Name: Chaudhari Ganesh Dadabhau

Deployment of Webserver on AWS through Ansible

TASK 2

1. Provision of EC2 instance through Ansible
2. Retrive the public ip of instance by using concept of dynamic inventory
3. Configure the web server through Ansible

Prerequisite: 1.For installation and configuration of ansible with stable version you check this link

https://www.linkedin.com/pulse/integration-ansible-docker-ganesh-chaudhari

2. You need to create AWS account

https://signin.aws.amazon.com/signin?redirect_uri=https%3A%2F%2Fconsole.aws.amazon.com%2Fconsole%2Fhome%3Fstate%3DhashArgs%2523%26isauthcode%3Dtrue&client_id=arn%3Aaws%3Aiam%3A%3A015428540659%3Auser%2Fhomepage&forceMobileApp=0&code_challenge=9DSc6iRFkHWe5V2aeBUIWBlIgOwGKd8gPPDtyXhYSh8&code_challenge_method=SHA-256

Lets starts,

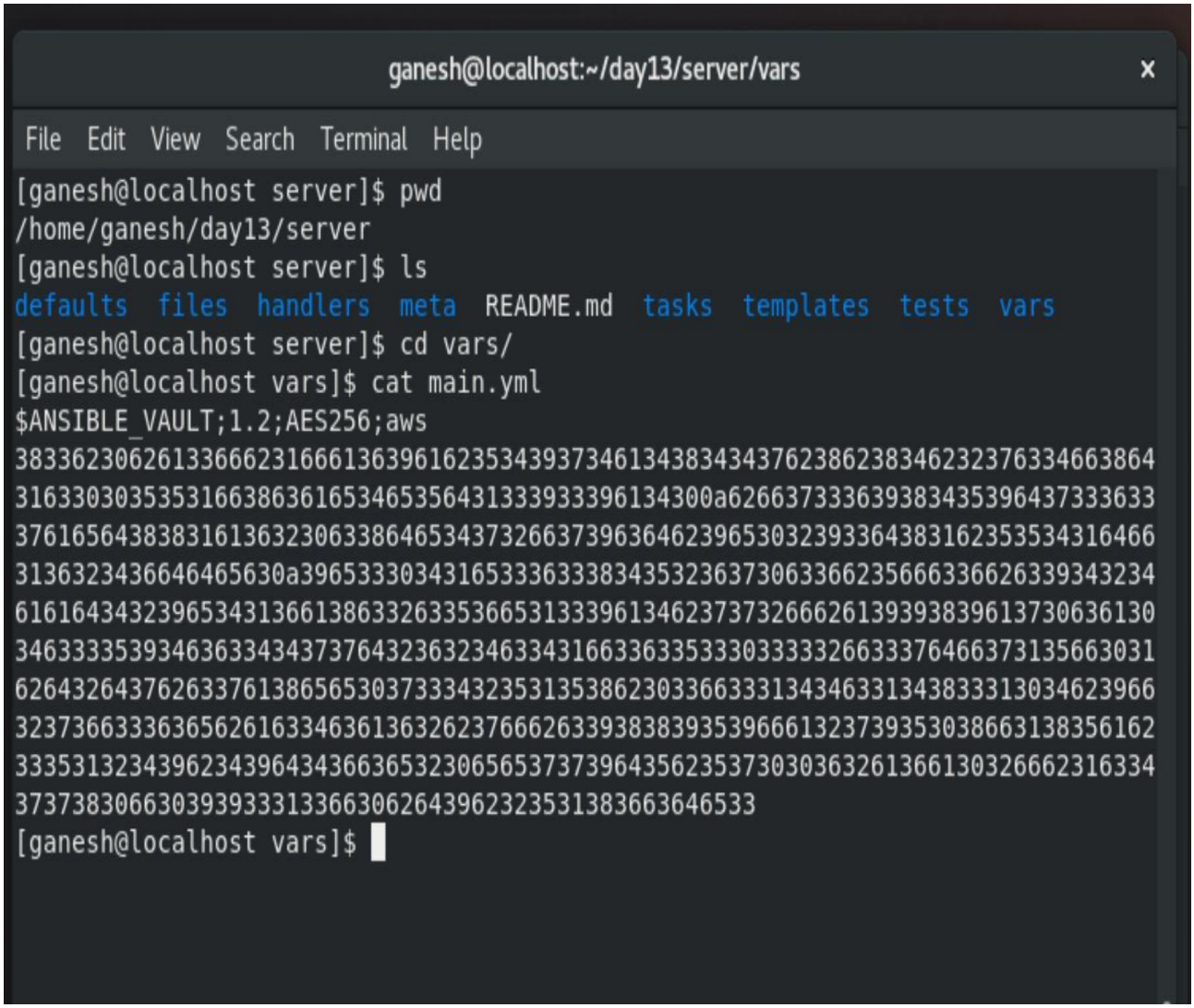
1. Provision of EC2 instance

First you need to login with normal user because login with root user isnot good practice. After that create ansible.cfg file for normal user in home directory and install boto and boto3 using pip3 which are used for aws connection . Write following details because aws uses ec2-user as user and uses private key with pem extension as password and ec-user is normal so we need to give root access for some tasks hence privillege escalation is created.

```
File  Edit  View  Search  Terminal  Help
[defaults]
inventory /home/ganesh/day13/host
host_key_checking=false
remote_user=ec2-user
private_key_file=/home/ganesh/ec2.pem


[privilege_escalation]
become = true
become_method=sudo
become_user=root
#become_ask_pass = false
```

After create one Role using **ansible-galaxy init server** go into **server/tasks/main.yml** and write following YAML script and write variable like access key and secret key in **server/vars/main.yml** and encrypt the main.yml using **ansible-vault encrypt --vault-id aws@prompt main.yml** like following

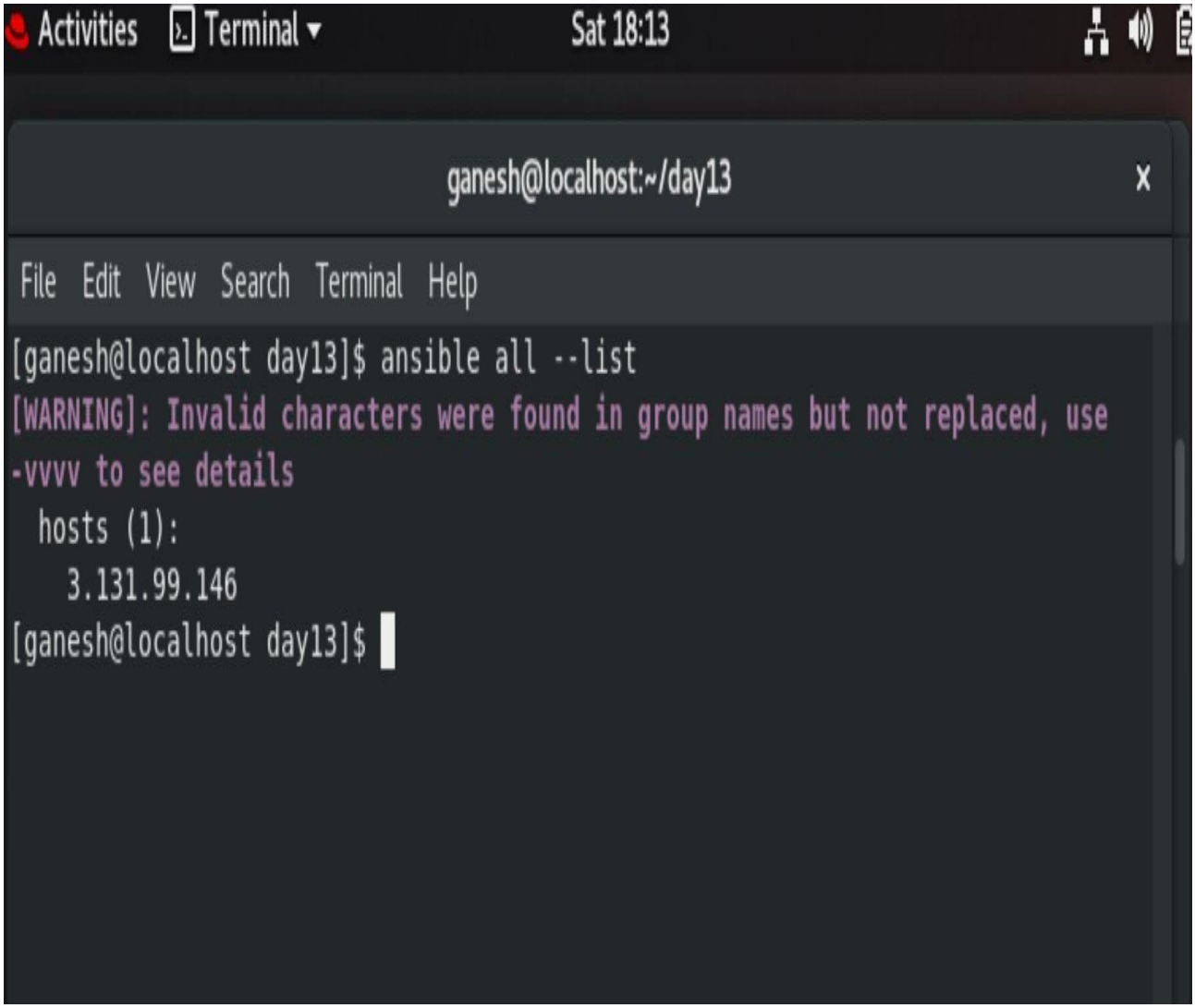
```
ganesh@localhost:~/day13/server/tasks
File  Edit  View  Search  Terminal  Help
[ganesh@localhost tasks]$ pwd
/home/ganesh/day13/server/tasks
[ganesh@localhost tasks]$ cat main.yml
---
# tasks file for server
- ec2:
    key_name: ec2
    instance_type: t2.micro
    image: ami-0a54aef4ef3b5f881
    wait: yes
    group_id: sg-07a0819de46c60076
    count: 1
    state: present
    vpc_subnet_id: subnet-37f5f24d
    assign_public_ip: yes
    region: us-east-2
    aws_access_key: "{{ accessk }}"
    aws_secret_key: "{{secretk}}"


[ganesh@localhost tasks]$
```



2. Retrive public ip dynamically

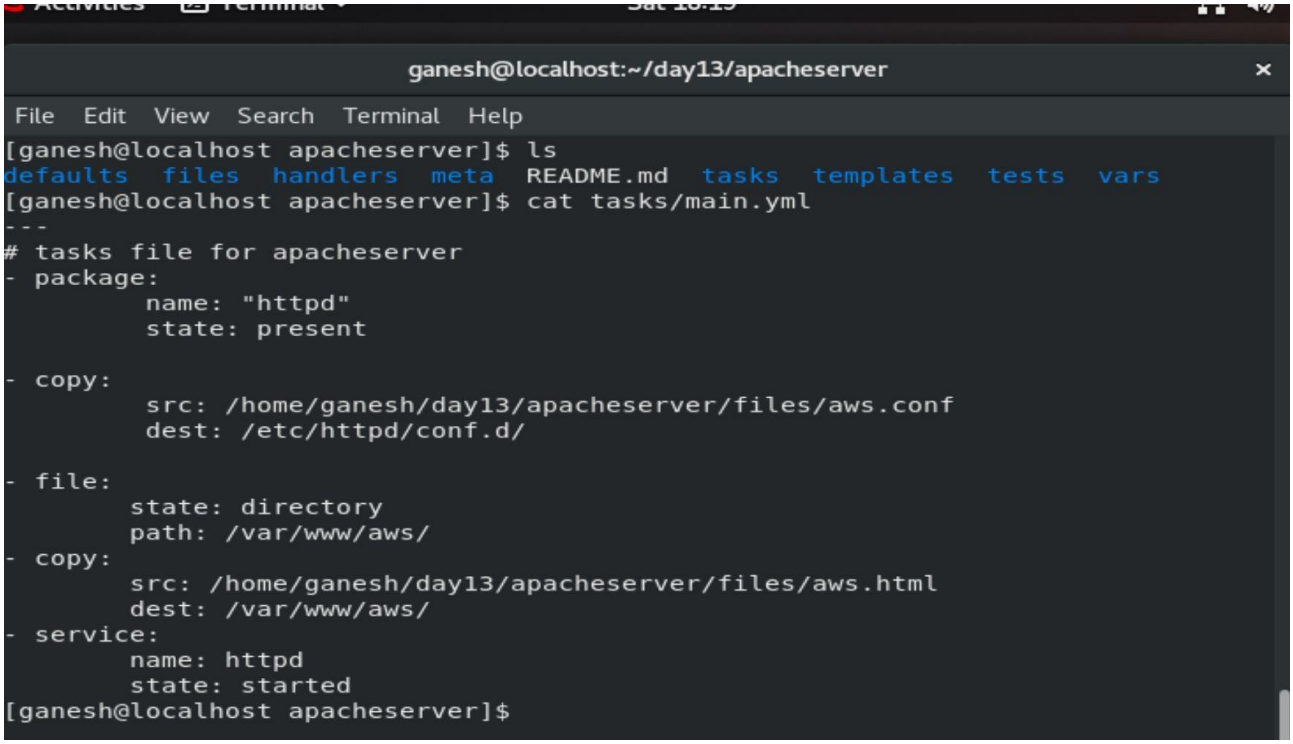
For dynamic inventory pull ec2.py python script for fetching public ip of running instance . For that use wget https://raw.githubusercontent.com/ansible/ansible/stable-2.9/contrib/inventory/ec2.py then change permission chmod +x ec2.py. Modify ec2.py syntax. Then export AWS_ACCESS_KEY_ID="" and AWS_SECRET_ACCESS_KEY="" use this link for reference **https://docs.ansible.com/ansible/latest/user_guide/intro_dynamic_inventory.html#inventory-script-example-aws-ec2**

Use ansible all --list

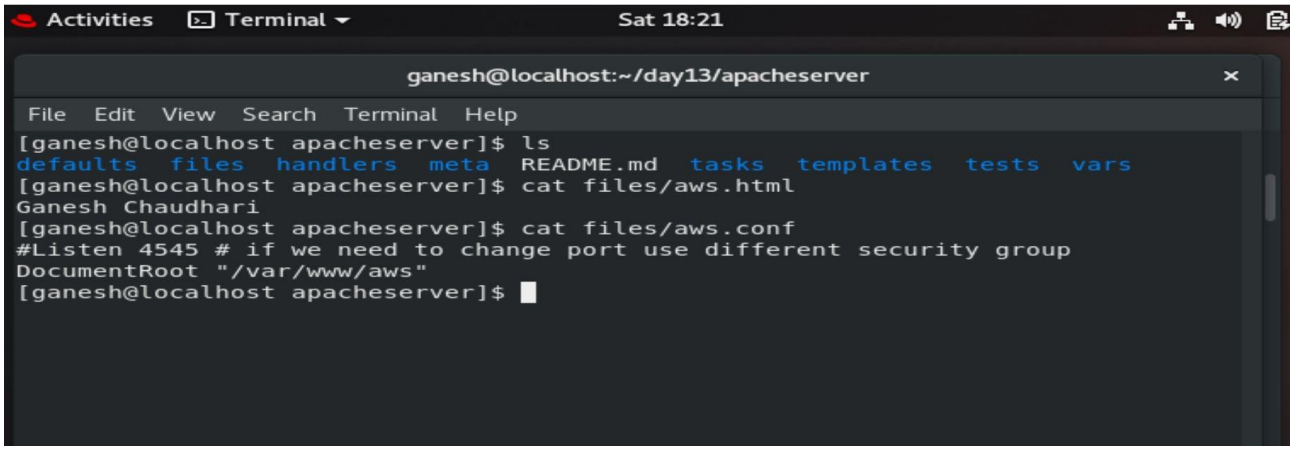


3. Configuration of webserver

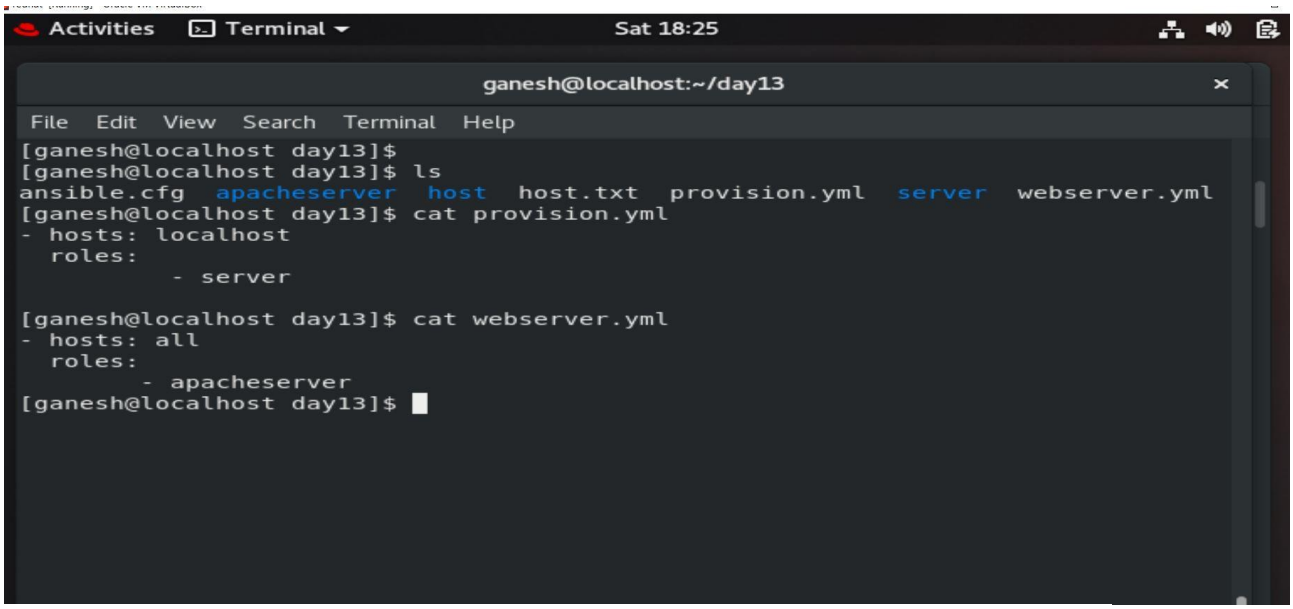
Create another Role apacheserver using ansible-galaxy init apacheserver and write apacheserver/tasks/main.yml



create static files in apacheserver/files like aws.conf which is configuration file of httpd and aws.html it is html page.



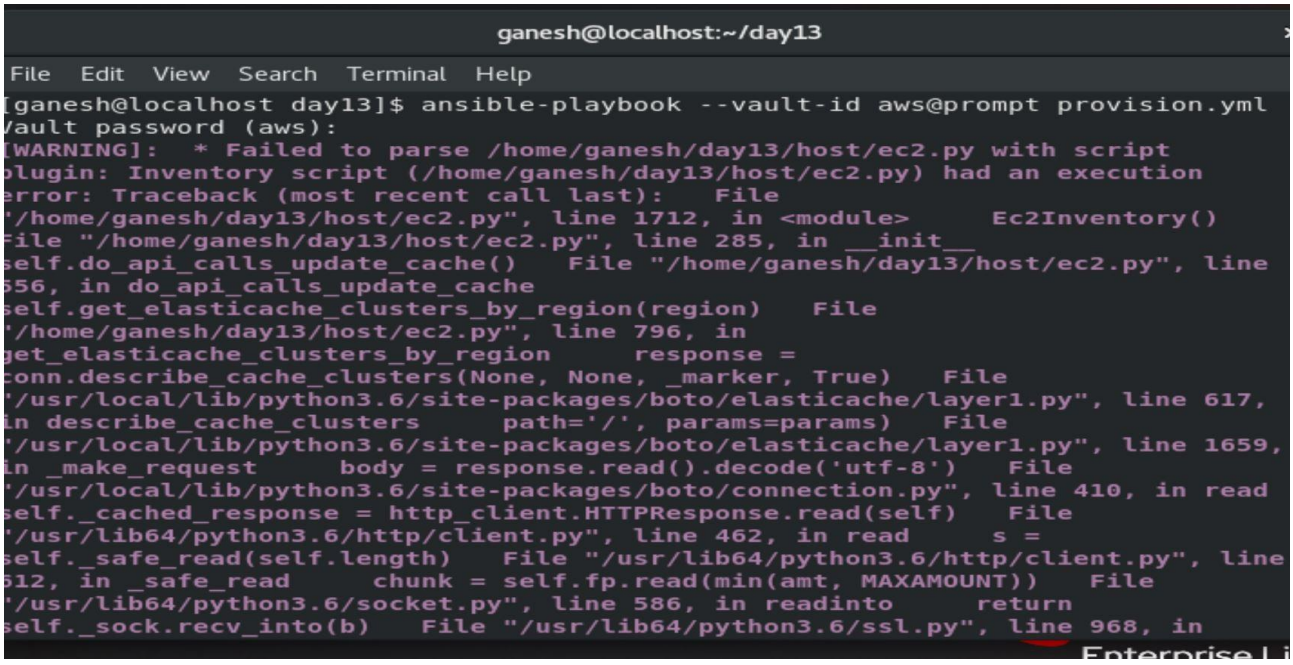
After that create two provision.yml for provision of EC2 and webserver.yml for configuration of httpd like

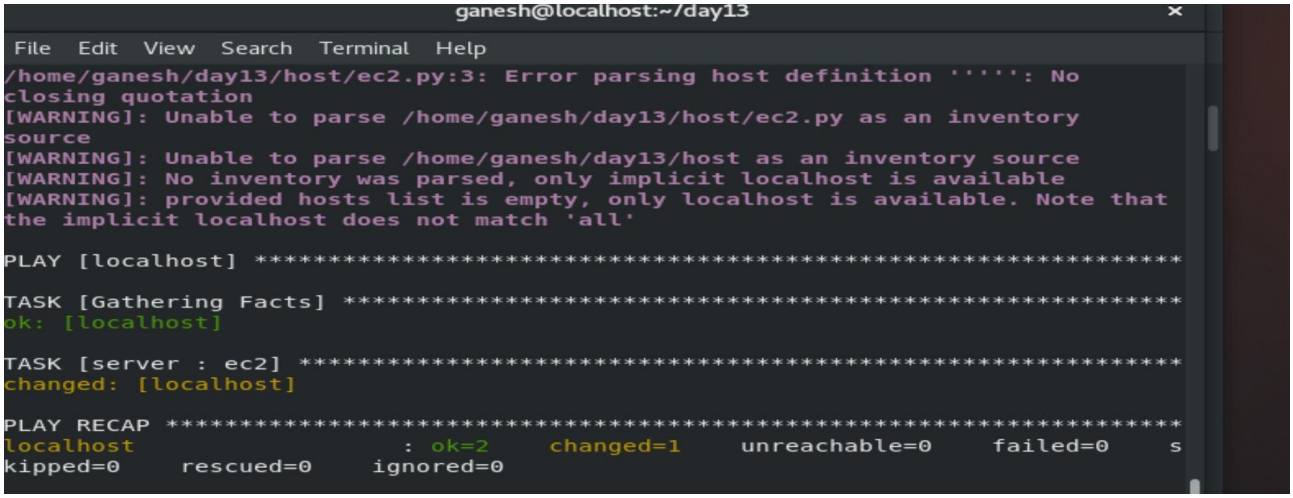


All coding done only we need to run

ansible-playbook --vault-id aws@prompt provision.yml

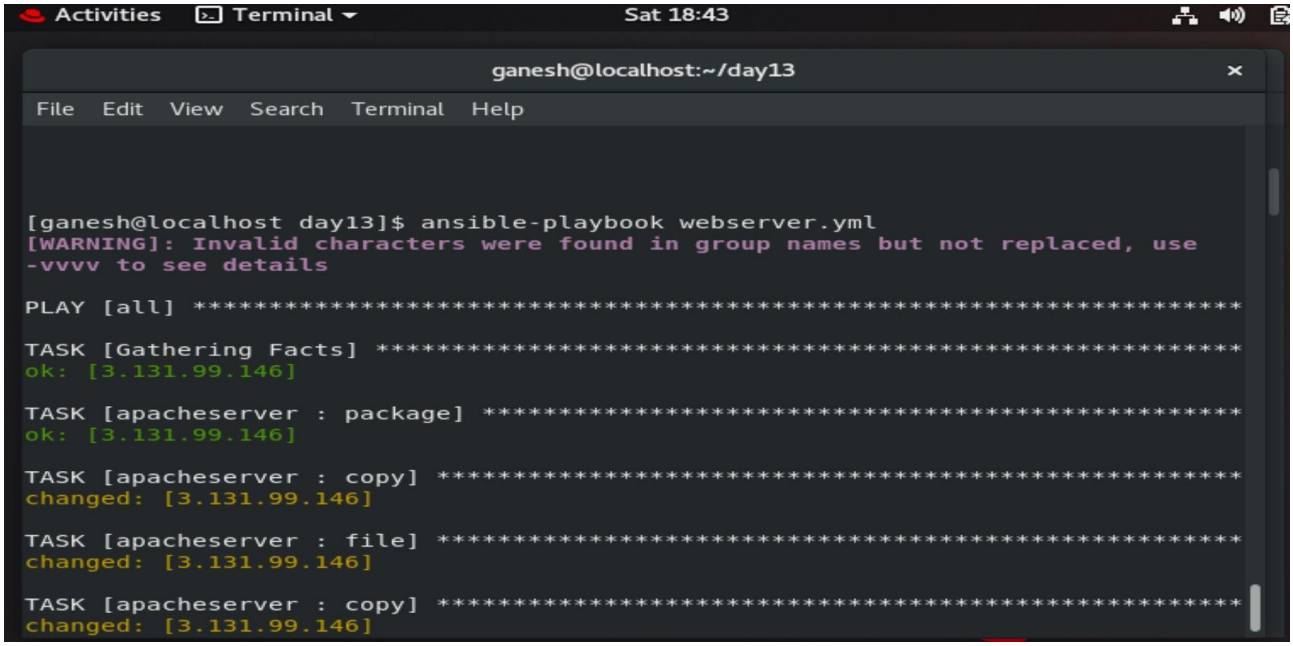
Output:

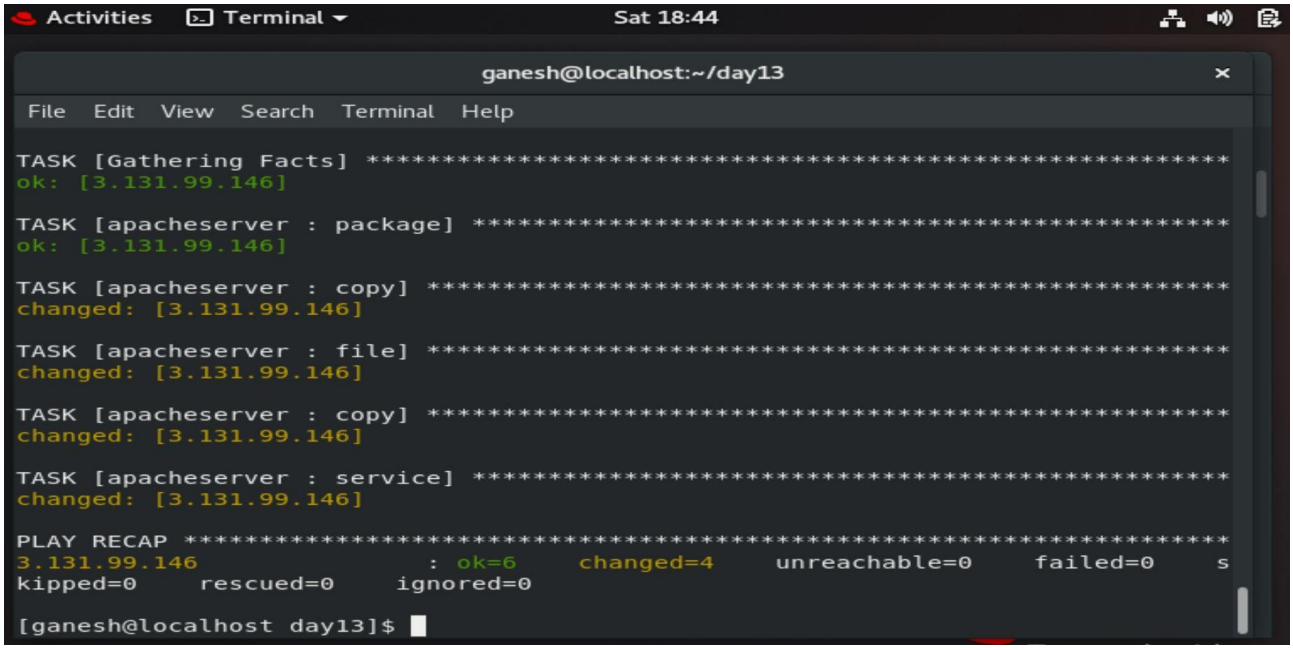


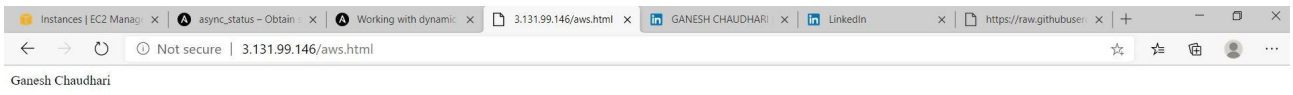


ansible-playbook webserver.yml

Output:







Task 2 successfully completed