

## Question:

Create an Ansible role that installs Python3 and deploys a custom configuration file for the application. Task:

- i. Create an Ansible Role named `python_app_deploy` with the correct directory structure.
- ii. Ensure Python3 is installed as part of the role.
- iii. Place a template file named `config.yml.j2` inside the `templates/` directory.

Use the template module in the role to copy `config.yml.j2` to `/etc/python_app/config.yml` on the target machine.

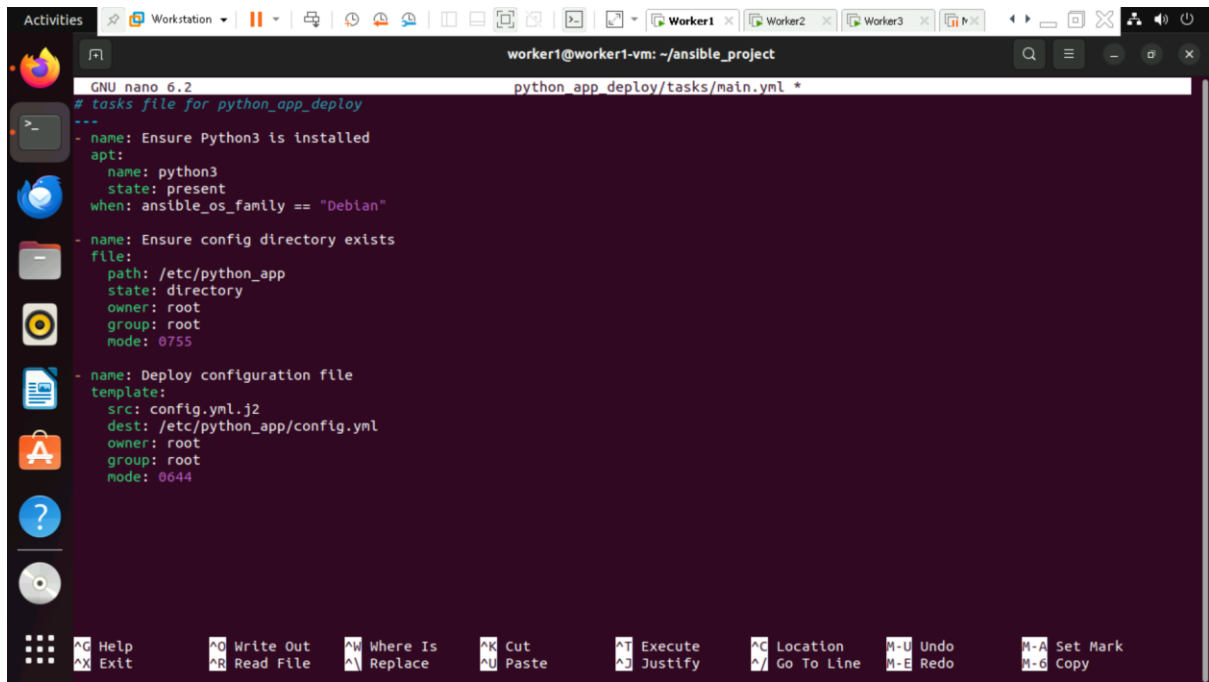
## Solution:

1. Create the Role on Worker1

```
worker1@worker1-vm: ~/ansible_project
worker1@worker1-vm:~/ansible$ mkdir -p ~/ansible_project && cd ~/ansible_project
ansible-galaxy init python_app_deploy
- Role python_app_deploy was created successfully
worker1@worker1-vm:~/ansible_project$
```

2. Add Task to Install Python3

```
worker1@worker1-vm:~/ansible_project$ nano python_app_deploy/tasks/main.yml
worker1@worker1-vm:~/ansible_project$
```

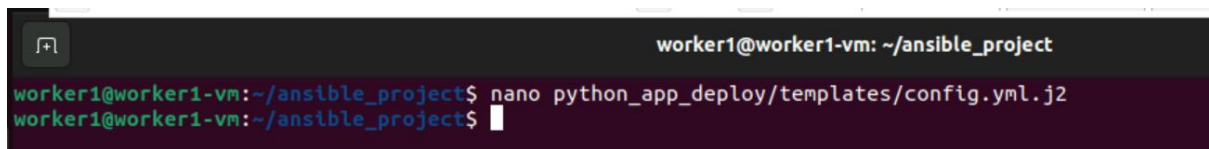


```
GNU nano 6.2 python_app_deploy/tasks/main.yml *
# tasks file for python_app_deploy
---
- name: Ensure Python3 is installed
  apt:
    name: python3
    state: present
    when: ansible_os_family == "Debian"

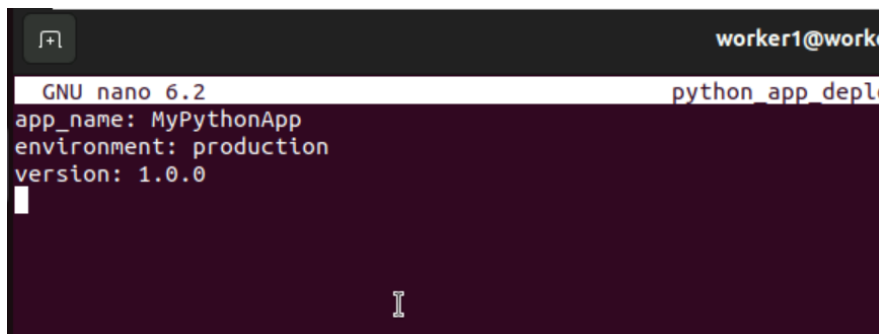
- name: Ensure config directory exists
  file:
    path: /etc/python_app
    state: directory
    owner: root
    group: root
    mode: 0755

- name: Deploy configuration file
  template:
    src: config.yml.j2
    dest: /etc/python_app/config.yml
    owner: root
    group: root
    mode: 0644
```

### 3. Add Template File

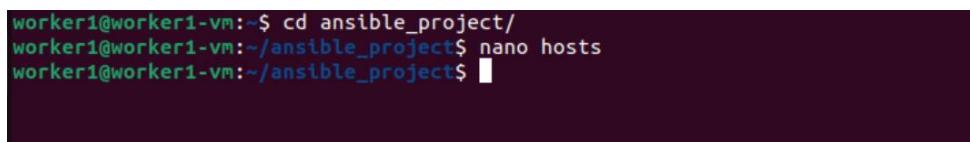


```
worker1@worker1-vm: ~/ansible_project
worker1@worker1-vm:~/ansible_project$ nano python_app_deploy/templates/config.yml.j2
worker1@worker1-vm:~/ansible_project$
```

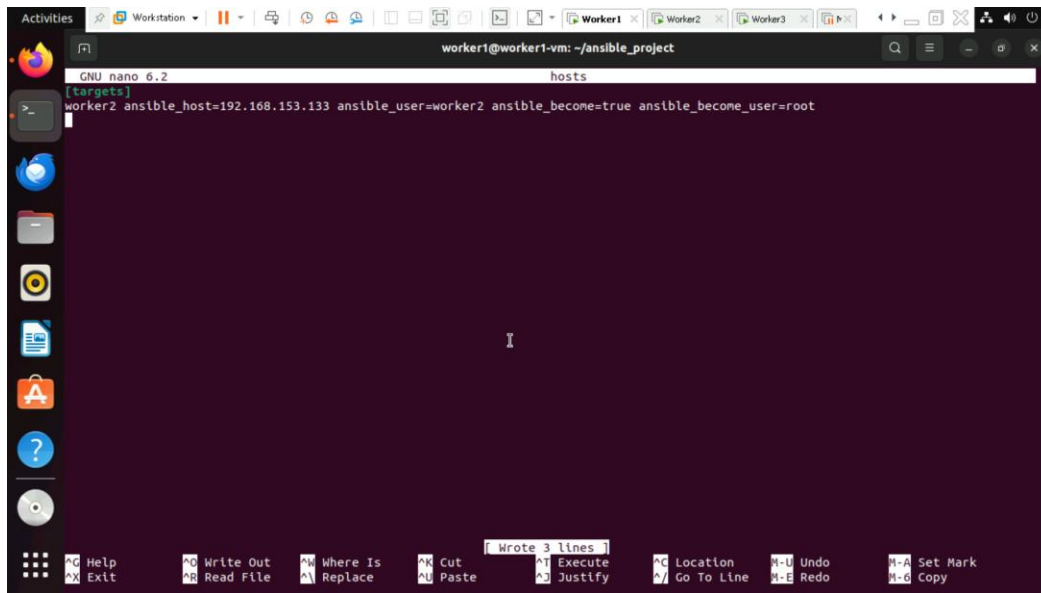


```
GNU nano 6.2 python_app_deploy/templates/config.yml.j2
app_name: MyPythonApp
environment: production
version: 1.0.0
```

### 4. Create Inventory file



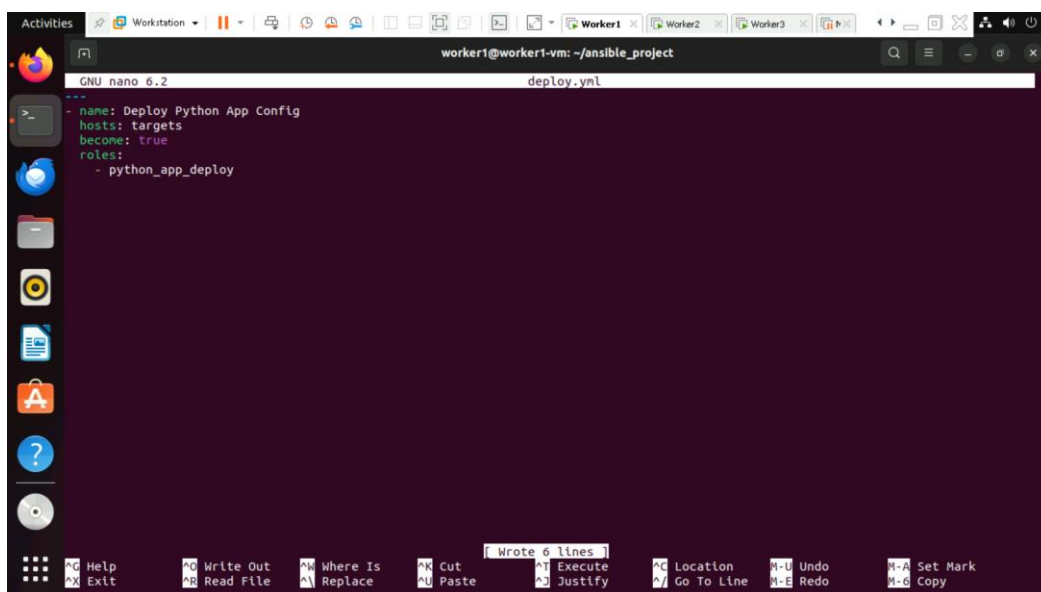
```
worker1@worker1-vm:~$ cd ansible_project/
worker1@worker1-vm:~/ansible_project$ nano hosts
worker1@worker1-vm:~/ansible_project$
```



A screenshot of a terminal window titled "worker1@worker1-vm: ~/ansible\_project". The terminal shows the nano 6.2 editor editing a file named "hosts". The content of the file is: `[targets]` and `worker2 ansible_host=192.168.153.133 ansible_user=worker2 ansible_become=true ansible_become_user=root`. The bottom status bar of the nano editor indicates "Wrote 3 lines".

## 5. Create Playbook

```
worker1@worker1-vm:~/ansible_project$ nano deploy.yml
worker1@worker1-vm:~/ansible_project$
```



A screenshot of a terminal window titled "worker1@worker1-vm: ~/ansible\_project". The terminal shows the nano 6.2 editor editing a file named "deploy.yml". The content of the file is: `---`, `- name: Deploy Python App Config`, `hosts: targets`, `become: true`, `roles:`, and `- python_app_deploy`. The bottom status bar of the nano editor indicates "Wrote 6 lines".

## 6. Run the playbook

```
Activities Workstation Worker1 Worker2 Worker3  
worker1@worker1-vm: ~/ansible_project  
worker1@worker1-vm:~/ansible_project$ ansible-playbook -i hosts deploy.yml --ask-become-pass  
BECOME password:  
PLAY [Deploy Python App Config] *****  
TASK [Gathering Facts] *****  
ok: [worker2]  
TASK [python_app_deploy : Ensure Python3 is installed] *****  
ok: [worker2]  
TASK [python_app_deploy : Ensure config directory exists] *****  
changed: [worker2]  
TASK [python_app_deploy : Deploy configuration file] *****  
changed: [worker2]  
PLAY RECAP *****  
worker2 : ok=4 changed=2 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0  
worker1@worker1-vm:~/ansible_project$
```