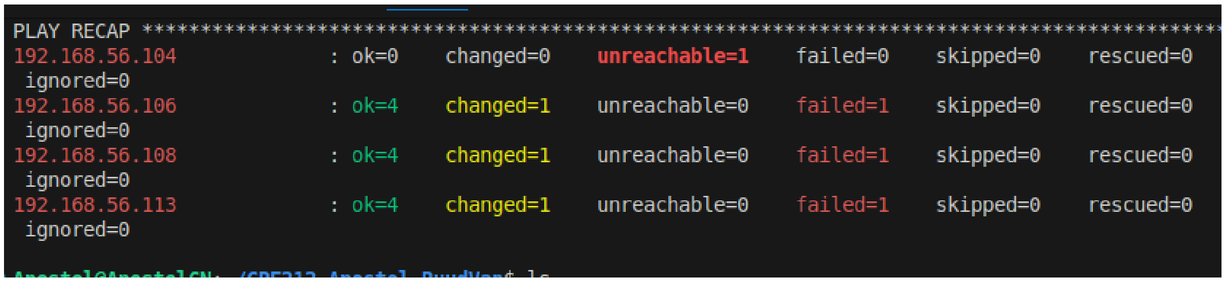
```
- name: copy default html file for site

  tags: apache, apache2, httpd
  copy:
    src: default_site.html
    dest: /var/www/html/index.html
    owner: root
```

```
          group: root
          mode: 0644
```



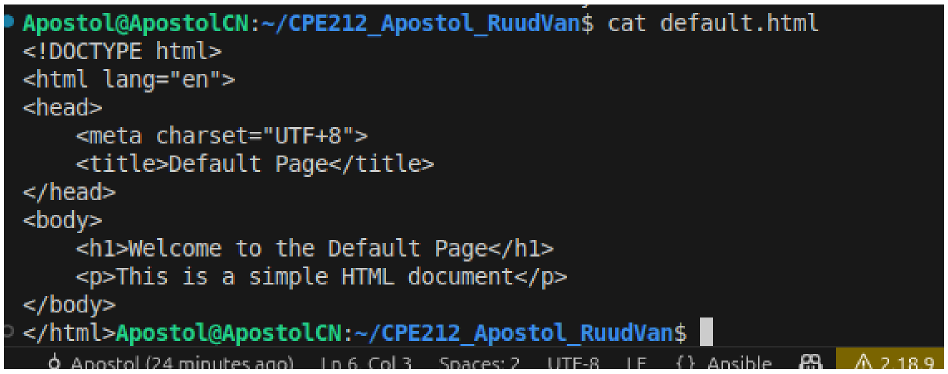
```yaml
---

- hosts: all
  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.



```
PLAY RECAP ***************************************************************************************
192.168.56.104             : ok=0    changed=0    unreachable=1    failed=0    skipped=0    rescued=0
 ignored=0
192.168.56.106             : ok=4    changed=1    unreachable=0    failed=1    skipped=0    rescued=0
 ignored=0
192.168.56.108             : ok=4    changed=1    unreachable=0    failed=1    skipped=0    rescued=0
 ignored=0
192.168.56.113             : ok=4    changed=1    unreachable=0    failed=1    skipped=0    rescued=0
 ignored=0
```

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.



```
Apostol@ApostolCN:~/CPE212_Apostol_RuudVan$ cat default.html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF+8">
    <title>Default Page</title>
</head>
<body>
    <h1>Welcome to the Default Page</h1>
    <p>This is a simple HTML document</p>
</body>
</html>Apostol@ApostolCN:~/CPE212_Apostol_RuudVan$ 
```

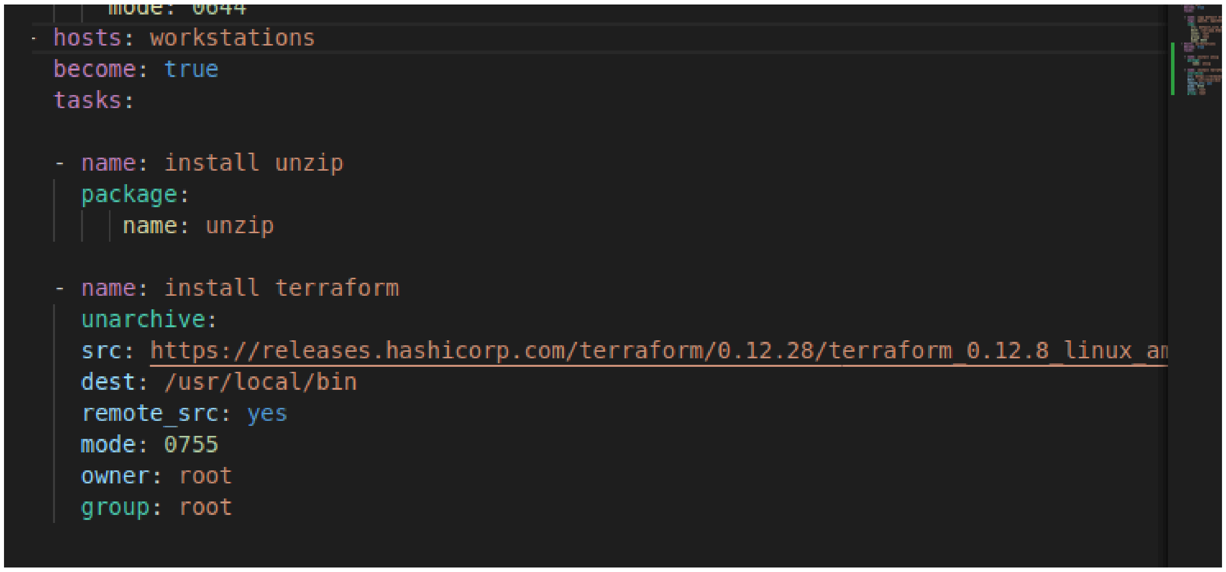
Apostol (24 minutes ago)   Ln 6, Col 3   Spaces: 2   UTF-8   LF   {} Ansible   ⚠ 2.18.9

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

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

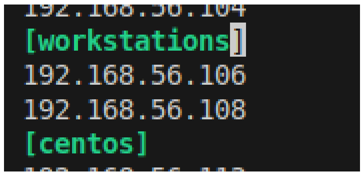
1. Edit the site.yml. Just before the web_servers play, create a new play:

```
- hosts: workstations
  become: true—copy
  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
```



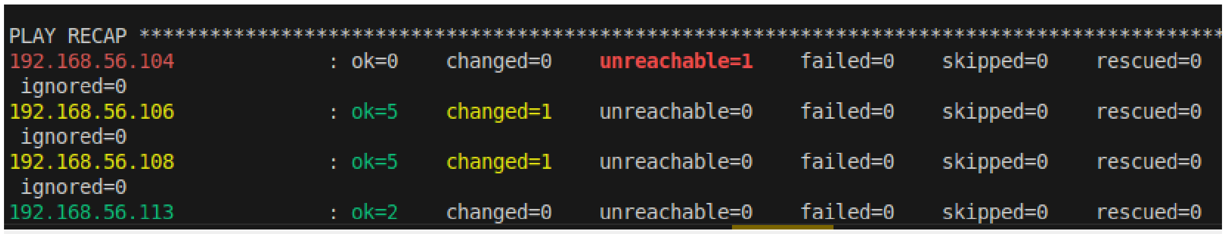
```
    mode: 0644
- hosts: workstations
  become: true
  tasks:

  - name: install unzip
    package:
      name: unzip

  - name: install terraform
    unarchive:
    src: https://releases.hashicorp.com/terraform/0.12.28/terraform_0.12.8_linux_an
    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.



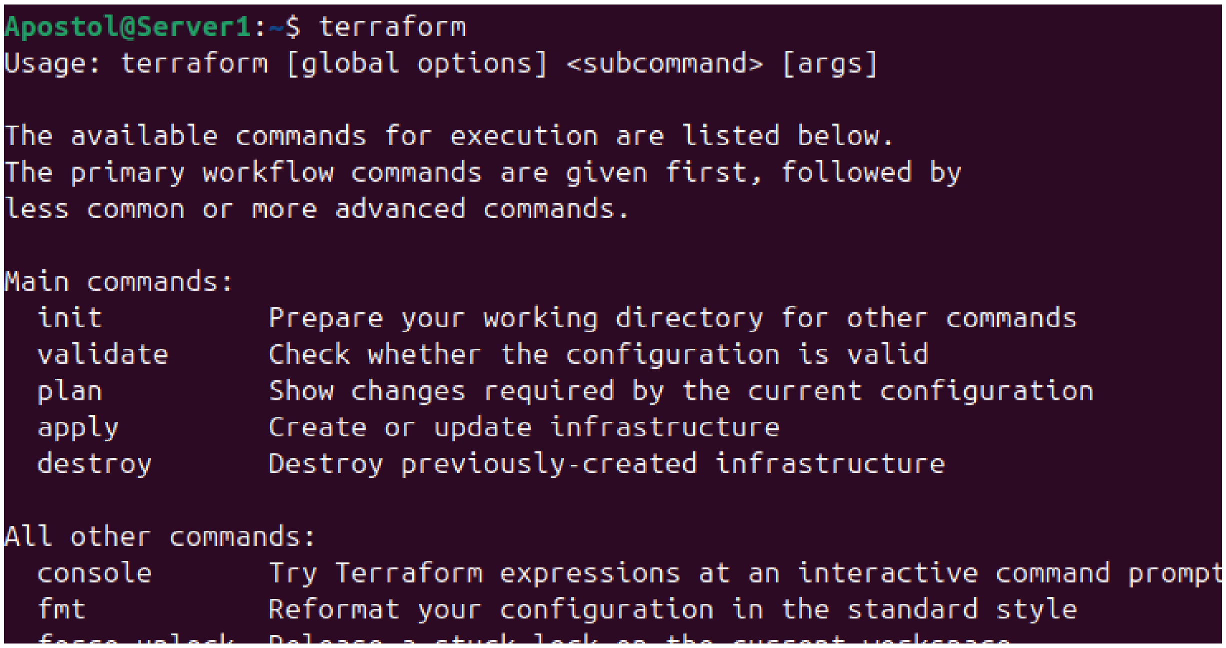
```
192.168.56.104
[workstations]
192.168.56.106
192.168.56.108
[centos]
```

3. Run the playbook. Describe the output.

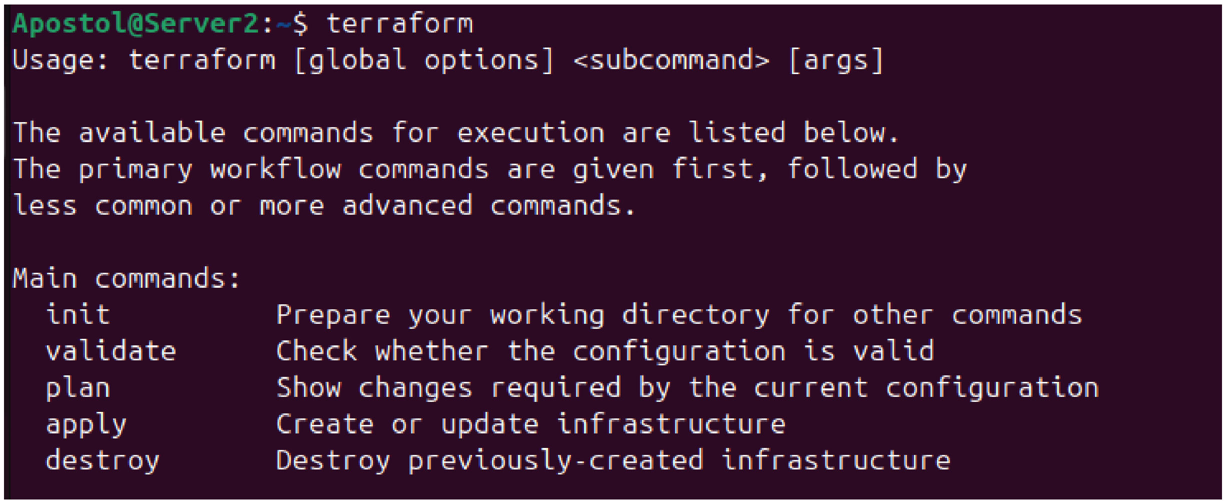


```
PLAY RECAP *****************************************************************************************
192.168.56.104             : ok=0    changed=0    unreachable=1    failed=0    skipped=0    rescued=0
 ignored=0
192.168.56.106             : ok=5    changed=1    unreachable=0    failed=0    skipped=0    rescued=0
 ignored=0
192.168.56.108             : ok=5    changed=1    unreachable=0    failed=0    skipped=0    rescued=0
 ignored=0
192.168.56.113             : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0
```

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



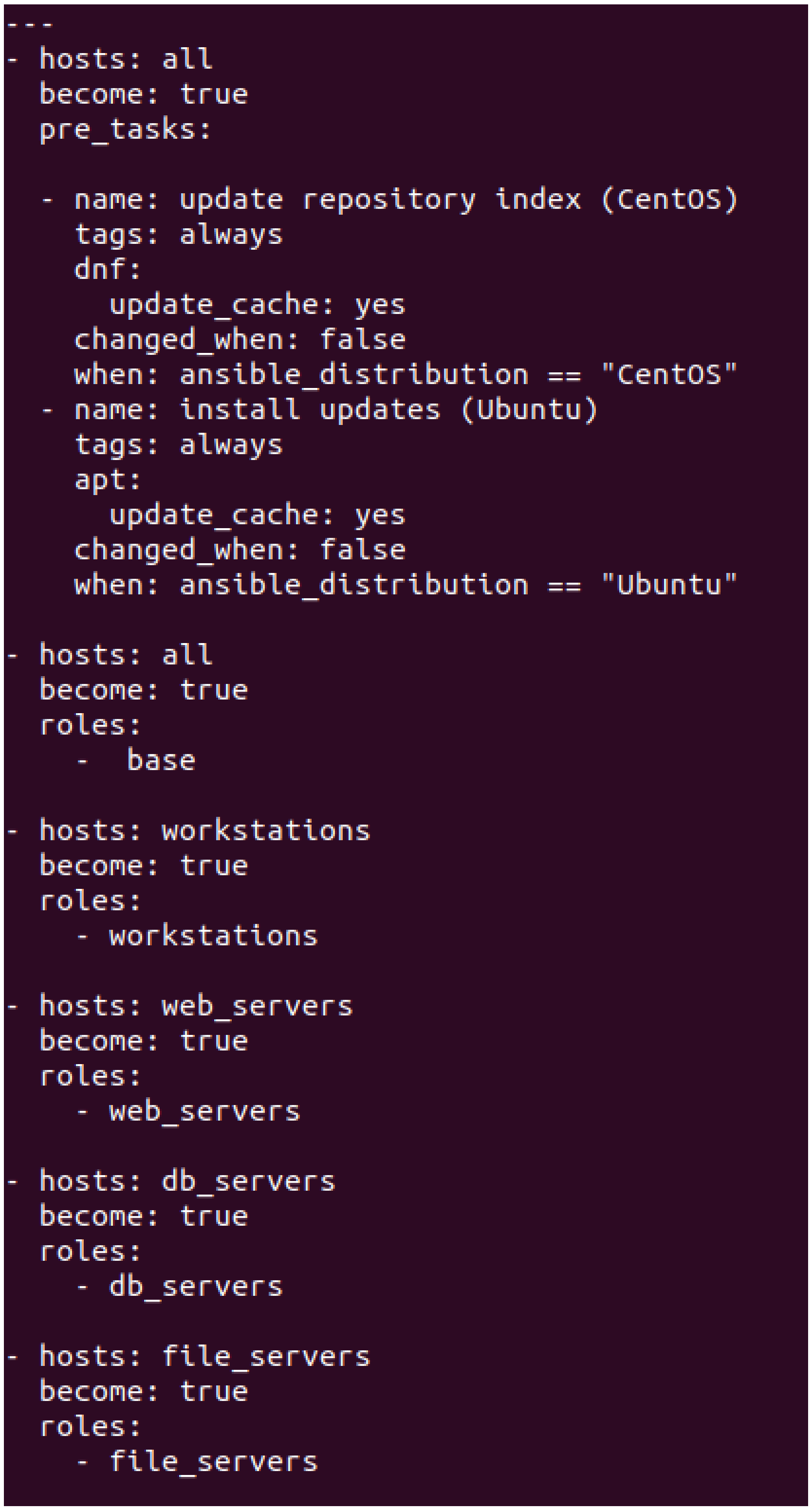
```
Apostol@Server1:~$ terraform
Usage: terraform [global options] <subcommand> [args]

The available commands for execution are listed below.
The primary workflow commands are given first, followed by
less common or more advanced commands.

Main commands:
  init          Prepare your working directory for other commands
  validate      Check whether the configuration is valid
  plan          Show changes required by the current configuration
  apply         Create or update infrastructure
  destroy       Destroy previously-created infrastructure

All other commands:
  console       Try Terraform expressions at an interactive command prompt
  fmt           Reformat your configuration in the standard style
```



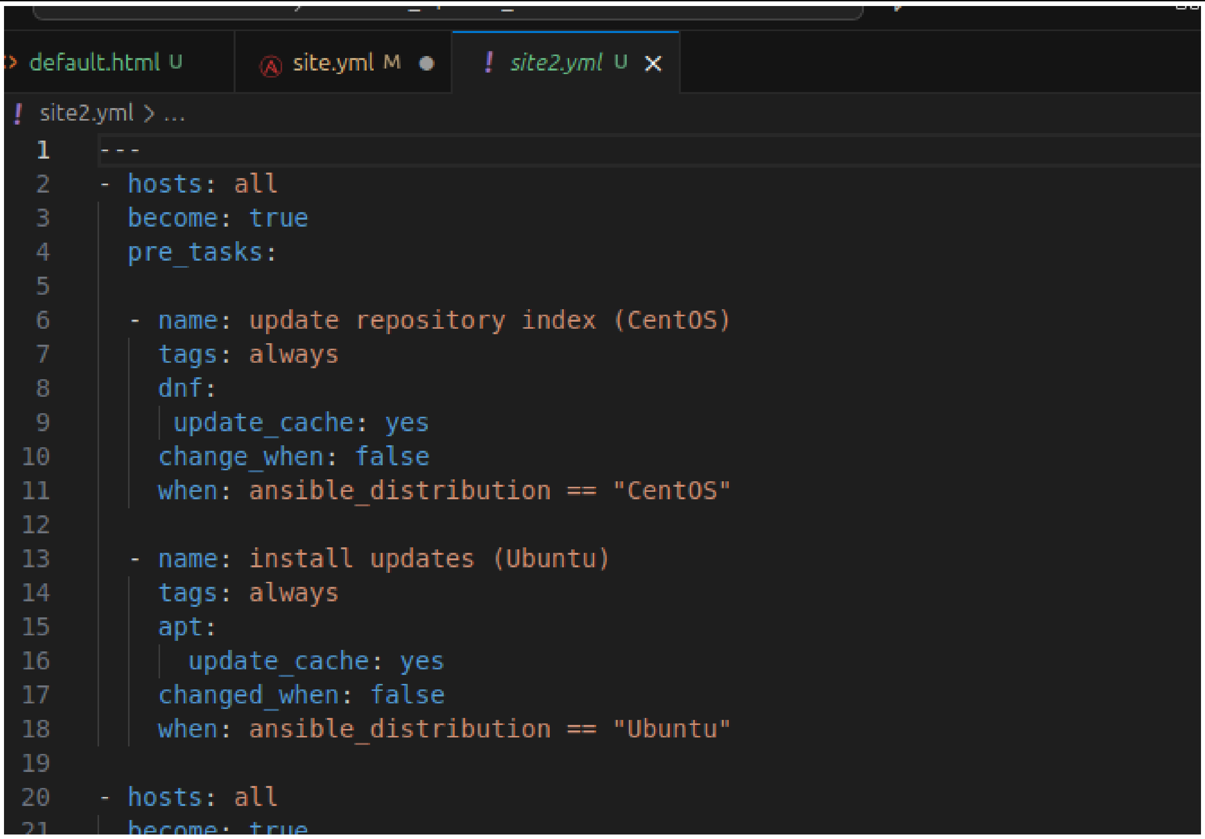
```
Apostol@Server2:~$ terraform
Usage: terraform [global options] <subcommand> [args]

The available commands for execution are listed below.
The primary workflow commands are given first, followed by
less common or more advanced commands.

Main commands:
  init          Prepare your working directory for other commands
  validate      Check whether the configuration is valid
  plan          Show changes required by the current configuration
  apply         Create or update infrastructure
  destroy       Destroy previously-created infrastructure
```

**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)



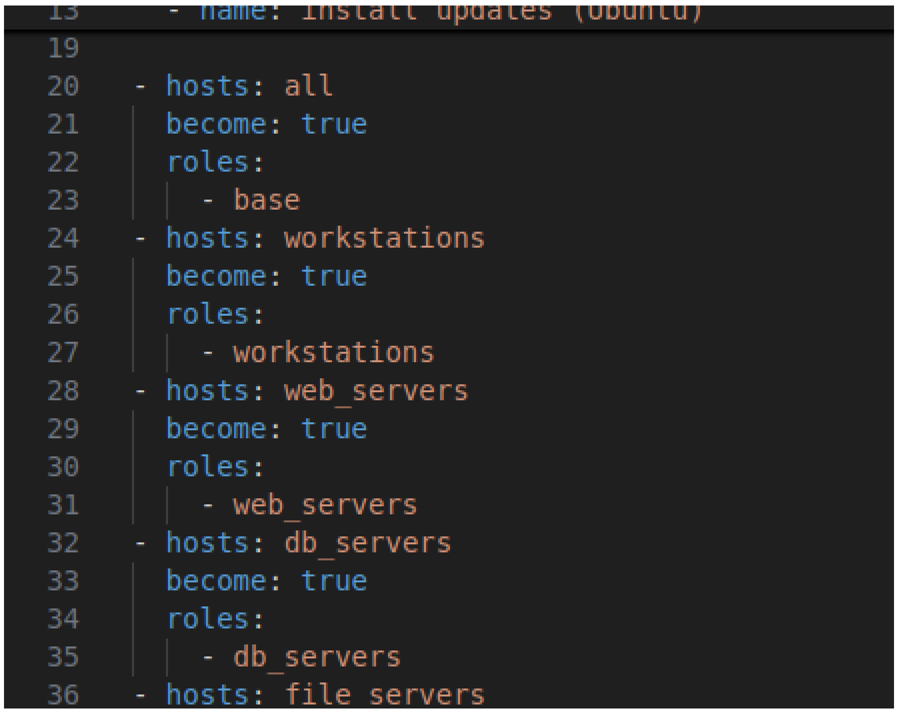
```
---
- hosts: all
  become: true
  pre_tasks:

  - 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
```



default.html U | site.yml M | site2.yml U

site2.yml > ...

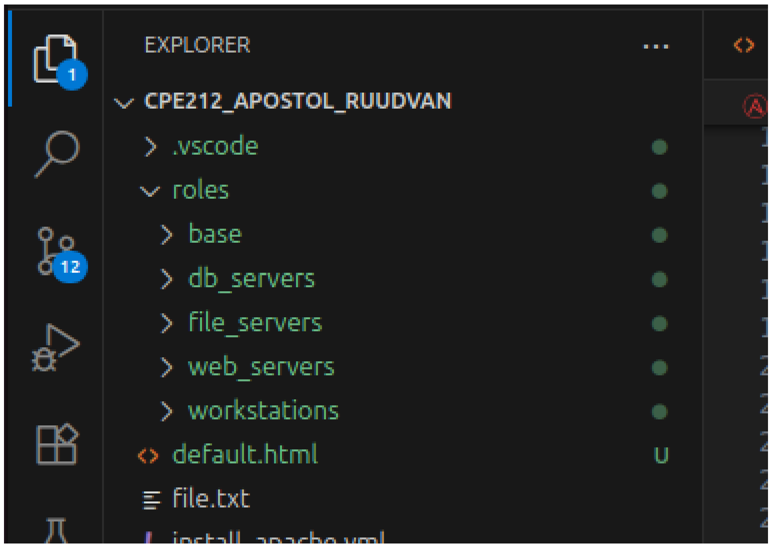
```
---
- hosts: all
  become: true
  pre_tasks:

  - name: update repository index (CentOS)
    tags: always
    dnf:
      update_cache: yes
    change_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
```



```
    - name: install updates (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
```

Save the file and exit.

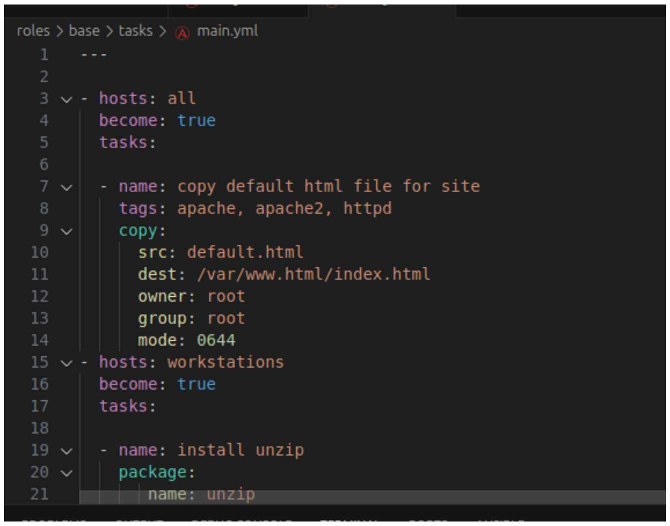
2. Under the same directory, create a new directory and name it roles. Enter the roles directory and create new directories: base, web_servers, file_servers, db_servers and workstations. For each directory, create a directory and name it tasks.



```
EXPLORER
CPE212_APOSTOL_RUUDVAN
  > .vscode
  v roles
    > base
    > db_servers
    > file_servers
    > web_servers
    > workstations
  default.html        U
  file.txt
```

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.

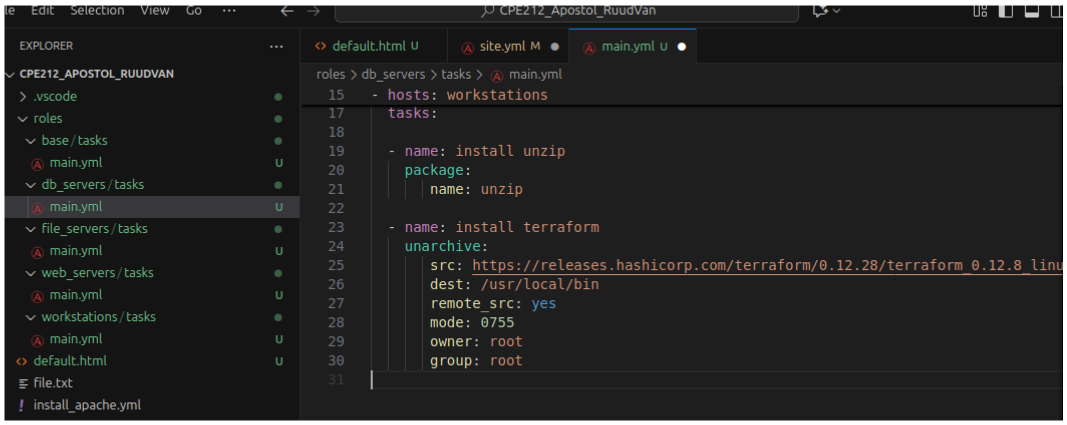
**Base:**



roles > base > tasks > main.yml

```yaml
---

- hosts: all
  become: true
  tasks:

  - name: copy default html file for site
    tags: apache, apache2, httpd
    copy:
      src: default.html
      dest: /var/www.html/index.html
      owner: root
      group: root
      mode: 0644
- hosts: workstations
  become: true
  tasks:

  - name: install unzip
    package:
      name: unzip
```

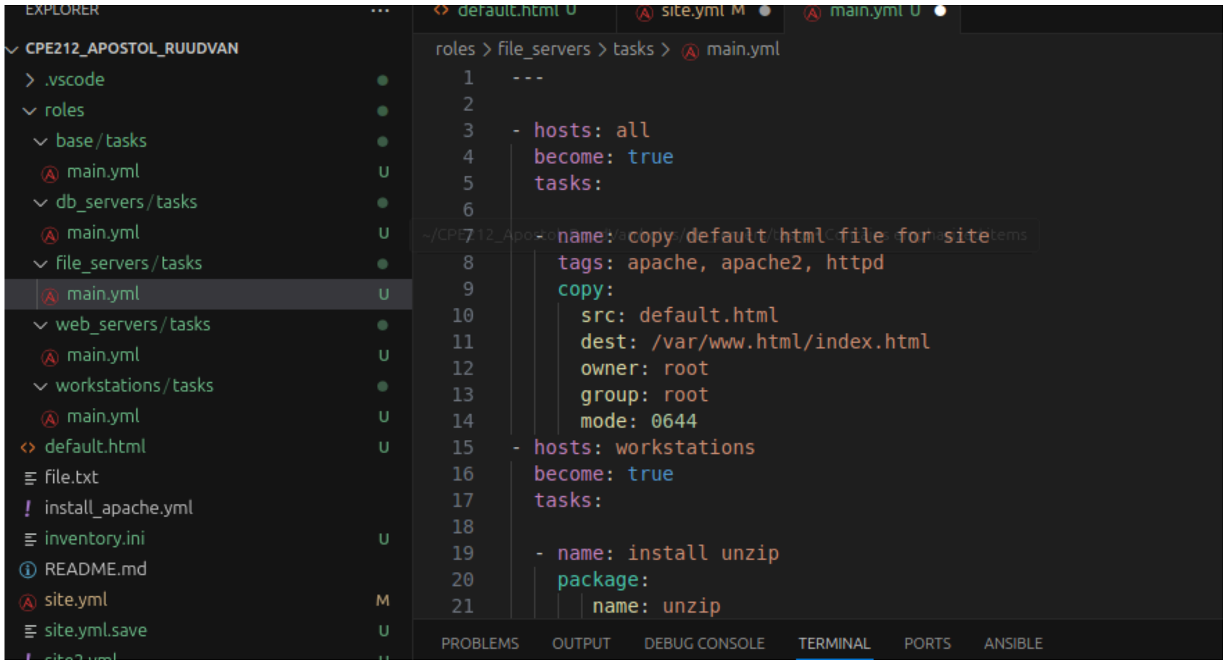
**db_servers:**



roles > db_servers > tasks > main.yml

```yaml
- hosts: workstations
  tasks:

  - name: install unzip
    package:
      name: unzip

  - name: install terraform
    unarchive:
      src: https://releases.hashicorp.com/terraform/0.12.28/terraform_0.12.8_linu
      dest: /usr/local/bin
      remote_src: yes
      mode: 0755
      owner: root
      group: root
```

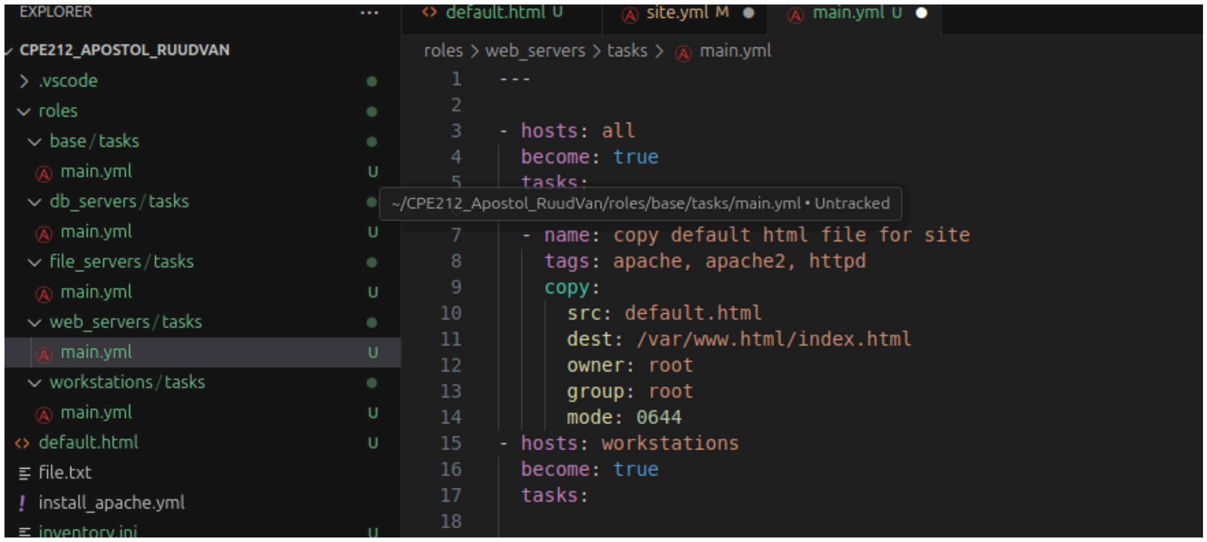
**file_servers:**



roles > file_servers > tasks > main.yml

```yaml
---

- hosts: all
  become: true
  tasks:

  - name: copy default html file for site
    tags: apache, apache2, httpd
    copy:
      src: default.html
      dest: /var/www.html/index.html
      owner: root
      group: root
      mode: 0644
- hosts: workstations
  become: true
  tasks:

  - name: install unzip
    package:
      name: unzip
```

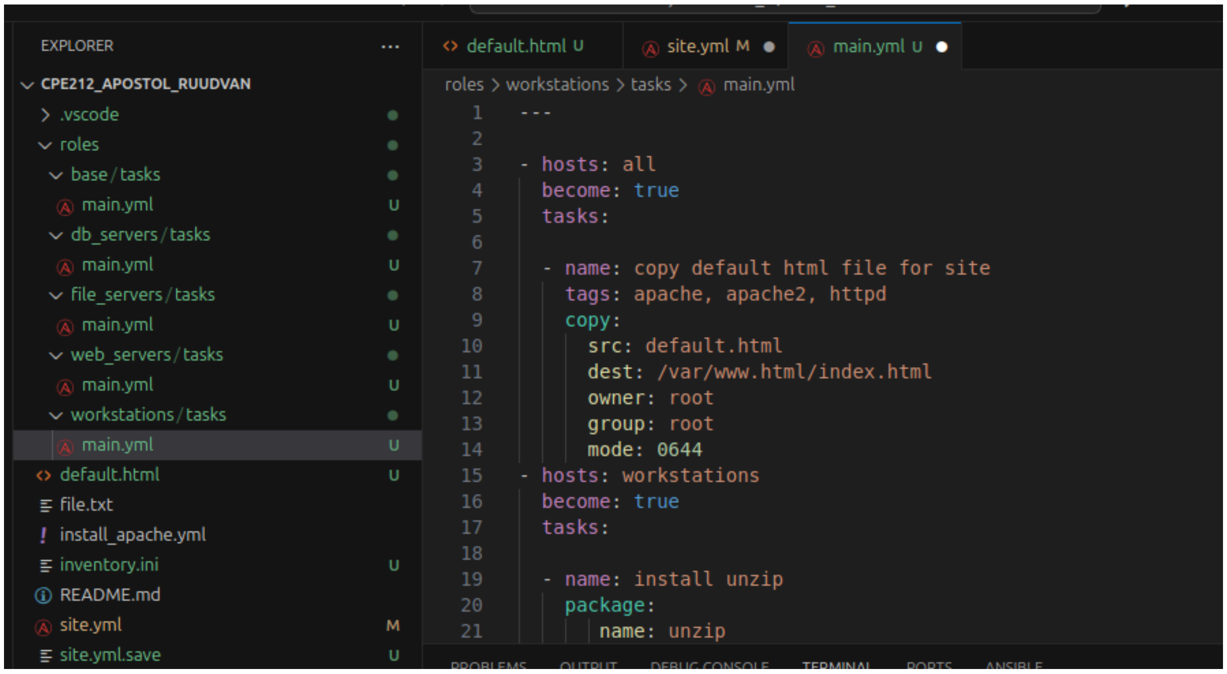
**web_servers:**



roles > web_servers > tasks > main.yml

```yaml
---

- hosts: all
  become: true
  tasks:
~/CPE212_Apostol_RuudVan/roles/base/tasks/main.yml • Untracked
    - name: copy default html file for site
      tags: apache, apache2, httpd
      copy:
        src: default.html
        dest: /var/www.html/index.html
        owner: root
        group: root
        mode: 0644
- hosts: workstations
  become: true
  tasks:

```

**workstations:**

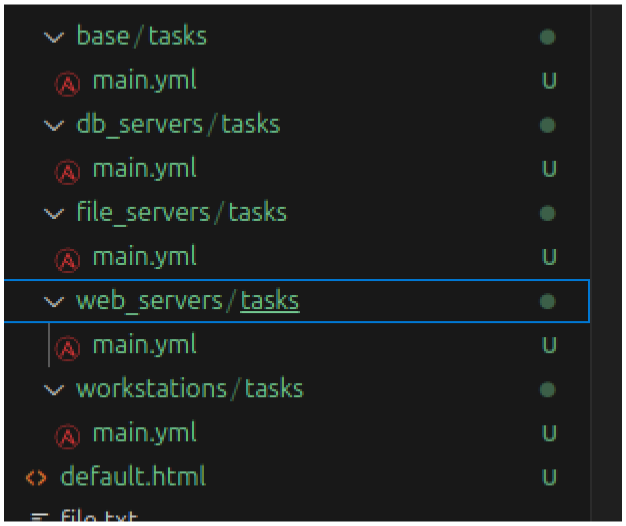


```
EXPLORER
CPE212_APOSTOL_RUUDVAN
> .vscode
v roles
  v base/tasks
    main.yml          U
  v db_servers/tasks
    main.yml          U
  v file_servers/tasks
    main.yml          U
  v web_servers/tasks
    main.yml          U
  v workstations/tasks
    main.yml          U
  default.html        U
  file.txt
  install_apache.yml
  inventory.ini       U
  README.md
  site.yml            M
  site.yml.save       U
```

default.html U | site.yml M | main.yml U

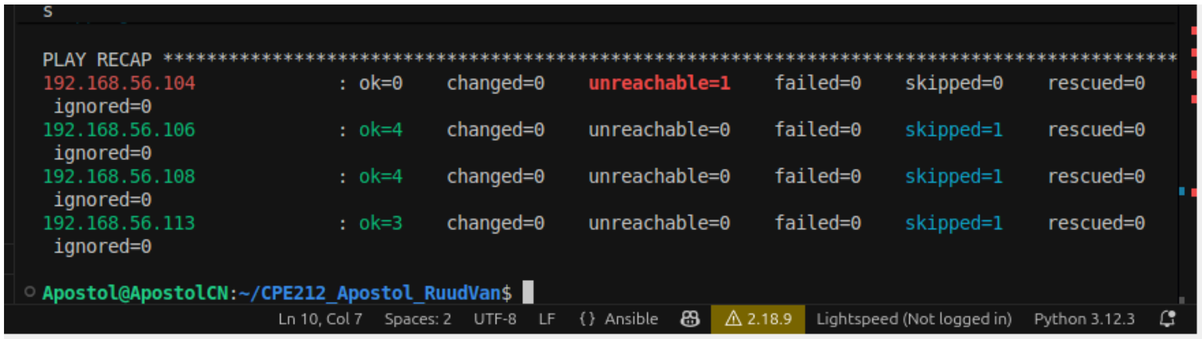
roles > workstations > tasks > main.yml

```yaml
---

- hosts: all
  become: true
  tasks:

  - name: copy default html file for site
    tags: apache, apache2, httpd
    copy:
      src: default.html
      dest: /var/www.html/index.html
      owner: root
      group: root
      mode: 0644
- hosts: workstations
  become: true
  tasks:

  - name: install unzip
    package:
      name: unzip
```



```
v base/tasks
  main.yml          U
v db_servers/tasks
  main.yml          U
v file_servers/tasks
  main.yml          U
v web_servers/tasks
  main.yml          U
v workstations/tasks
  main.yml          U
default.html        U
```

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



```
PLAY RECAP *********************************************************************************
192.168.56.104             : ok=0    changed=0    unreachable=1    failed=0    skipped=0    rescued=0
 ignored=0
192.168.56.106             : ok=4    changed=0    unreachable=0    failed=0    skipped=1    rescued=0
 ignored=0
192.168.56.108             : ok=4    changed=0    unreachable=0    failed=0    skipped=1    rescued=0
 ignored=0
192.168.56.113             : ok=3    changed=0    unreachable=0    failed=0    skipped=1    rescued=0
 ignored=0

Apostol@ApostolCN:~/CPE212_Apostol_RuudVan$
```

Ln 10, Col 7   Spaces: 2   UTF-8   LF   {} Ansible   ⚠ 2.18.9   Lightspeed (Not logged in)   Python 3.12.3

**Reflections:**

Answer the following:

1. **What is the importance of creating roles?**
   - In essence, Ansible roles transform automation from a collection of individual scripts into a structured, scalable, and maintainable framework, making it a cornerstone for effective infrastructure management.
2. **What is the importance of managing files?**
   - Managing files in an Ansible playbook is essential because it ensures consistent configuration across systems, automates the deployment of necessary files like templates and scripts, and reduces the risk of human error.