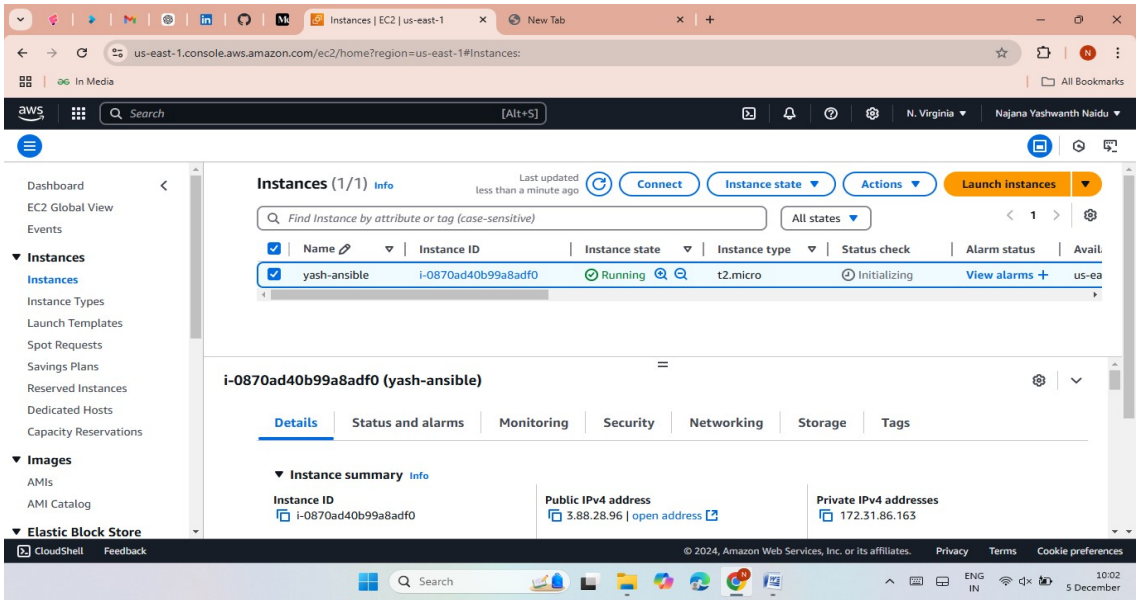
PROJECT

Yashwanth Naidu

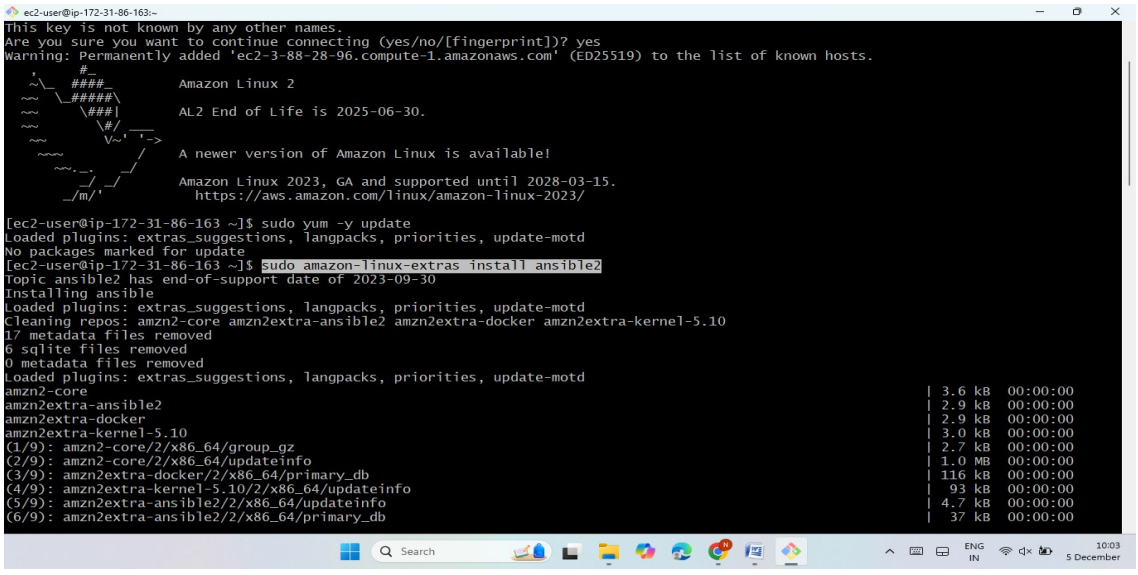
Create and deploy a static and dynamic application infrastructure with ansible-IAC (infrastructure as a code).

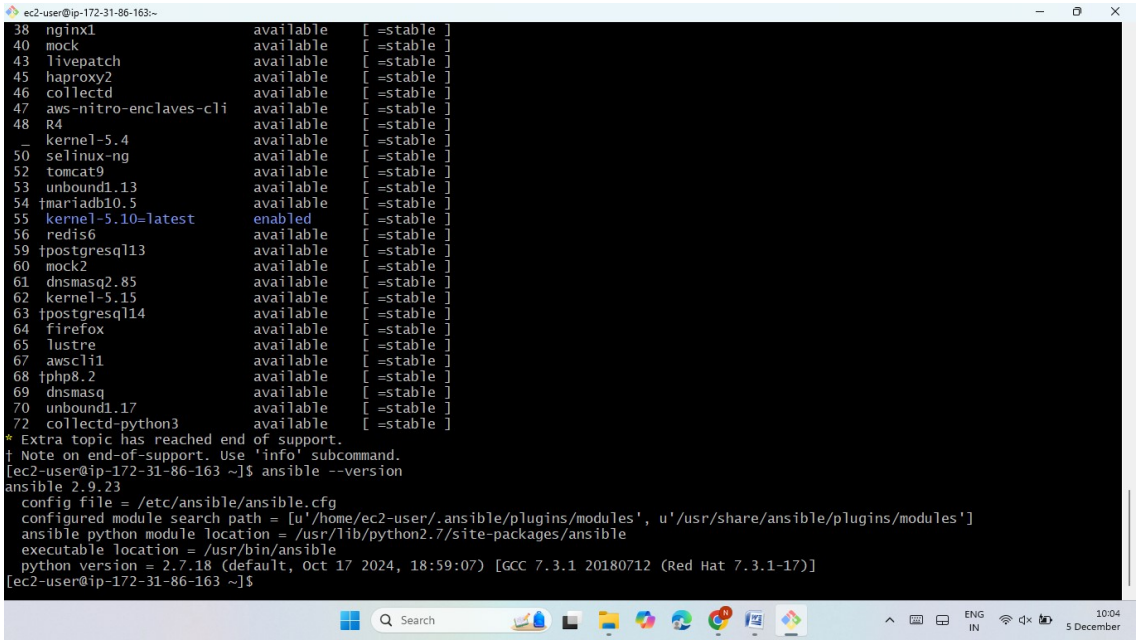
METHOD-1

- Launch an Instances along with required ports.

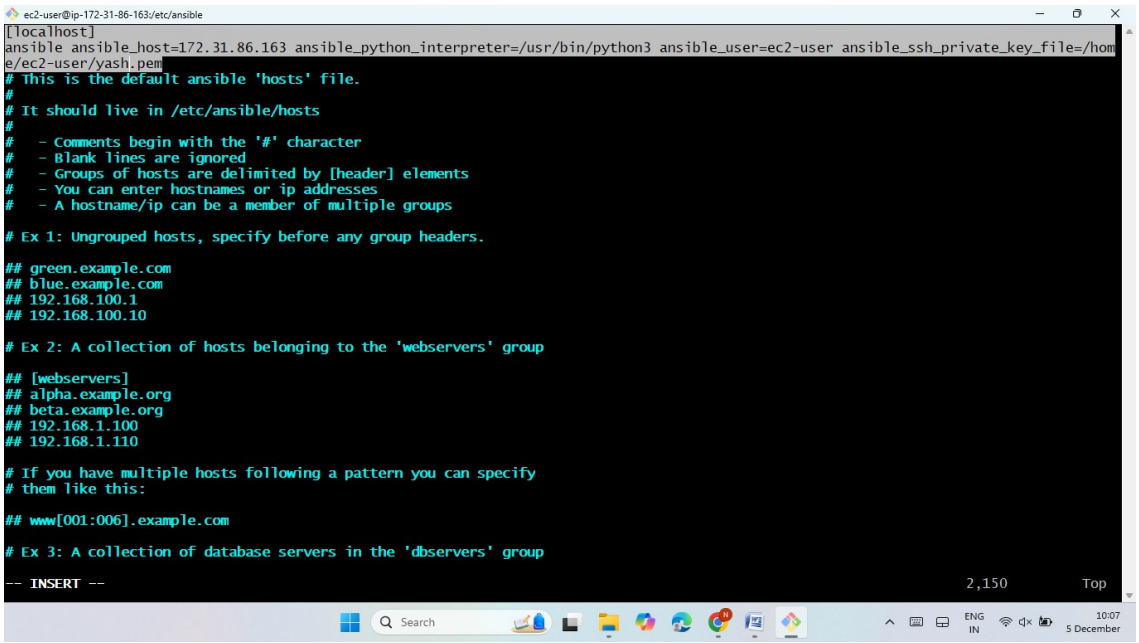


- Now connect to GitBash then install ansible2 with the command.
- Sudo amazon-linux-extras install ansible2

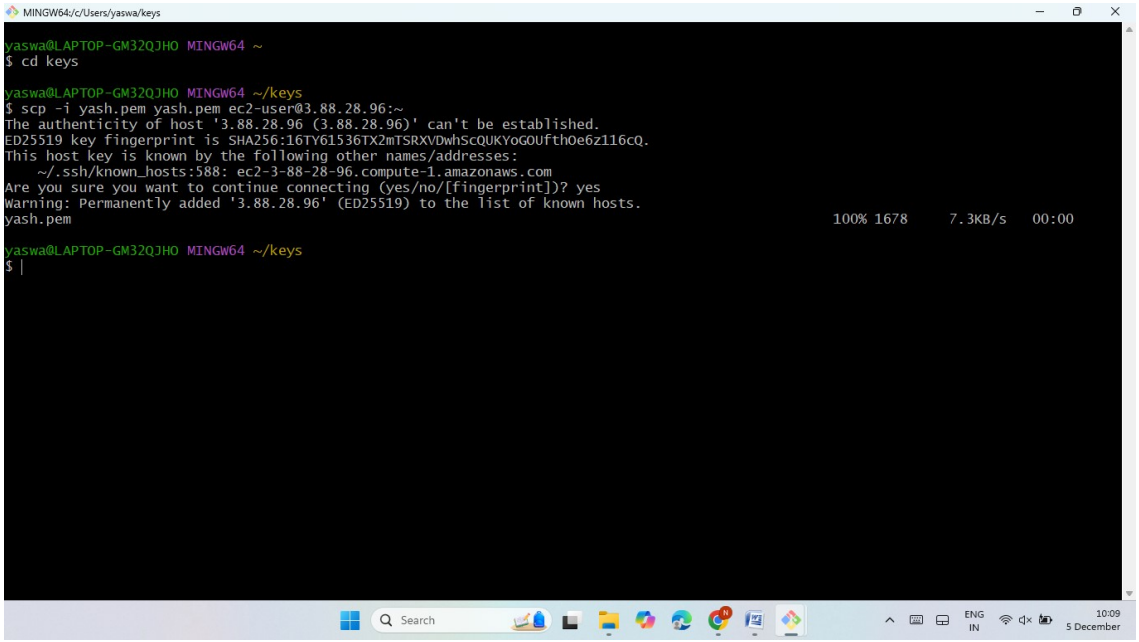




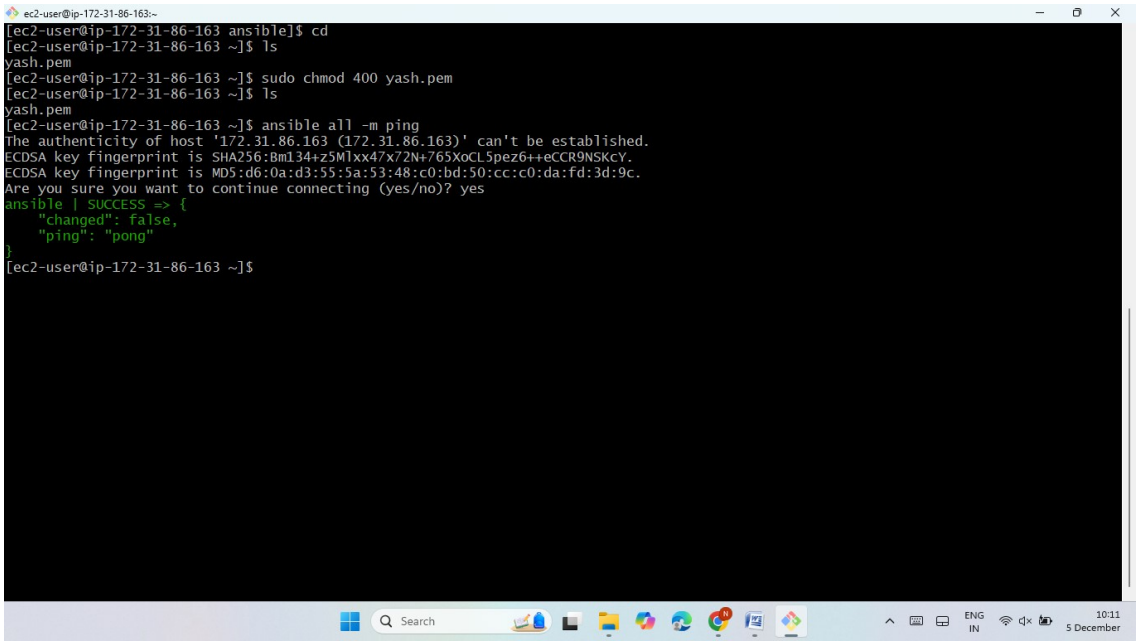
- Now Edit host file (path= /etc/ansible)
  [localhost]
  ansible ansible_host=10.0.49.164 ansible_python_interpreter=/usr/bin/python3
  ansible_user=ec2-user ansible_ssh_private_key_file=/home/ec2-user/yash.pem



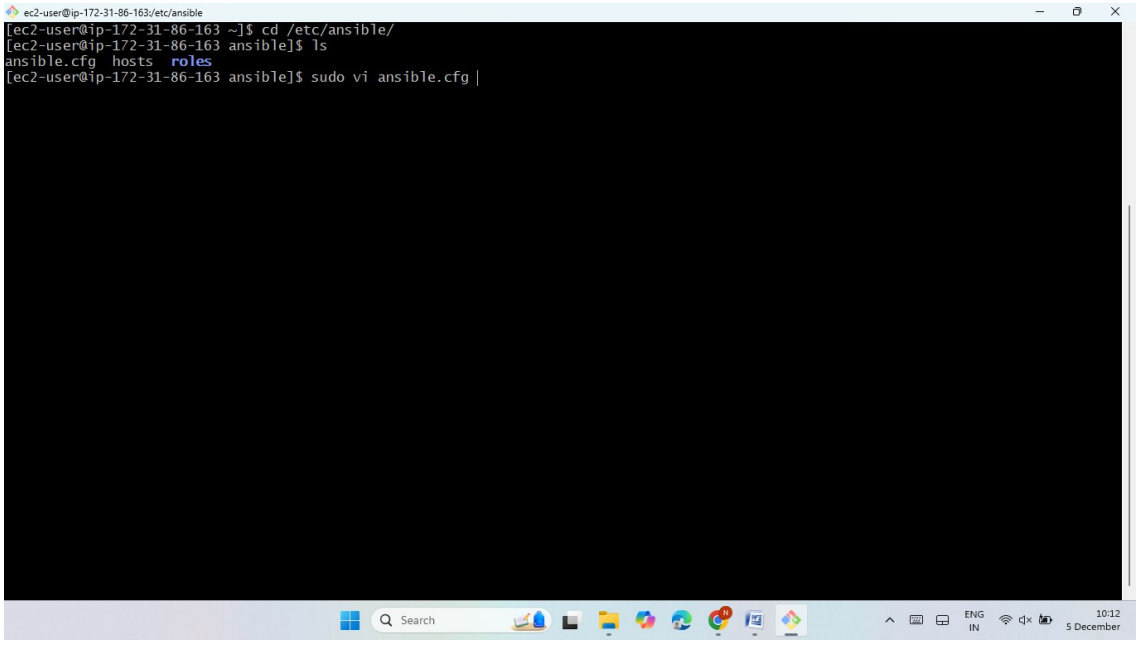
- Now open another terminal then copy the private key form local to instances.



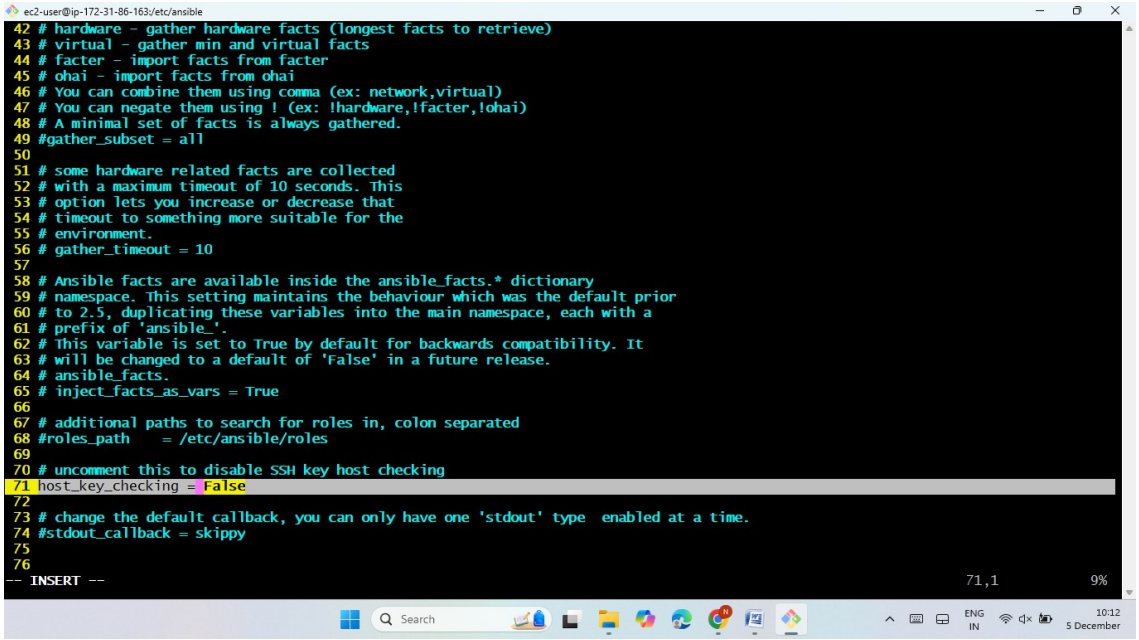
- Here the private key was copied then give permission to the key file.



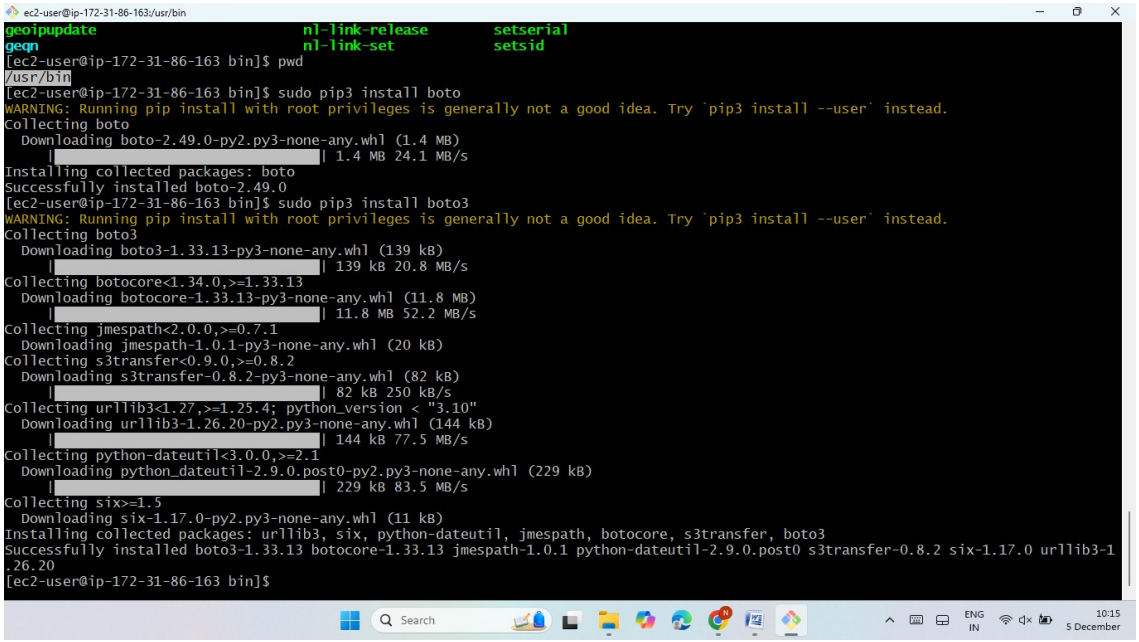
- Now go to ansible.cfg file (path= /ect/ansible)

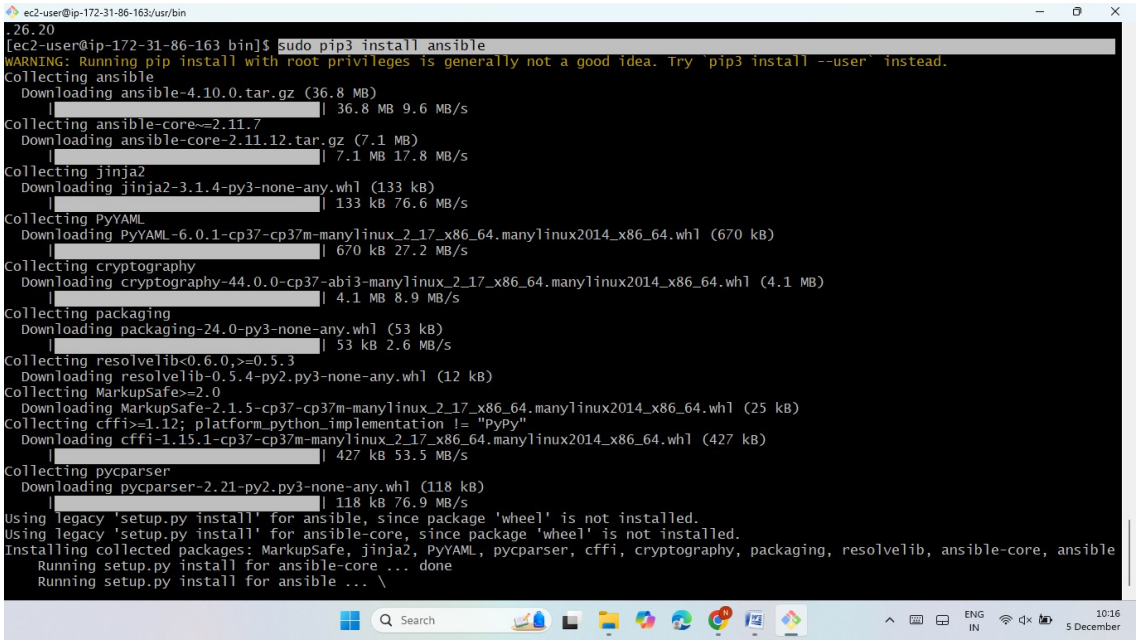


- Uncomment the 71 th line in the ansible configuration file.

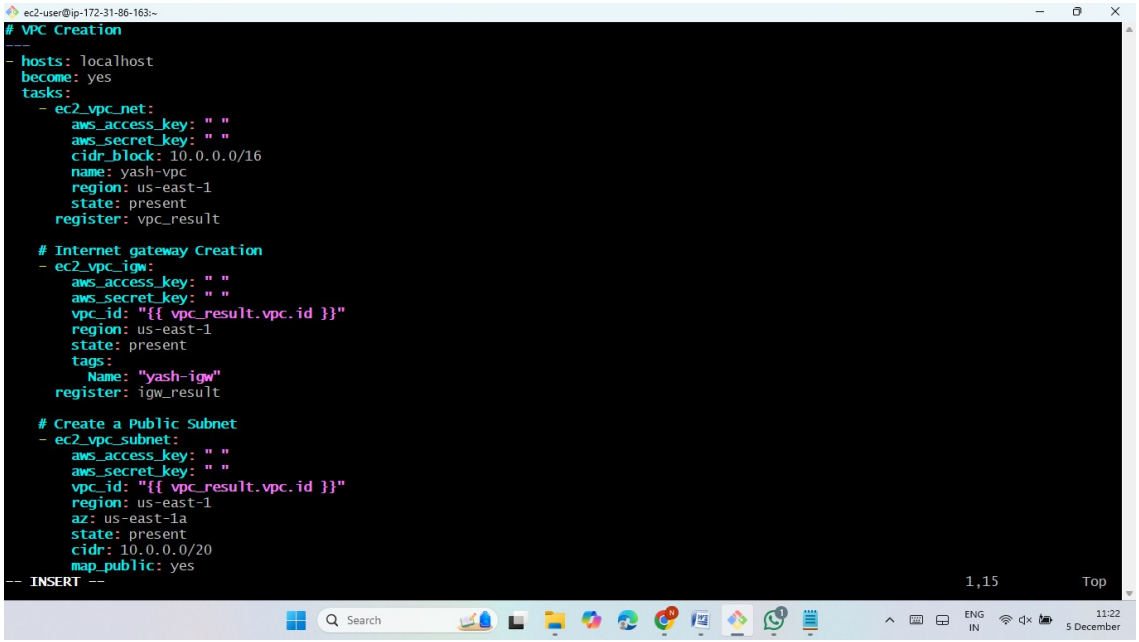


- Now go to /usr/bin/ path then install boto, boto3, ansible with the command. **Sudo pip3 install boto boto3 ansible**.

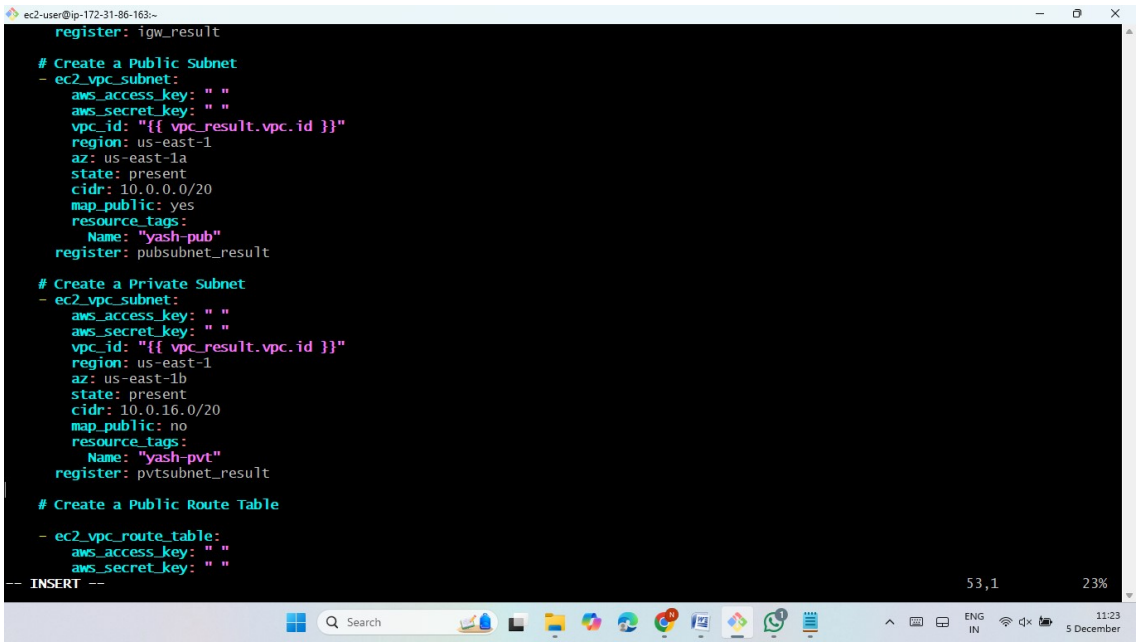




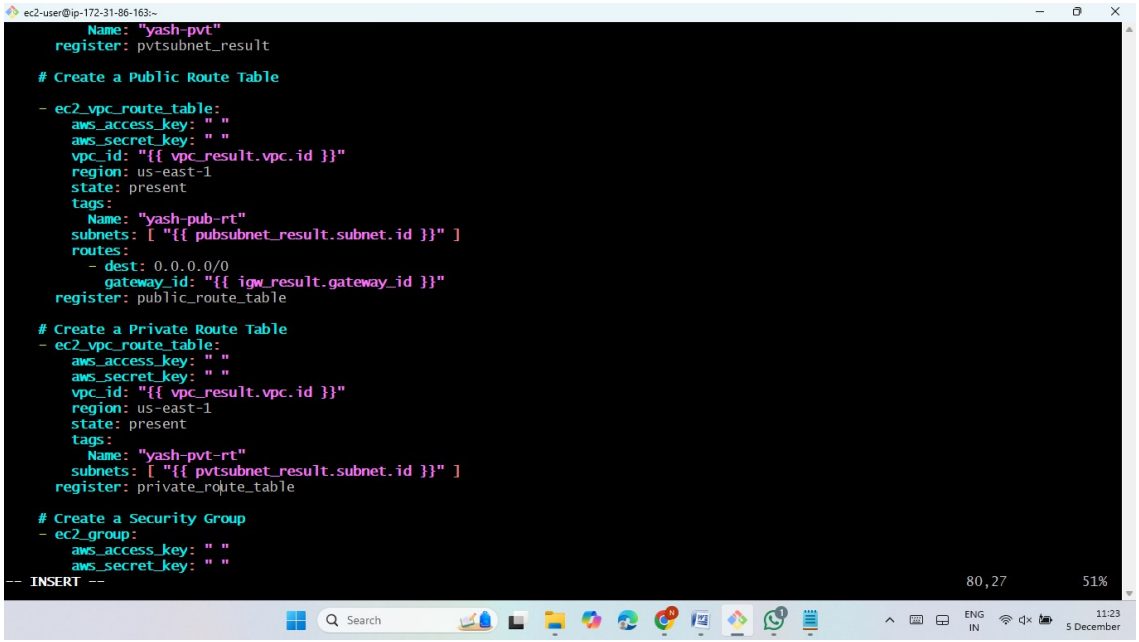
- Now create a file for host the static and dynamic application with the Ansible Script(yaml).
- First create a file for host 2 Instances, VPC, IGW, Private and Public Subnets, Private and Public Route Tables, Security Group.

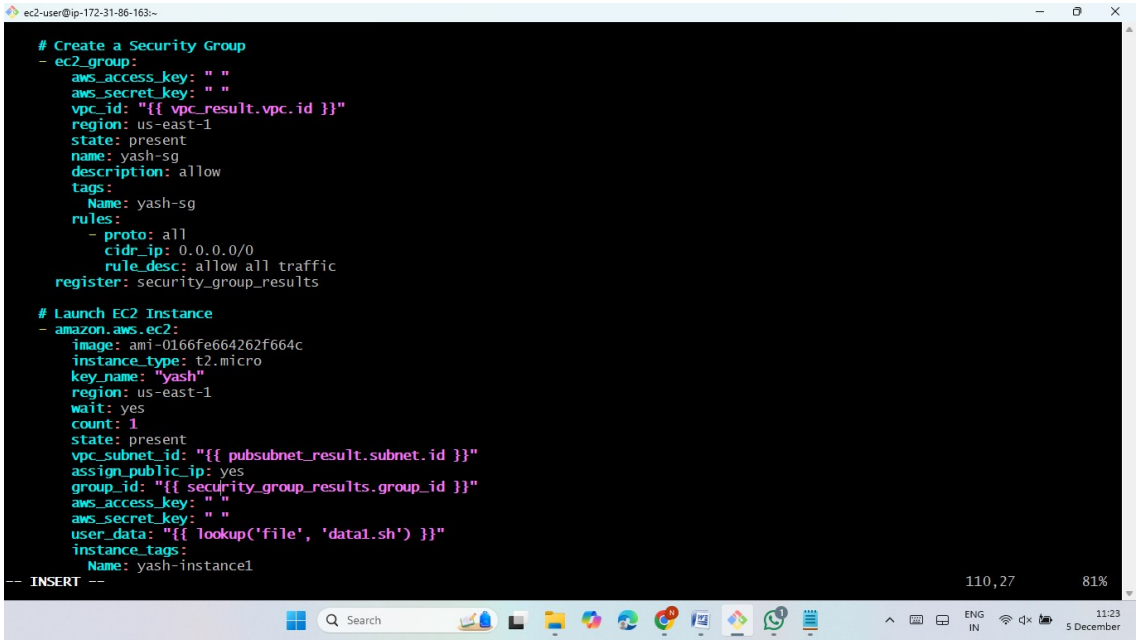


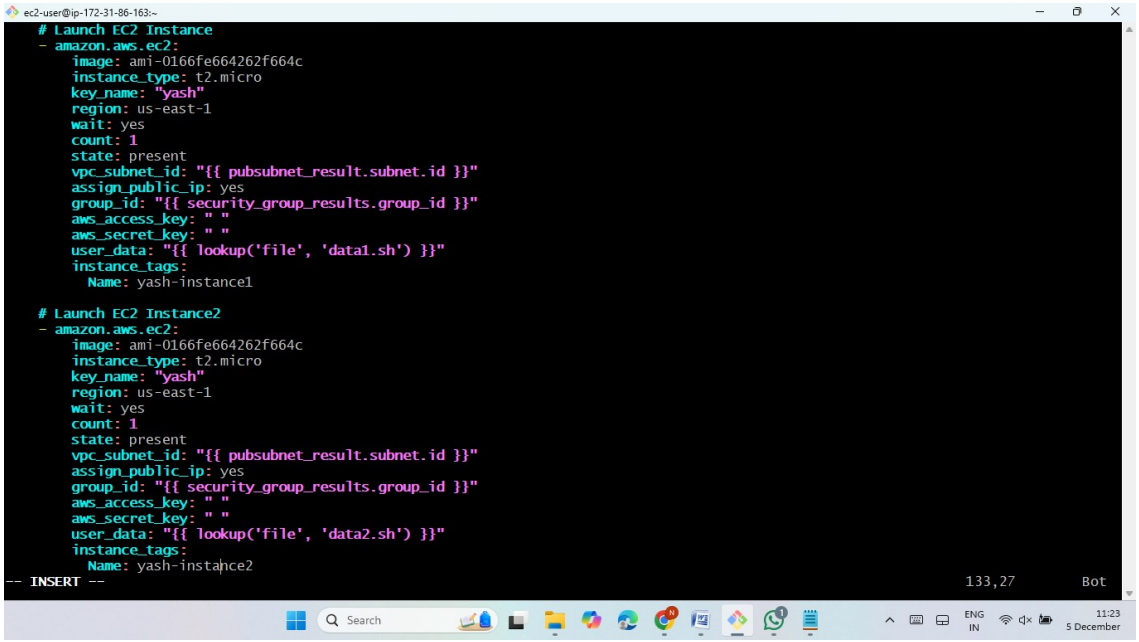
```
ec2-user@ip-172-31-86-163:~
# VPC Creation
---
- hosts: localhost
  become: yes
  tasks:
    - ec2_vpc_net:
        aws_access_key: " "
        aws_secret_key: " "
        cidr_block: 10.0.0.0/16
        name: yash-vpc
        region: us-east-1
        state: present
      register: vpc_result

    # Internet gateway Creation
    - ec2_vpc_igw:
        aws_access_key: " "
        aws_secret_key: " "
        vpc_id: "{{ vpc_result.vpc.id }}"
        region: us-east-1
        state: present
        tags:
          Name: "yash-igw"
      register: igw_result

    # Create a Public Subnet
    - ec2_vpc_subnet:
        aws_access_key: " "
        aws_secret_key: " "
        vpc_id: "{{ vpc_result.vpc.id }}"
        region: us-east-1
        az: us-east-1a
        state: present
        cidr: 10.0.0.0/20
        map_public: yes
-- INSERT --                                                    1,15          Top
```



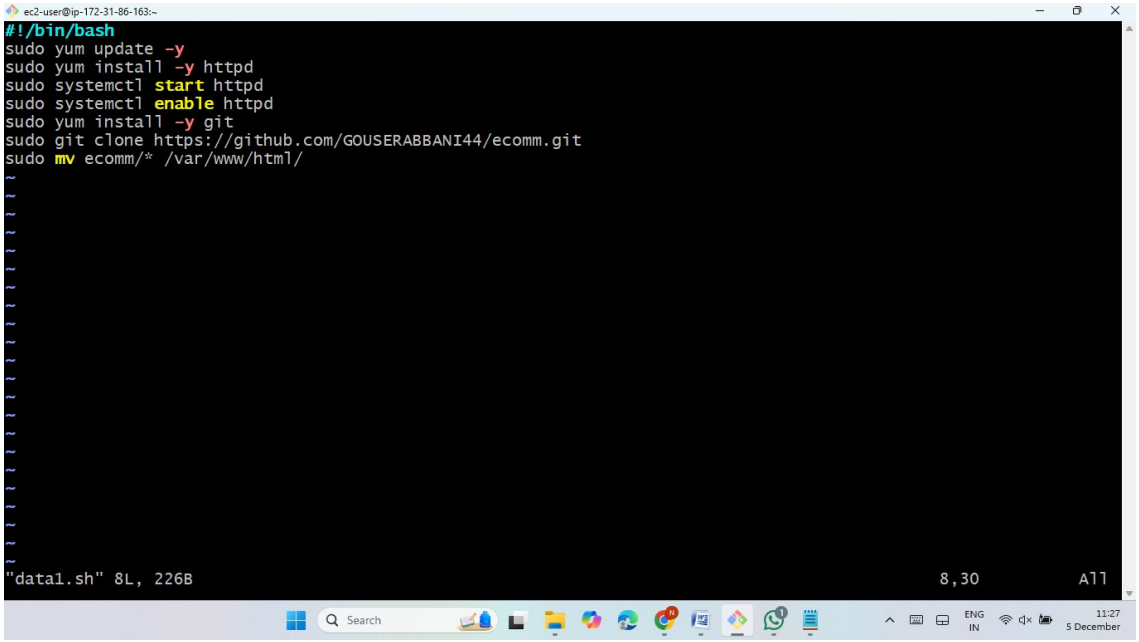
```
ec2-user@ip-172-31-86-163:~
      register: igw_result

    # Create a Public Subnet
    - ec2_vpc_subnet:
        aws_access_key: " "
        aws_secret_key: " "
        vpc_id: "{{ vpc_result.vpc.id }}"
        region: us-east-1
        az: us-east-1a
        state: present
        cidr: 10.0.0.0/20
        map_public: yes
        resource_tags:
          Name: "yash-pub"
      register: pubsubnet_result

    # Create a Private Subnet
    - ec2_vpc_subnet:
        aws_access_key: " "
        aws_secret_key: " "
        vpc_id: "{{ vpc_result.vpc.id }}"
        region: us-east-1
        az: us-east-1b
        state: present
        cidr: 10.0.16.0/20
        map_public: no
        resource_tags:
          Name: "yash-pvt"
      register: pvtsubnet_result

    # Create a Public Route Table

    - ec2_vpc_route_table:
        aws_access_key: " "
        aws_secret_key: " "
-- INSERT --                                                    53,1          23%
```


ec2-user@ip-172-31-86-163:~
Name: "yash-pvt"
register: pvtsubnet_result
Create a Public Route Table
- ec2_vpc_route_table:
aws_access_key: " "
aws_secret_key: " "
vpc_id: "{{ vpc_result.vpc.id }}"
region: us-east-1
state: present
tags:
Name: "yash-pub-rt"
subnets: ["{{ pubsubnet_result.subnet.id }}"]
routes:
- dest: 0.0.0.0/0
gateway_id: "{{ igw_result.gateway_id }}"
register: public_route_table
Create a Private Route Table
- ec2_vpc_route_table:
aws_access_key: " "
aws_secret_key: " "
vpc_id: "{{ vpc_result.vpc.id }}"
region: us-east-1
state: present
tags:
Name: "yash-pvt-rt"
subnets: ["{{ pvtsubnet_result.subnet.id }}"]
register: private_route_table
Create a Security Group
- ec2_group:
aws_access_key: " "
aws_secret_key: " "
-- INSERT --
80,27
51%
Search
ENG
IN
11:23
5 December

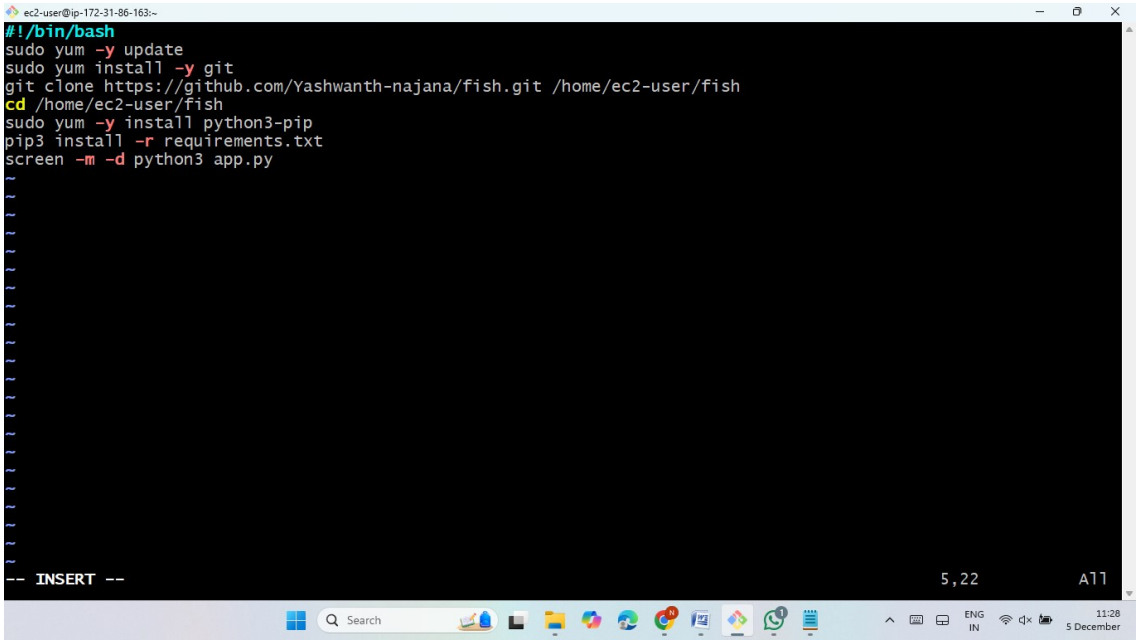

ec2-user@ip-172-31-86-163:~
Create a Security Group
- ec2_group:
aws_access_key: " "
aws_secret_key: " "
vpc_id: "{{ vpc_result.vpc.id }}"
region: us-east-1
state: present
name: yash-sg
description: allow
tags:
Name: yash-sg
rules:
- proto: all
cidr_ip: 0.0.0.0/0
rule_desc: allow all traffic
register: security_group_results
Launch EC2 Instance
- amazon.aws.ec2:
image: ami-0166fe664262f664c
instance_type: t2.micro
key_name: "yash"
region: us-east-1
wait: yes
count: 1
state: present
vpc_subnet_id: "{{ pubsubnet_result.subnet.id }}"
assign_public_ip: yes
group_id: "{{ security_group_results.group_id }}"
aws_access_key: " "
aws_secret_key: " "
user_data: "{{ lookup('file', 'data1.sh') }}"
instance_tags:
Name: yash-instance1
-- INSERT --
110,27
81%
Search
ENG
IN
11:23
5 December



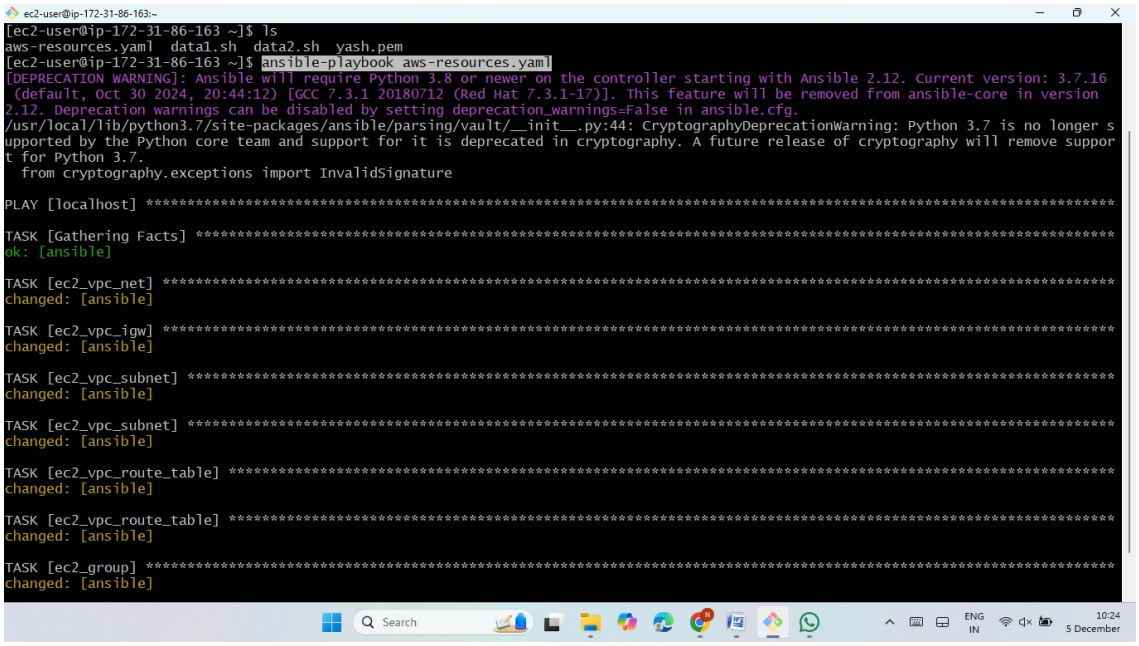
- Now write a bash script for host static application in instance1.

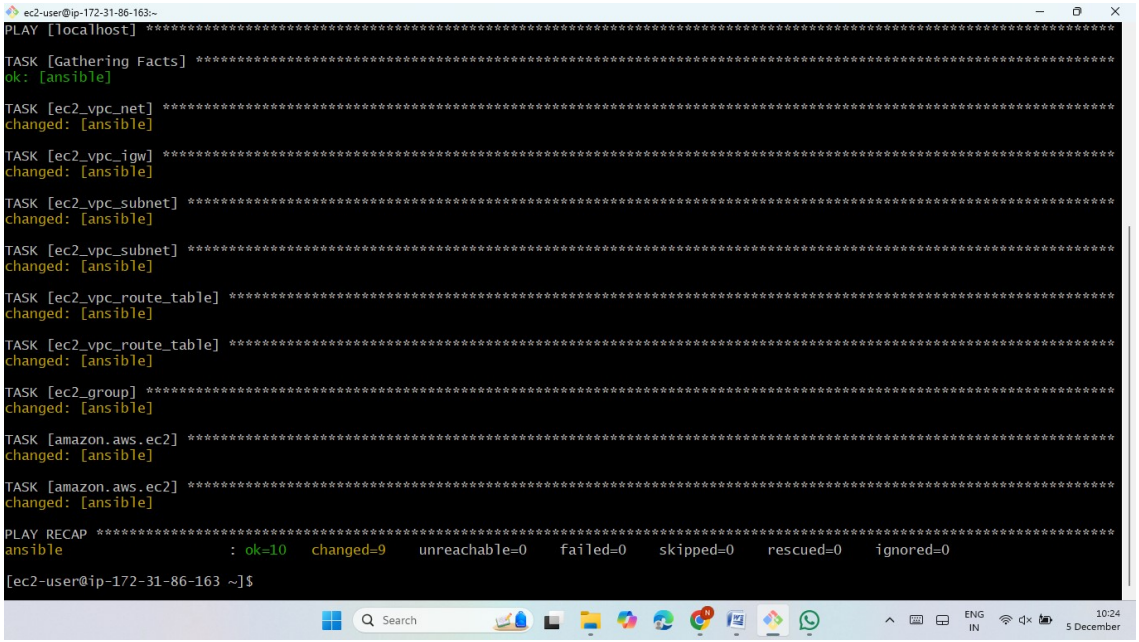


- Now write a bash script for host static application in instance2.

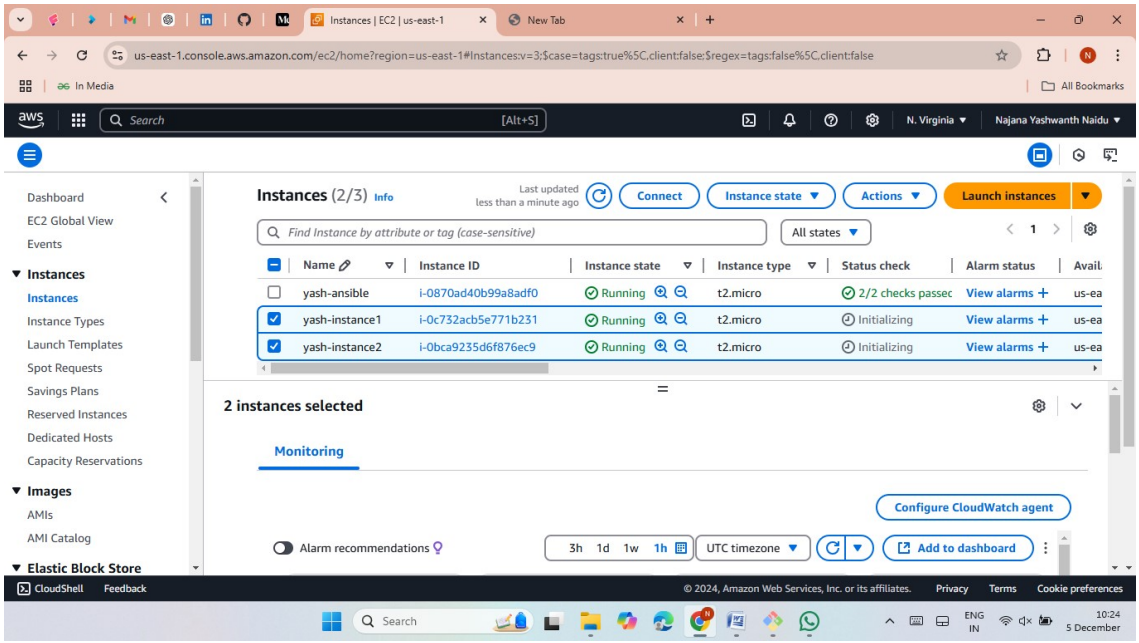


- Now run the AWS resources with the command.
  Ansible-playbook <name of file>
- Here the resources was created successfully.

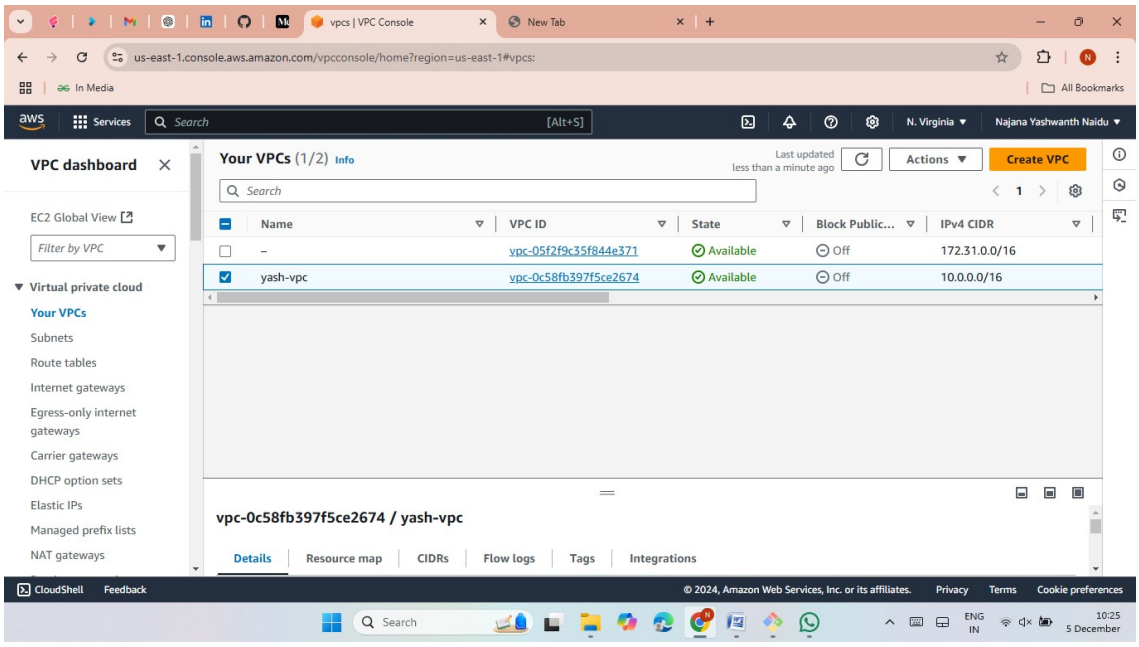




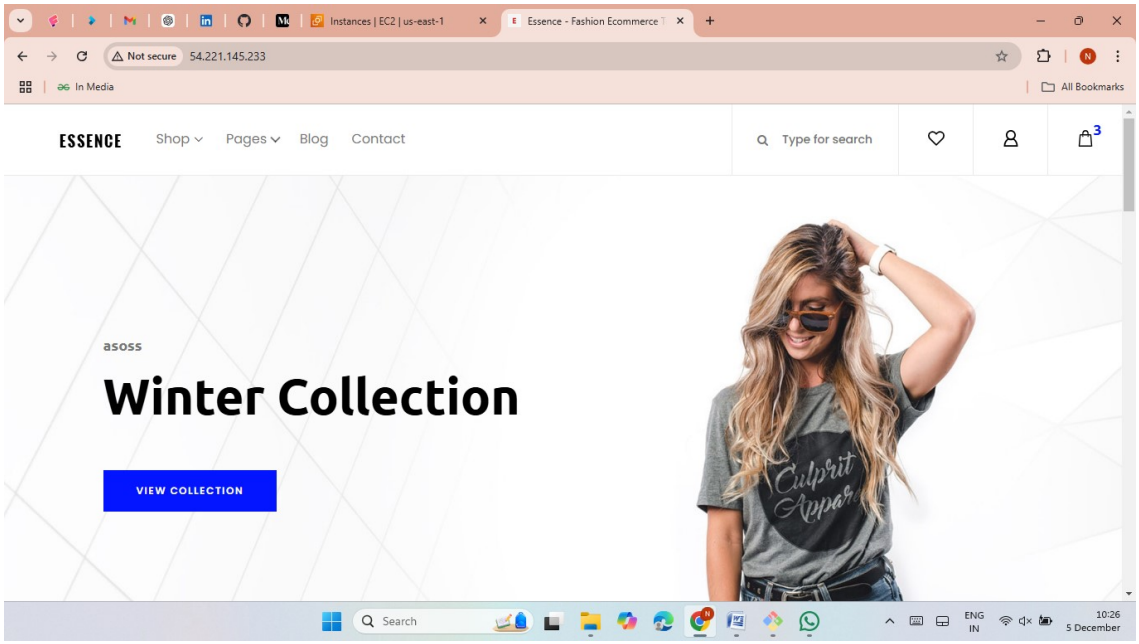
- Here the 2 instances was launched.



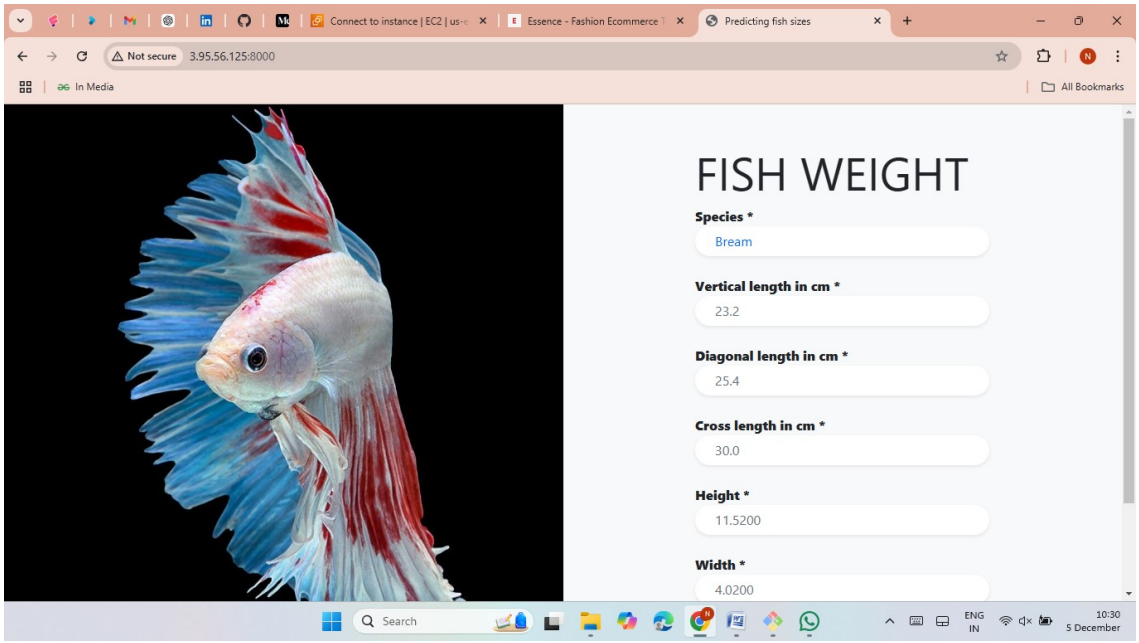
- Here the VPC, IGW, Subnets, Route Tables and Security group was created.



- Now copy the Instance 1 Public IP then Past it on Google along with port 80.
- The static ecomm application was hosted successfully.

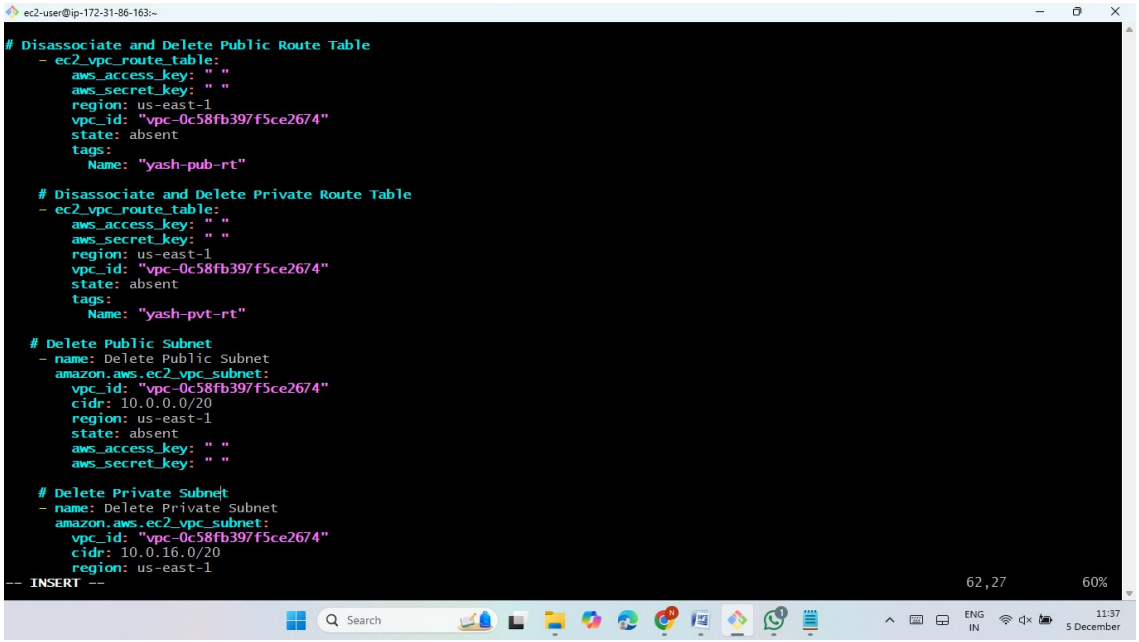


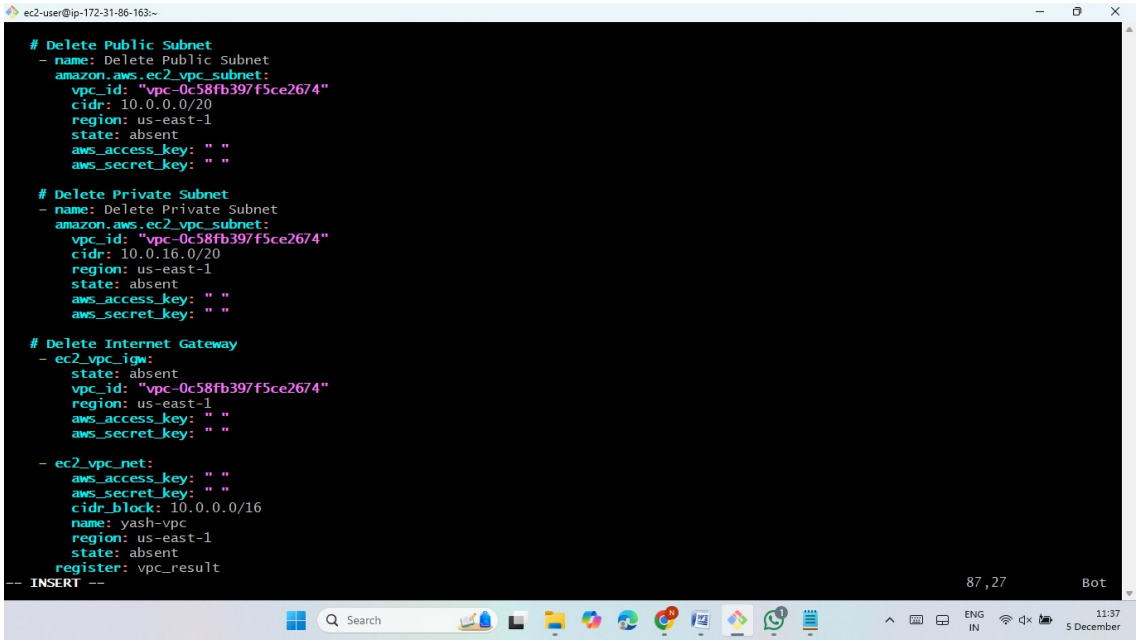
- Now copy the Instance2 Public IP then Past it on Google along with port 8000.
- The dynamic python application was hosted successfully.



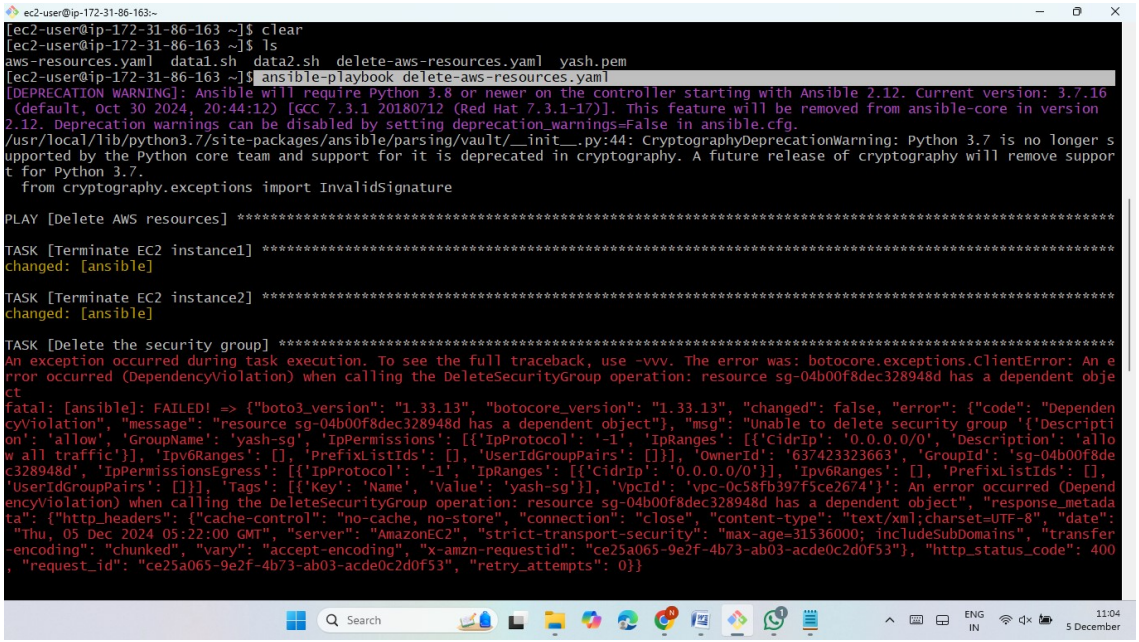
- Now create file for delete the created resources.

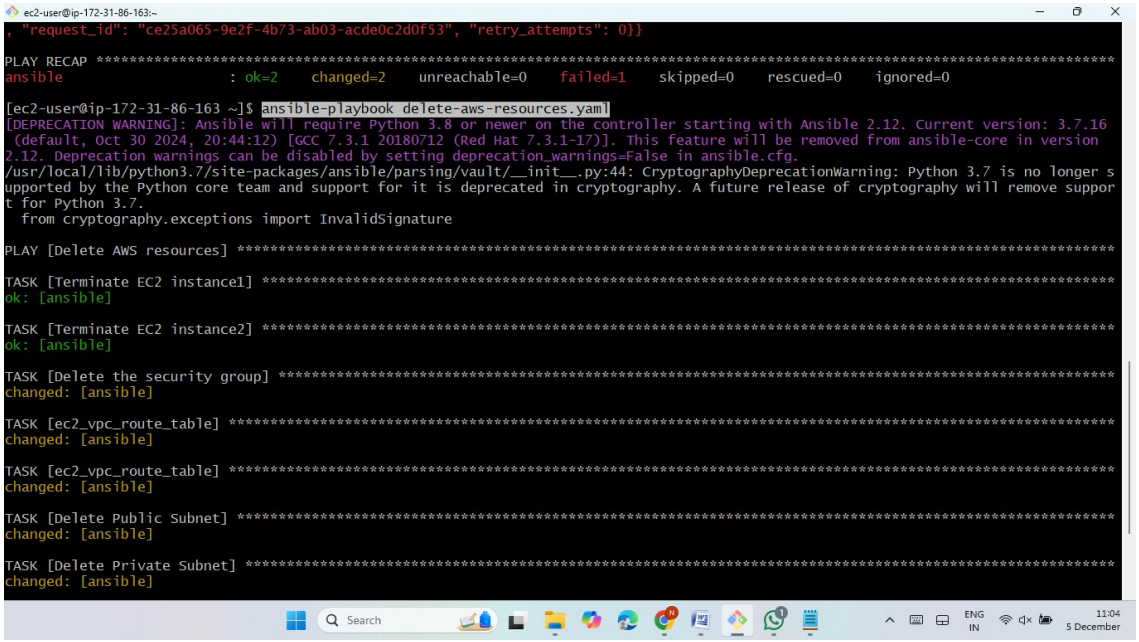
```
- name: Delete AWS resources
  hosts: localhost  # If running locally, otherwise specify your inventory
  gather_facts: no
  tasks:
    - name: Terminate EC2 instance1
      amazon.aws.ec2:
        instance_ids:
          - i-0c732acb5e771b231
        region: us-east-1
        state: absent
        aws_access_key: " "
        aws_secret_key: " "

    - name: Terminate EC2 instance2
      amazon.aws.ec2:
        instance_ids:
          - i-0bca9235d6f876ec9
        region: us-east-1
        state: absent
        aws_access_key: " "
        aws_secret_key: " "

    - name: Delete the security group
      amazon.aws.ec2_group:
        aws_access_key: " "
        aws_secret_key: " "
        region: us-east-1
        state: absent
        group_id: "sg-04b00f8dec328948d"
      register: sg_delete_result

# Disassociate and Delete Public Route Table
    - ec2_vpc_route_table:
        aws_access_key: " "
        aws_secret_key: " "
        region: us-east-1
        vpc_id: "vpc-0c58fb397f5ce2674"
-- INSERT --                                                    1,27          Top
```

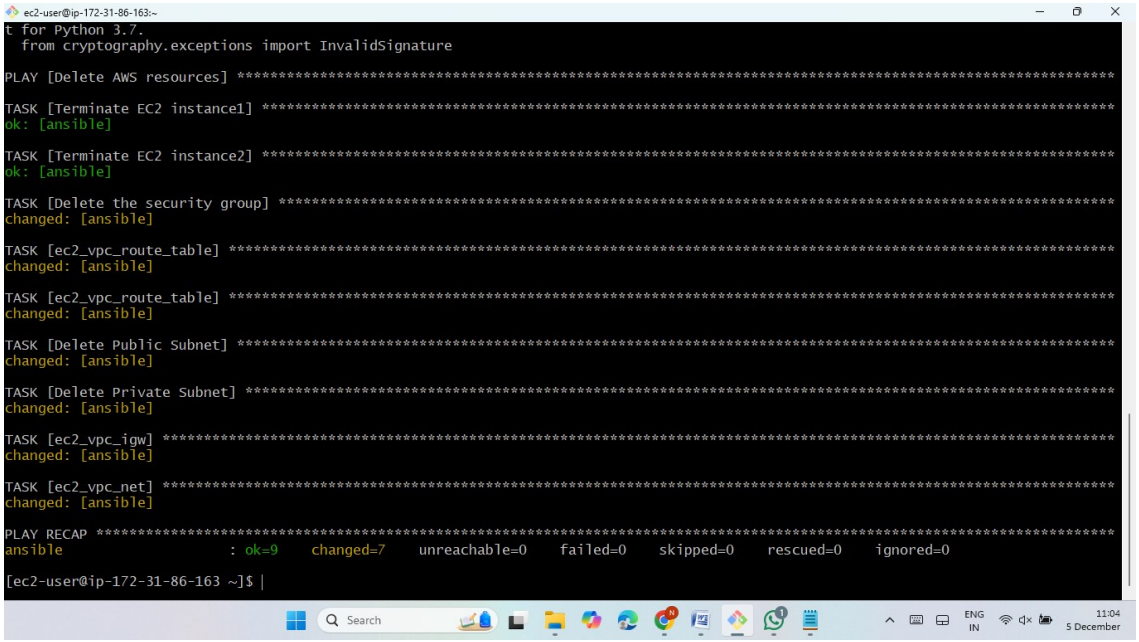

ec2-user@ip-172-31-86-163:~
Disassociate and Delete Public Route Table
- ec2_vpc_route_table:
aws_access_key: " "
aws_secret_key: " "
region: us-east-1
vpc_id: "vpc-0c58fb397f5ce2674"
state: absent
tags:
Name: "yash-pub-rt"
Disassociate and Delete Private Route Table
- ec2_vpc_route_table:
aws_access_key: " "
aws_secret_key: " "
region: us-east-1
vpc_id: "vpc-0c58fb397f5ce2674"
state: absent
tags:
Name: "yash-pvt-rt"
Delete Public Subnet
- name: Delete Public Subnet
amazon.aws.ec2_vpc_subnet:
vpc_id: "vpc-0c58fb397f5ce2674"
cidr: 10.0.0.0/20
region: us-east-1
state: absent
aws_access_key: " "
aws_secret_key: " "
Delete Private Subnet
- name: Delete Private Subnet
amazon.aws.ec2_vpc_subnet:
vpc_id: "vpc-0c58fb397f5ce2674"
cidr: 10.0.16.0/20
region: us-east-1
-- INSERT --
62,27
60%
Search
ENG
IN
11:37
5 December


ec2-user@ip-172-31-86-163:~
Delete Public Subnet
- name: Delete Public Subnet
amazon.aws.ec2_vpc_subnet:
vpc_id: "vpc-0c58fb397f5ce2674"
cidr: 10.0.0.0/20
region: us-east-1
state: absent
aws_access_key: " "
aws_secret_key: " "
Delete Private Subnet
- name: Delete Private Subnet
amazon.aws.ec2_vpc_subnet:
vpc_id: "vpc-0c58fb397f5ce2674"
cidr: 10.0.16.0/20
region: us-east-1
state: absent
aws_access_key: " "
aws_secret_key: " "
Delete Internet Gateway
- ec2_vpc_igw:
state: absent
vpc_id: "vpc-0c58fb397f5ce2674"
region: us-east-1
aws_access_key: " "
aws_secret_key: " "
- ec2_vpc_net:
aws_access_key: " "
aws_secret_key: " "
cidr_block: 10.0.0.0/16
name: yash-vpc
region: us-east-1
state: absent
register: vpc_result
-- INSERT --
87,27
Bot
Search
ENG
IN
11:37
5 December

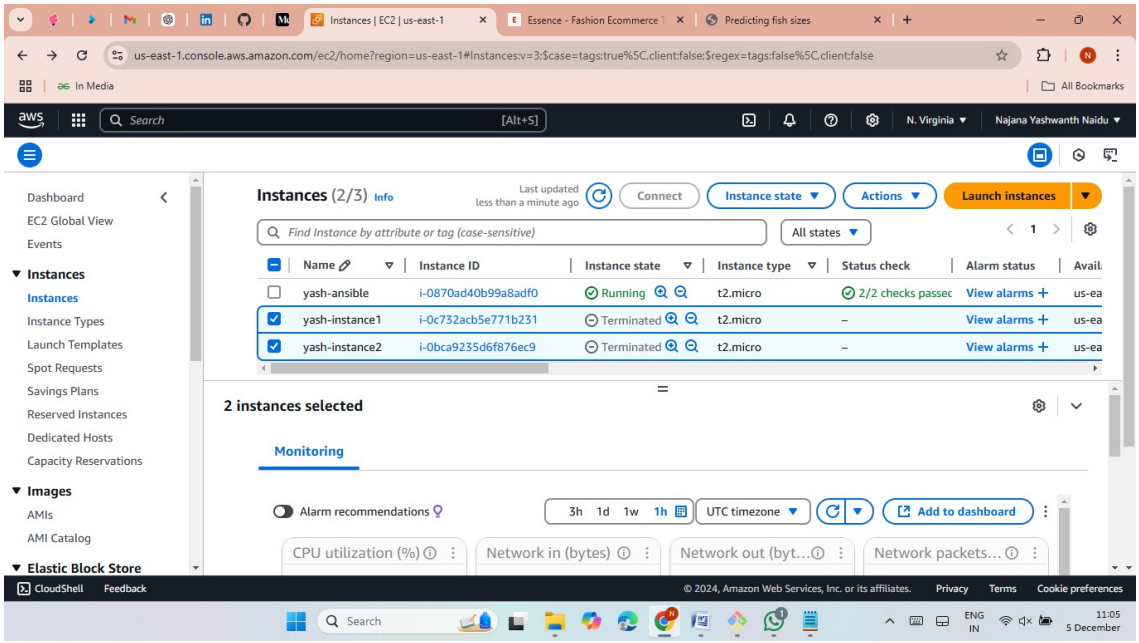
- Now run the file with the command **ansible-playbook <name of the file>.**
- Here the created resources was deleted successfully.



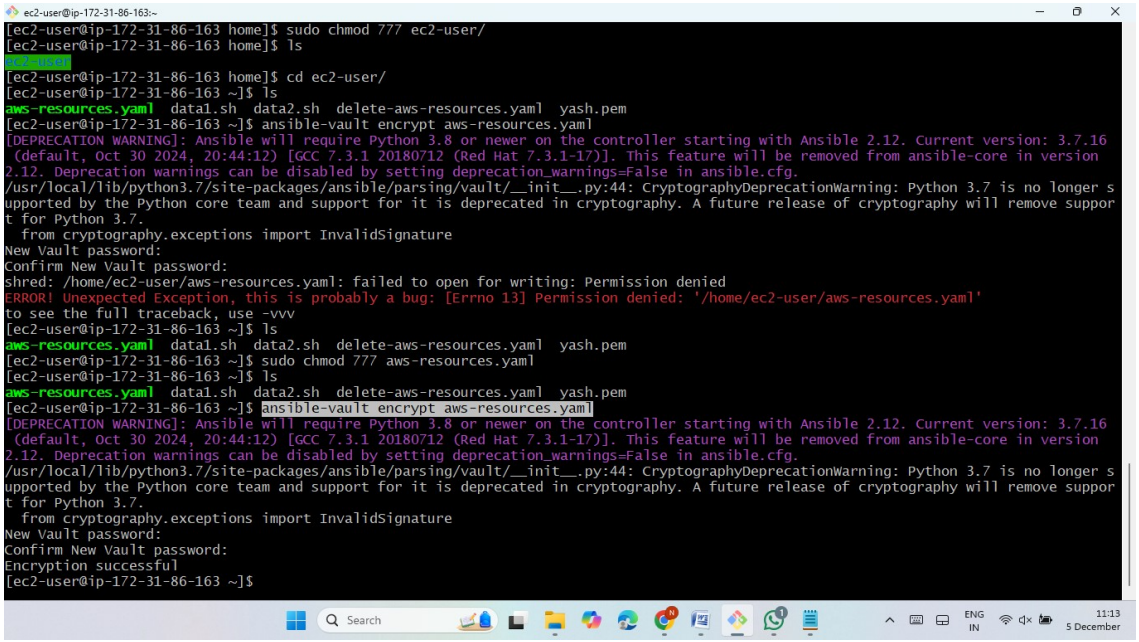




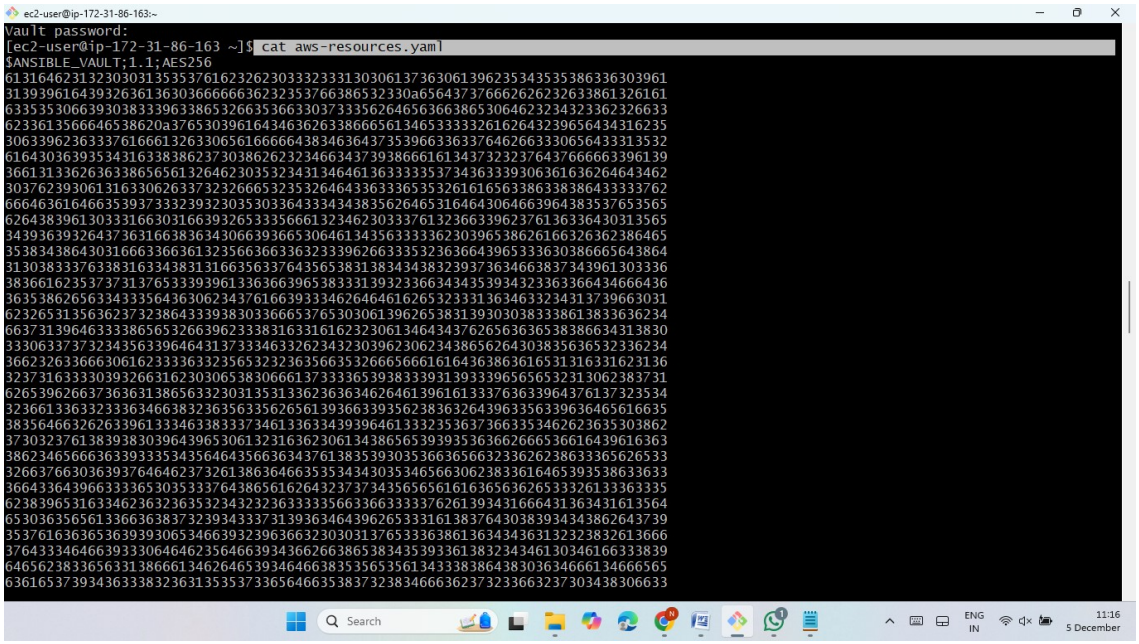
➢ Here the instances was deleted.



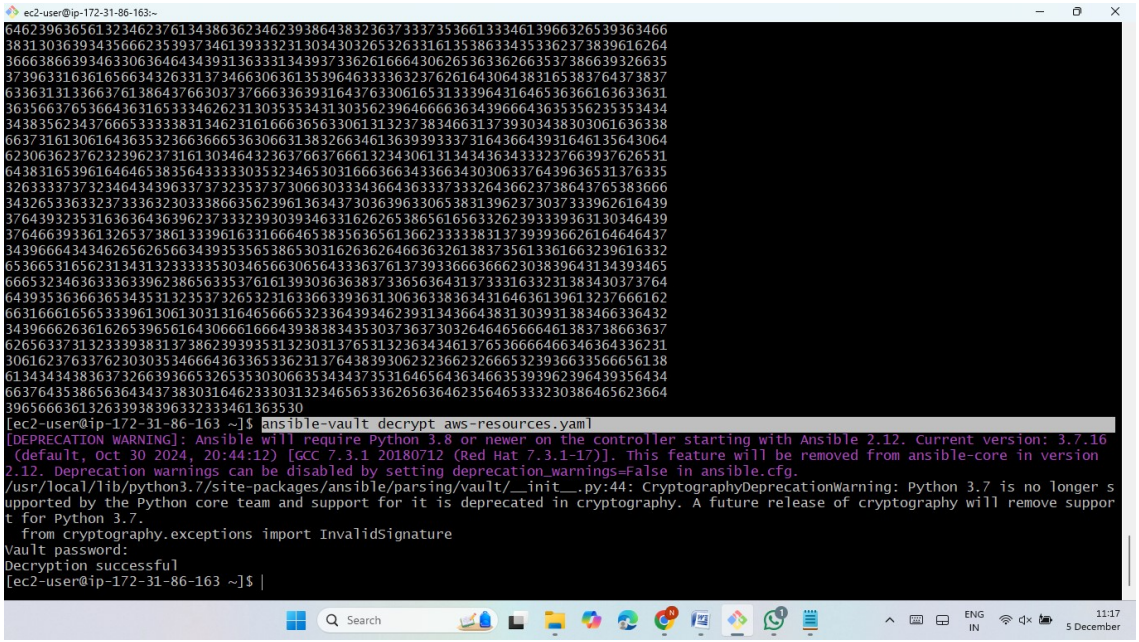
- Now Give 777 permissions to /home/ec2-user/filename.
- Now encrypt the created file with the command.
  ansible-vault encrypt aws-resources.yml



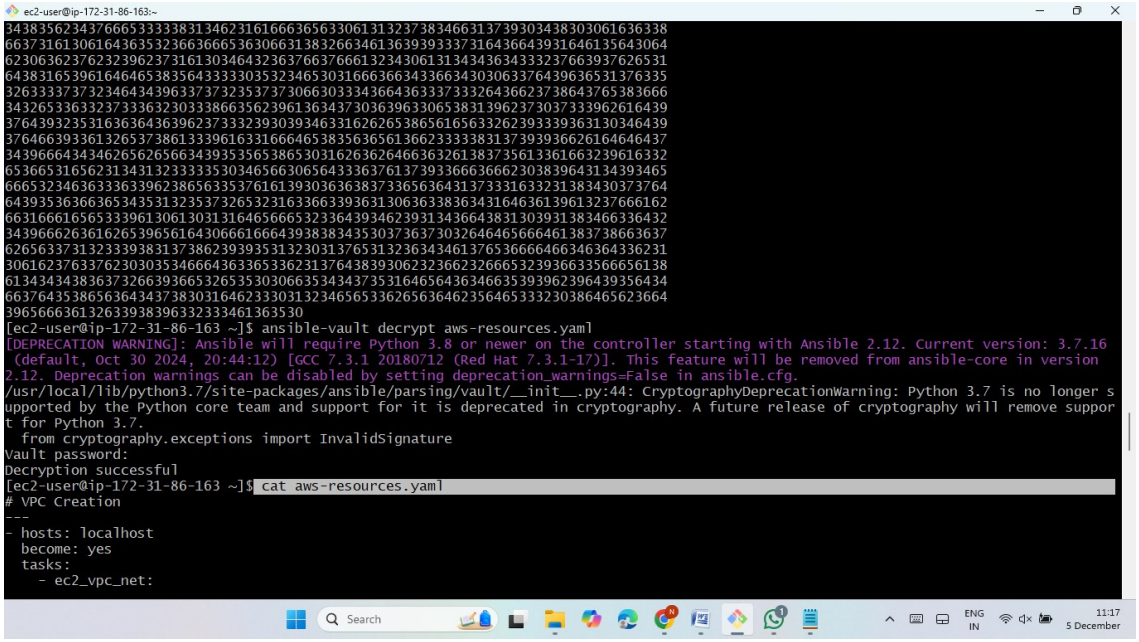
- Here the file was encrypted successfully.



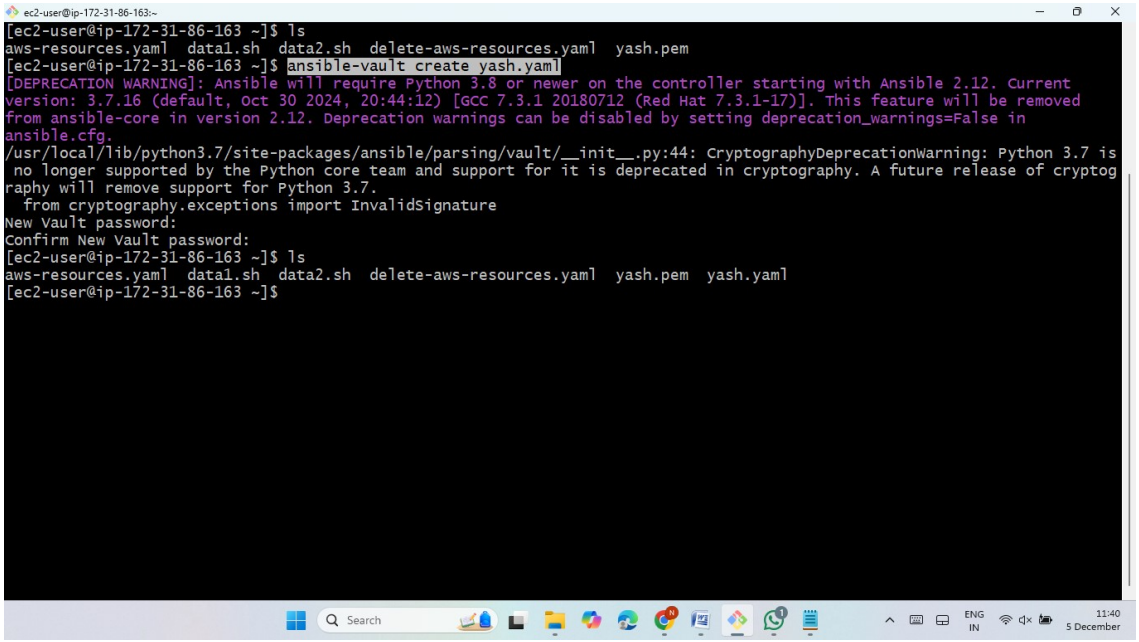
- Now decrypt the encrypted file with the command.
- **ansible-vault decrypt aws-resources.yaml**



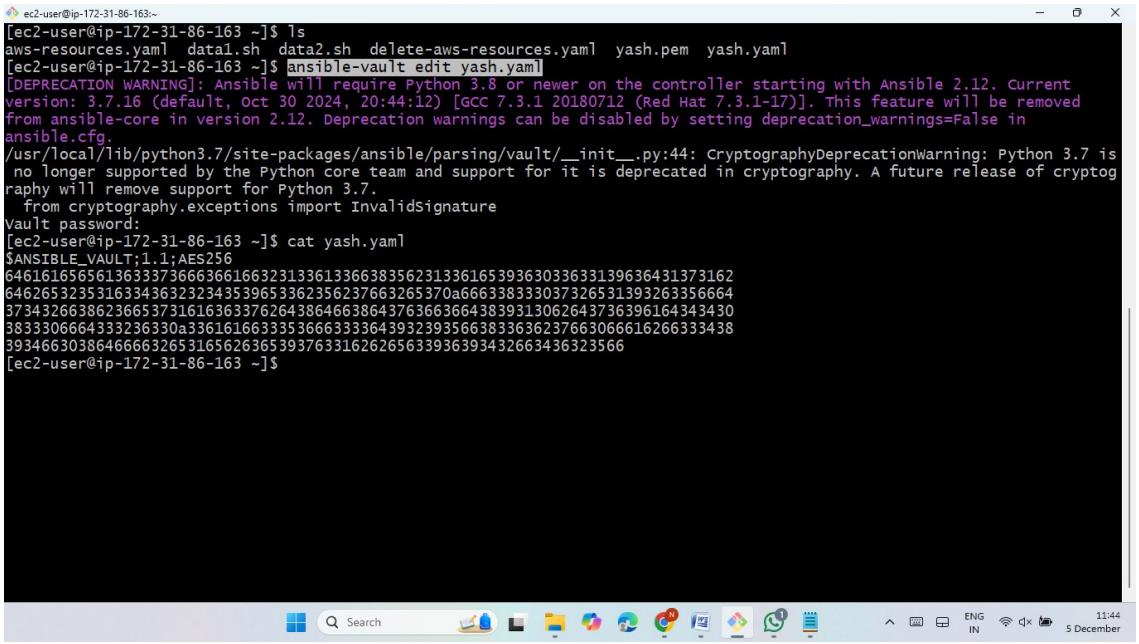
- Here the file was decrypted.



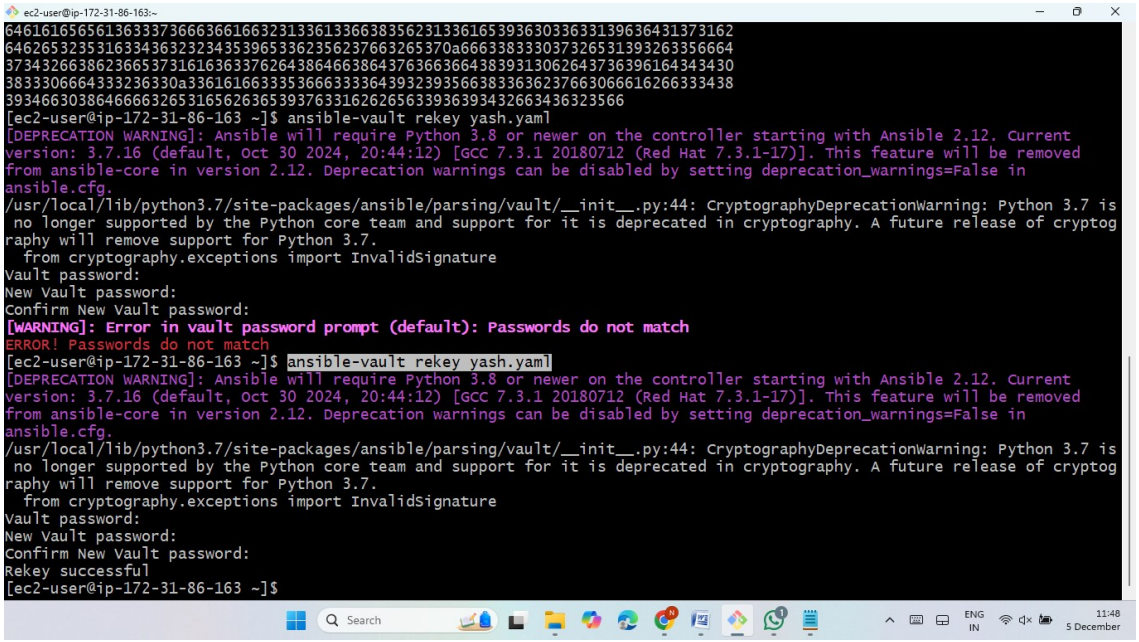
- Now create an encrypt files with the command.
  ansible-vault create yash.yml



- Now Edit the encrypted file with the command.
  ansible-vault edit yash.yml

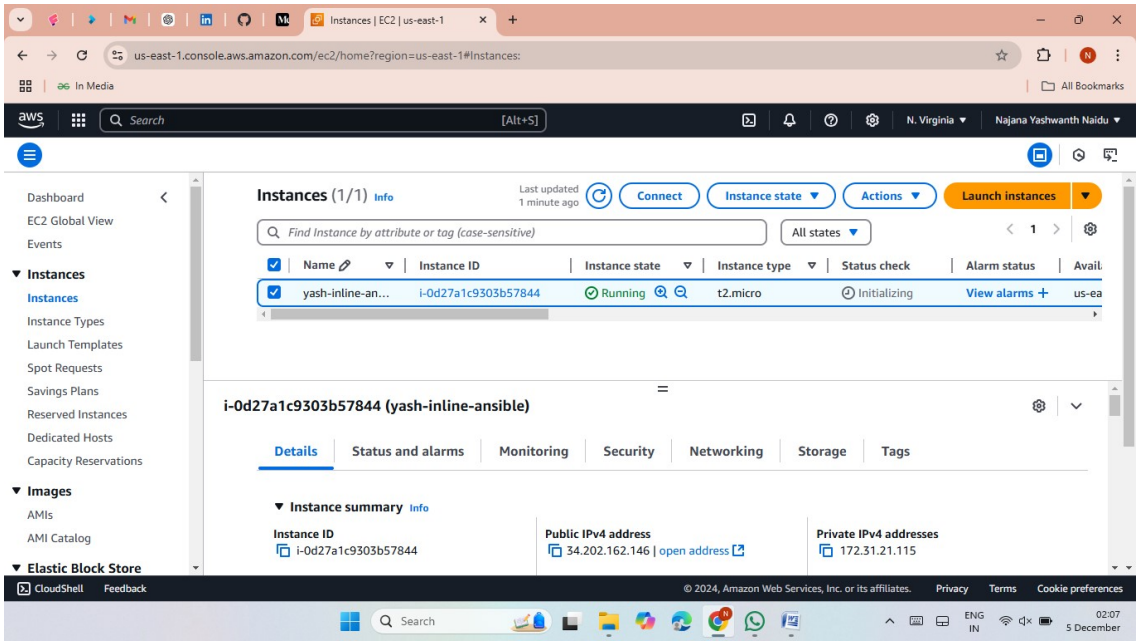


- Now reset the file password (reset the encrypted file password).
  ansible-vault rekey yash.yaml

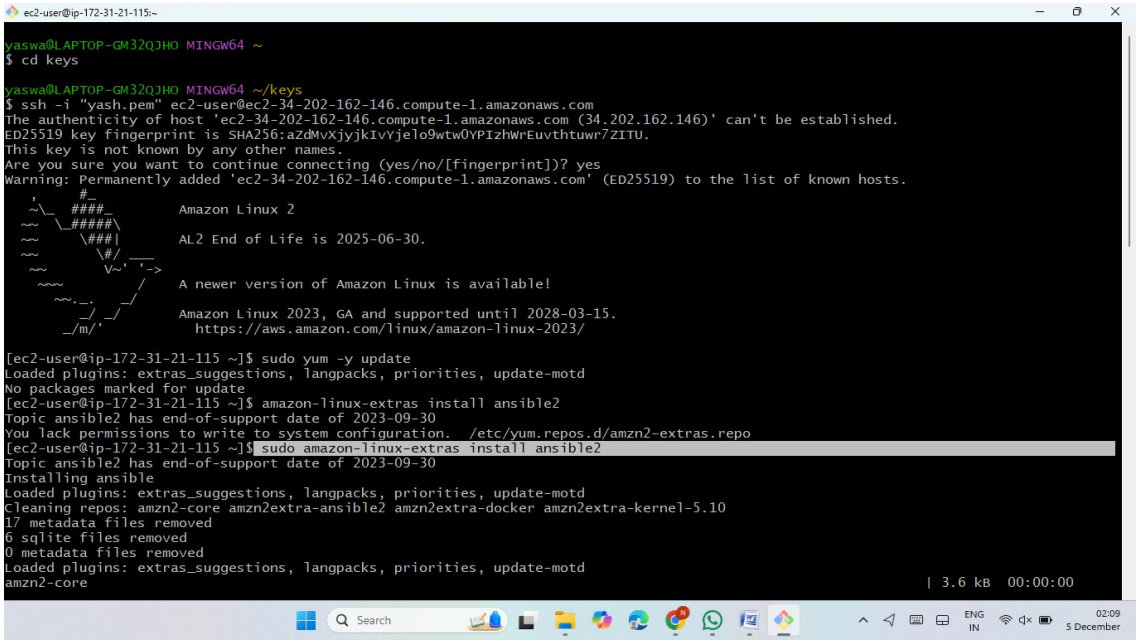


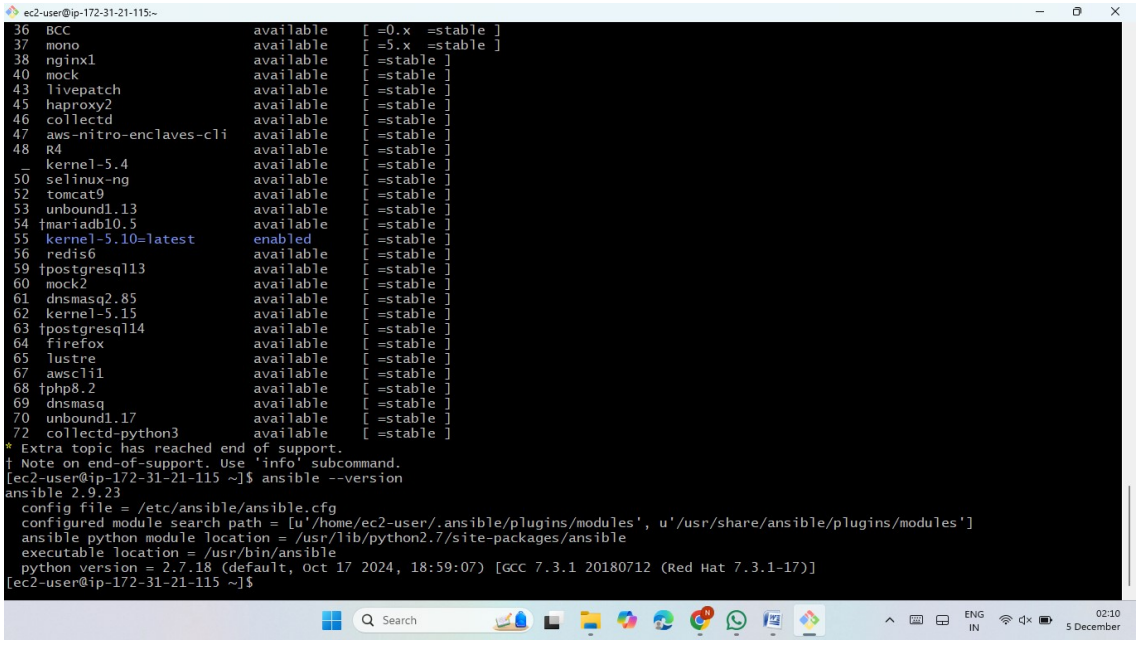
METHOD-2 (INLINE)

- Launch an Instances along with required ports.



- Now connect to GitBash then install ansible2 with the command.
  Sudo amazon-linux-extras install ansible2

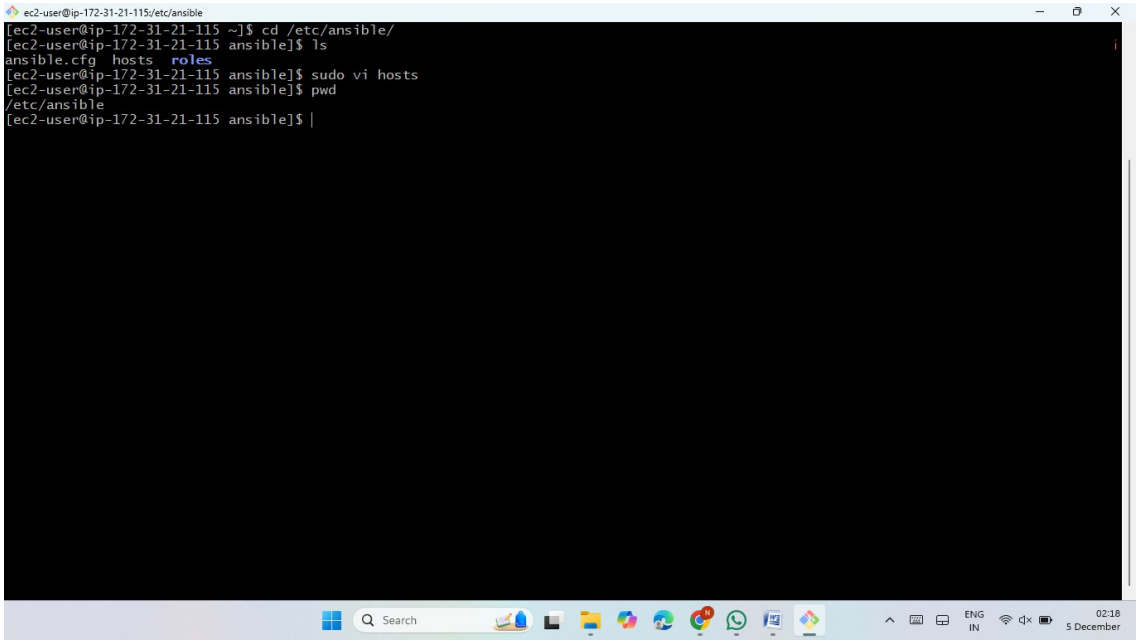


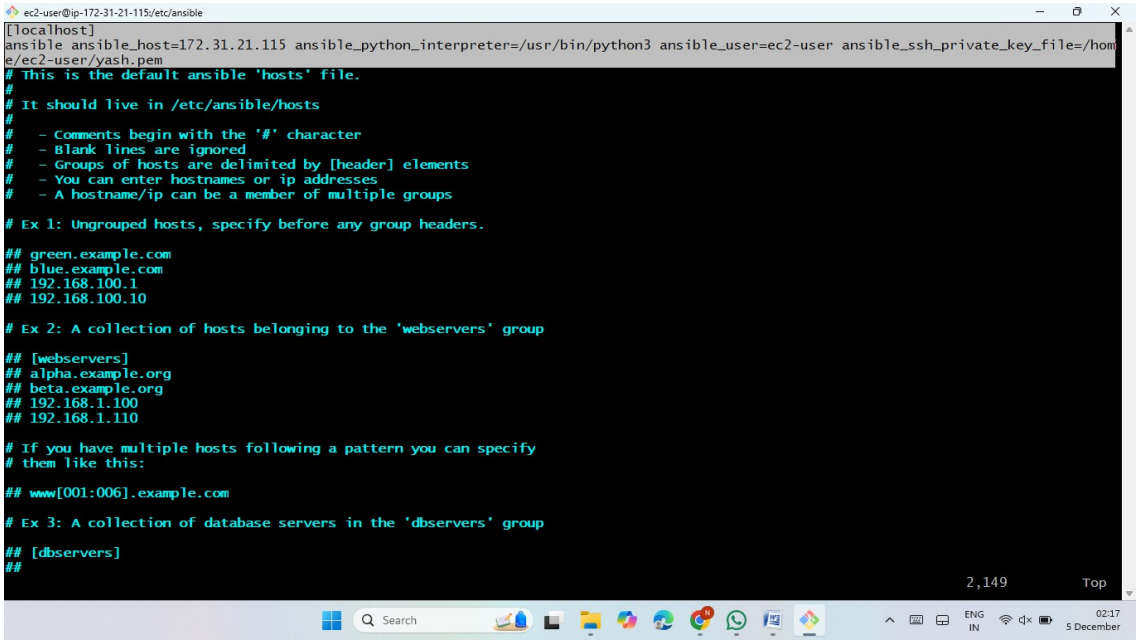


- Now Edit host file (path= /etc/ansible)

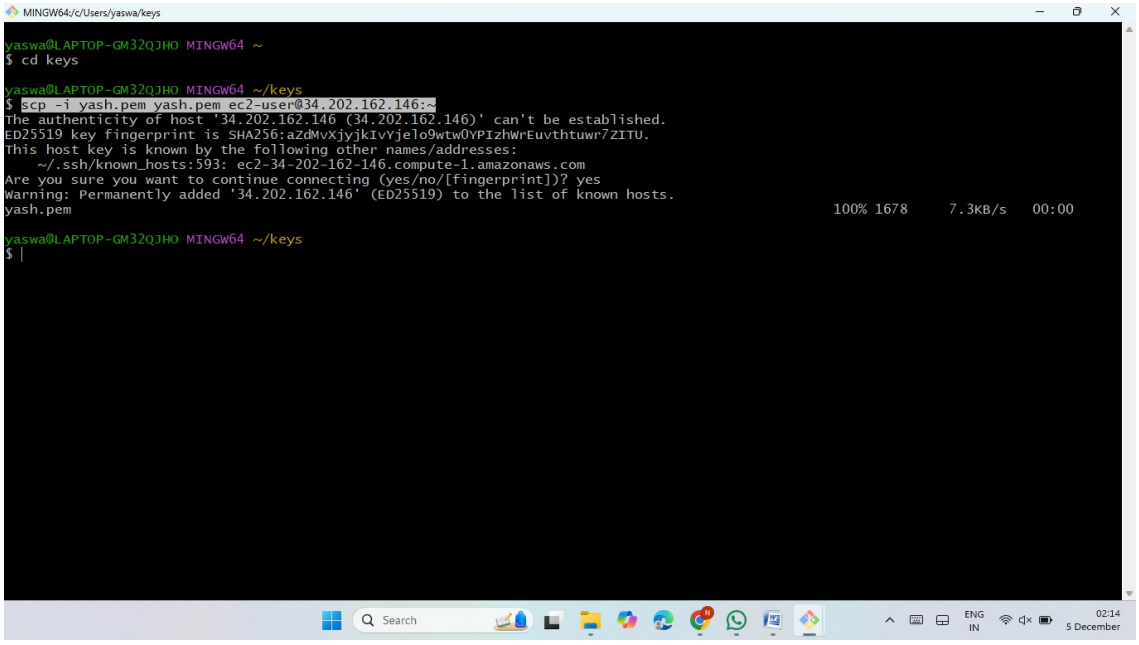
  [localhost]

  ansible ansible_host=10.0.49.164 ansible_python_interpreter=/usr/bin/python3 ansible_user=ec2-user ansible_ssh_private_key_file=/home/ec2-user/yash.pem

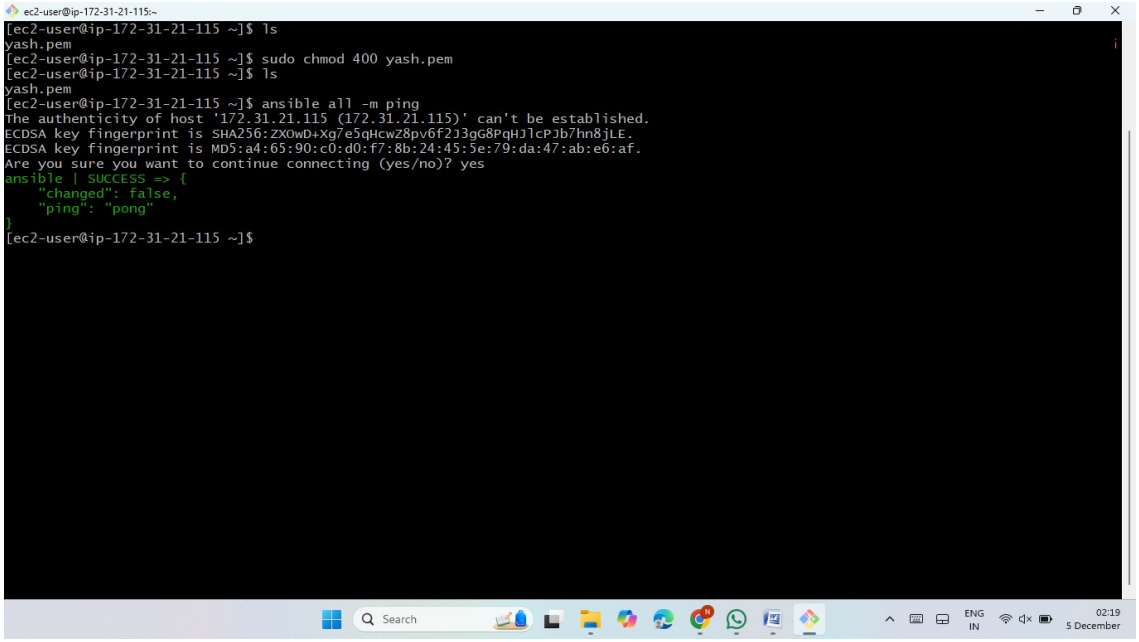




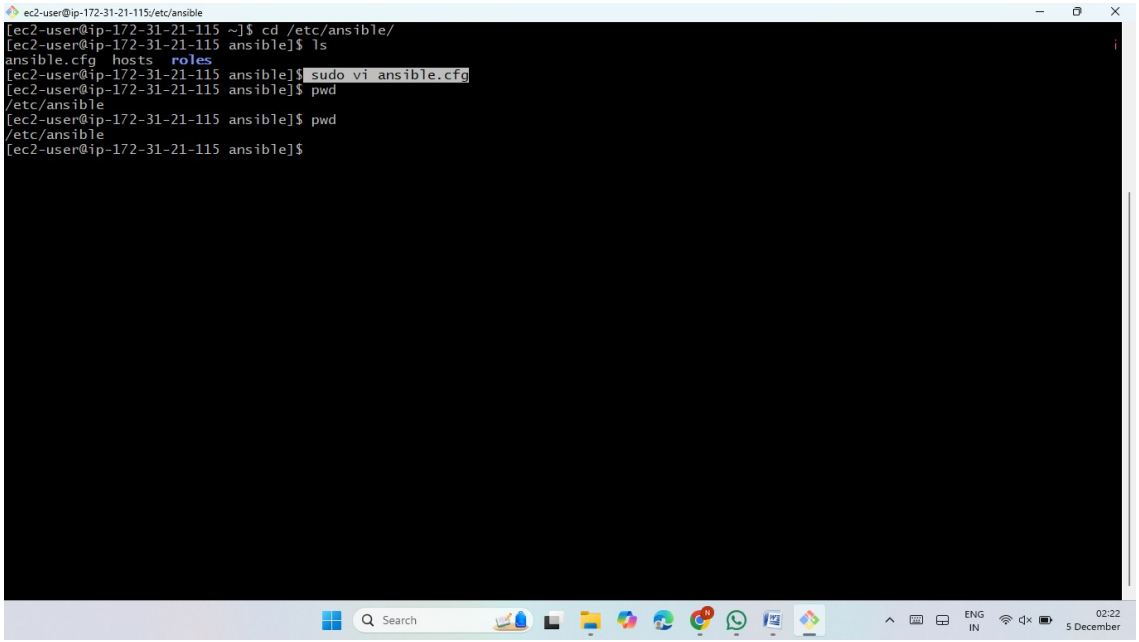
- Now open another terminal then copy the private key form local to instances.

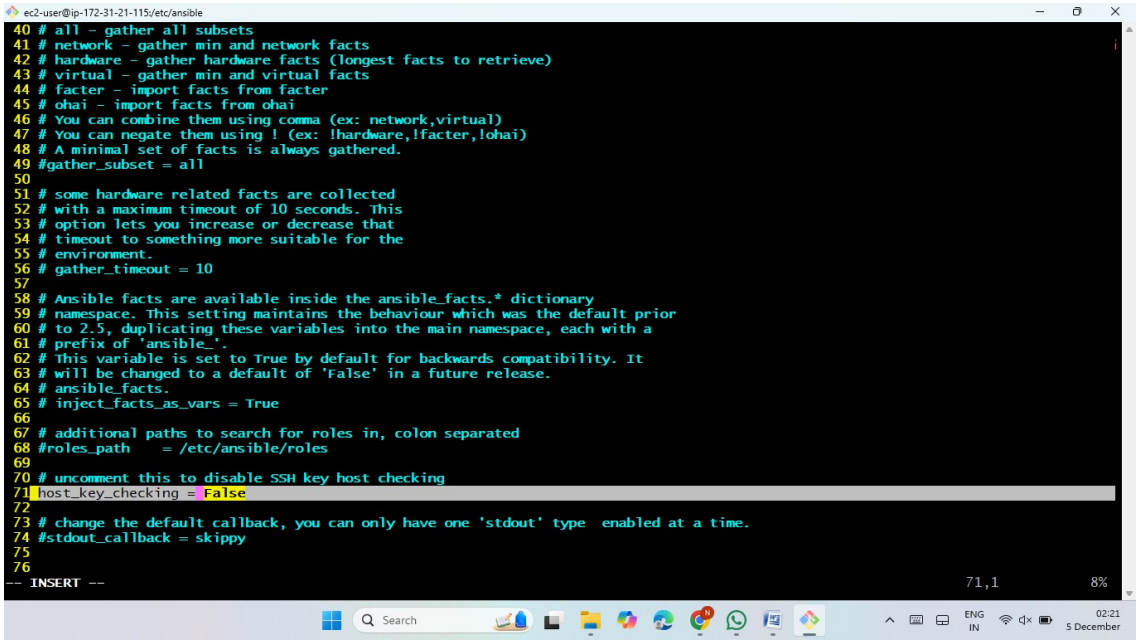


- Here the private key was copied then give permission to the key file.

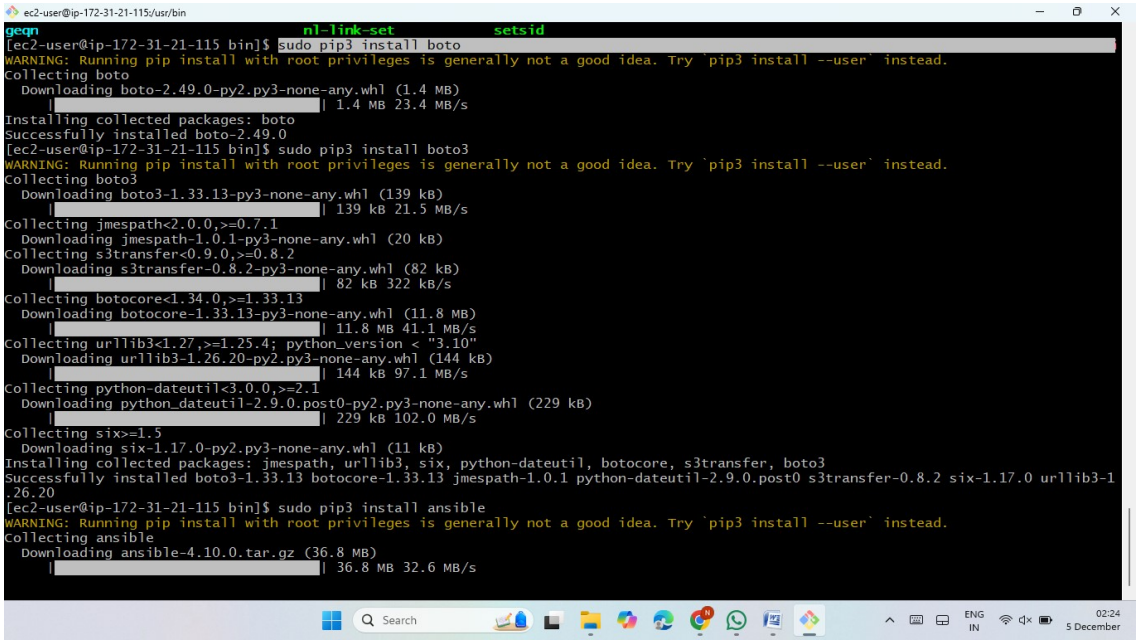


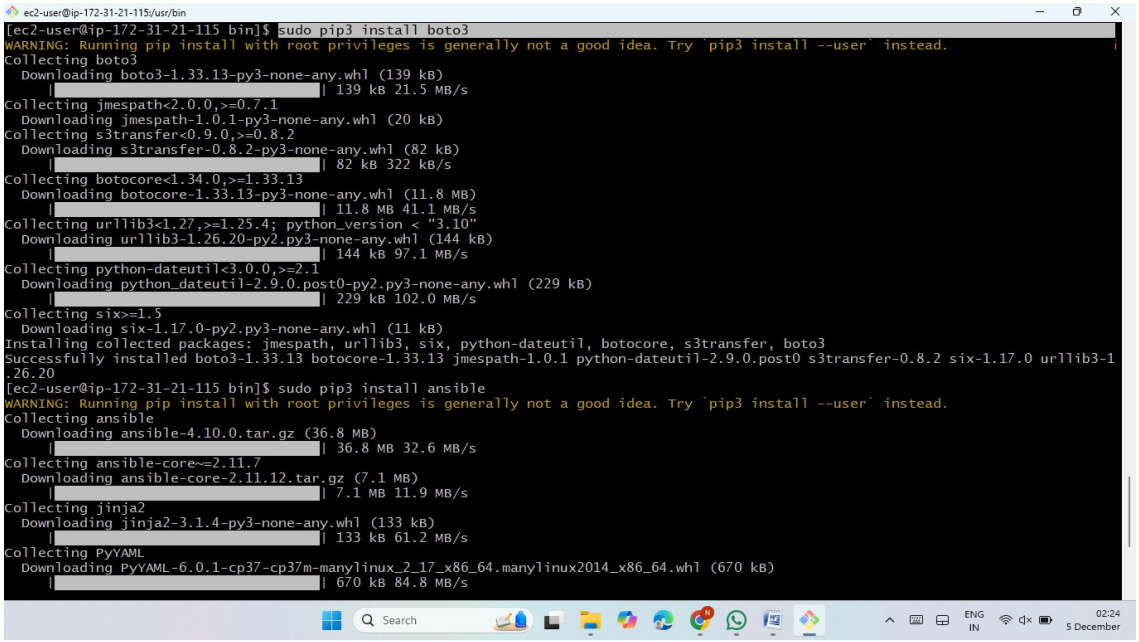
- Now go to ansible.cfg file (path= /ect/ansible)
- Uncomment the 71 th line in the ansible configuration file.

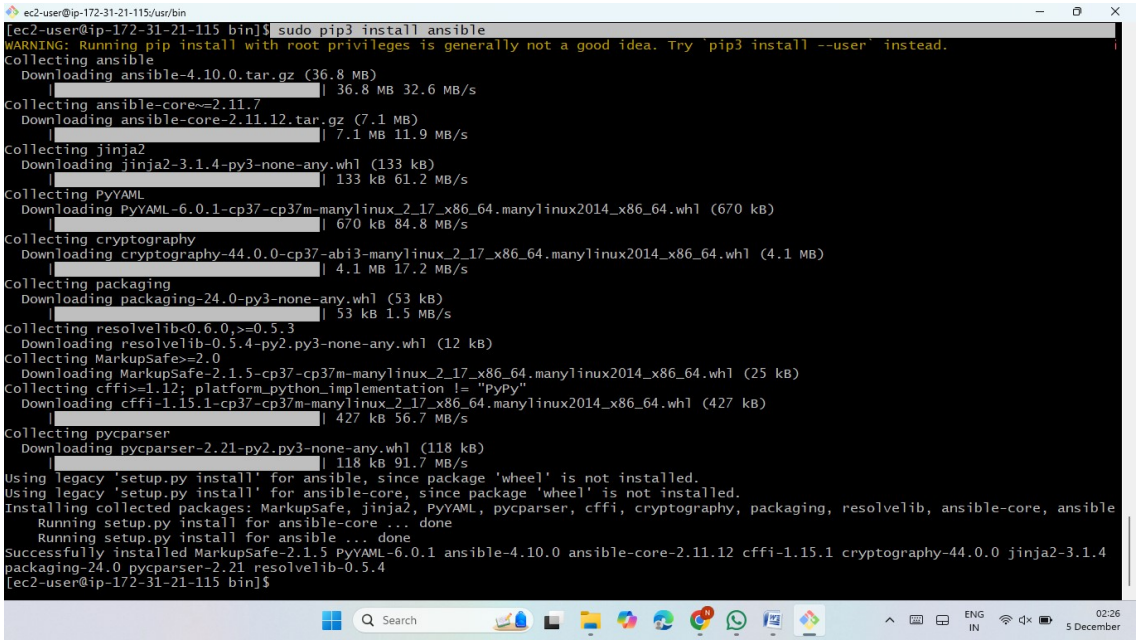




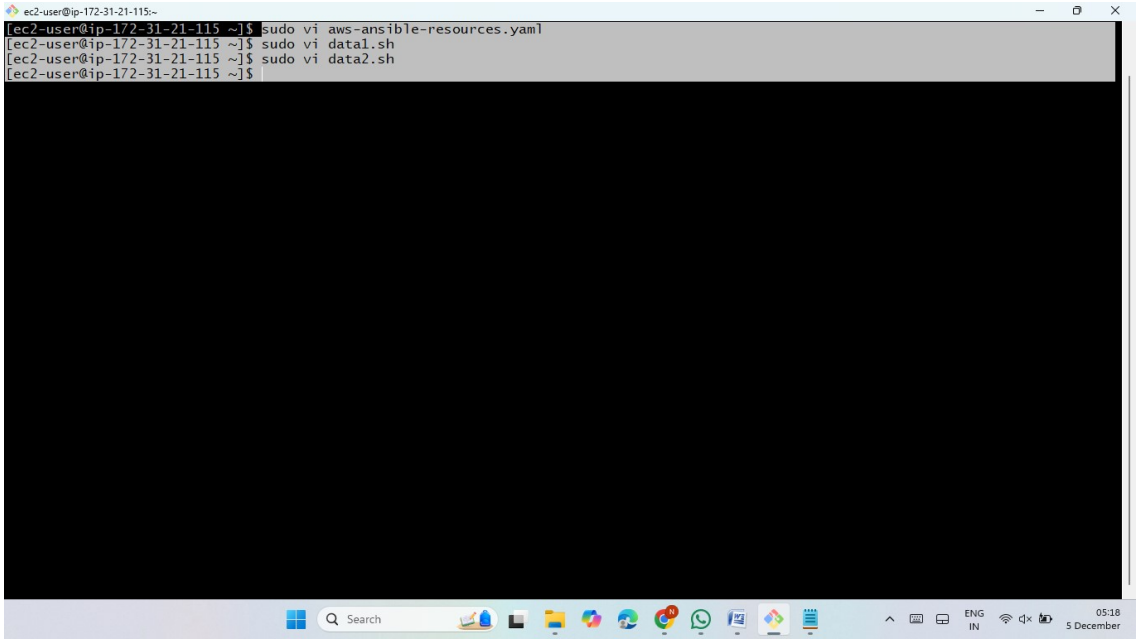
- Now go to /usr/bin/ path then install boto, boto3, ansible with the command.
  Sudo pip3 install boto boto3 ansible.

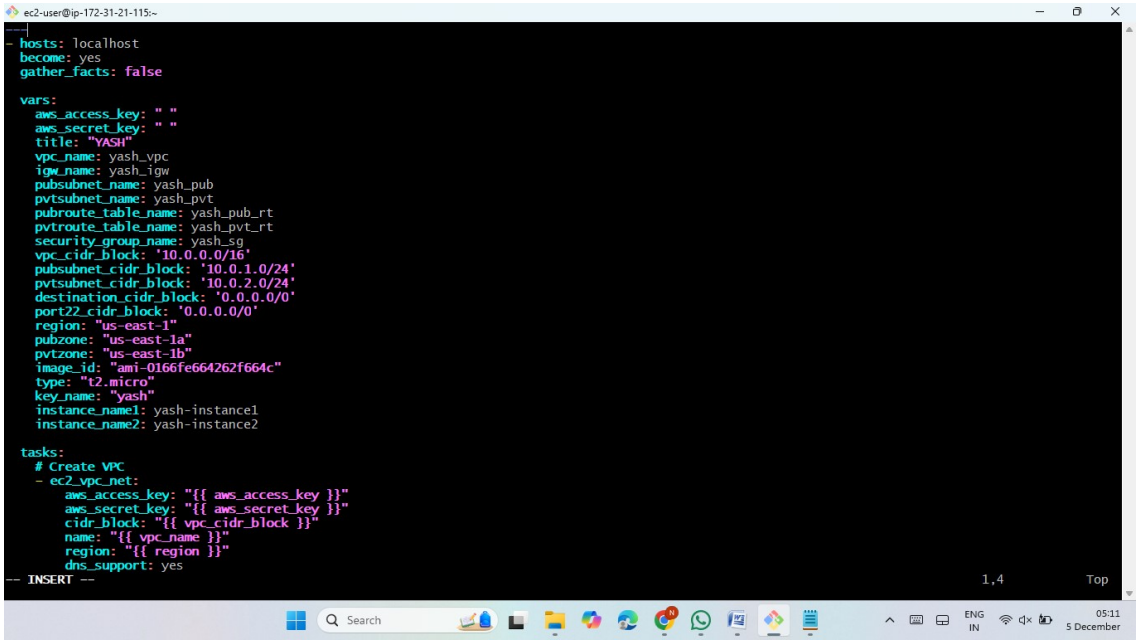




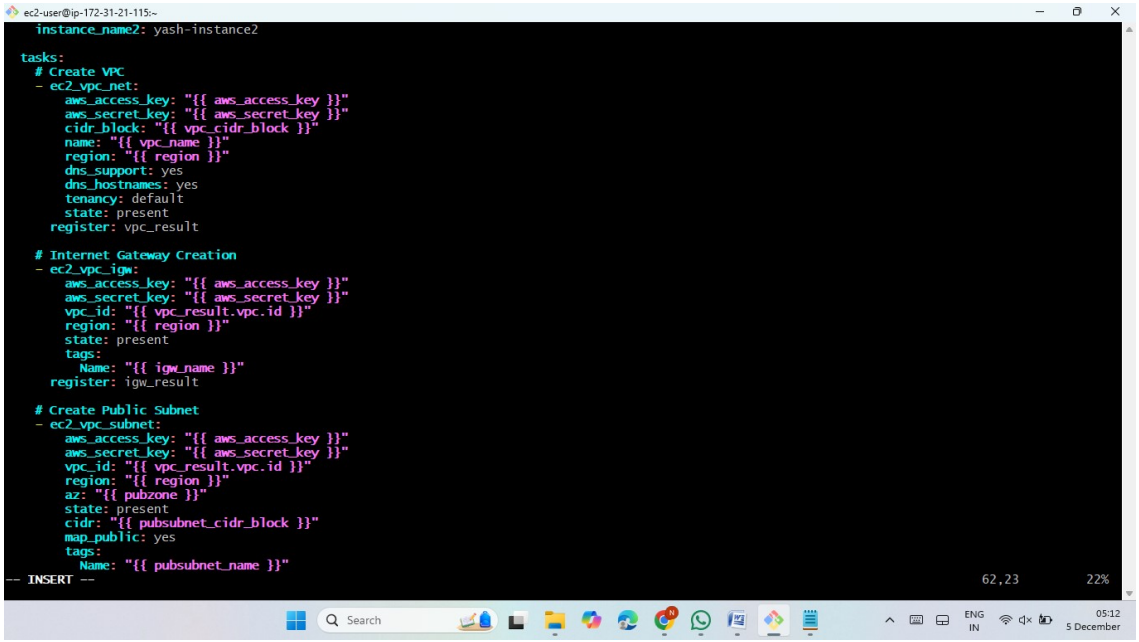


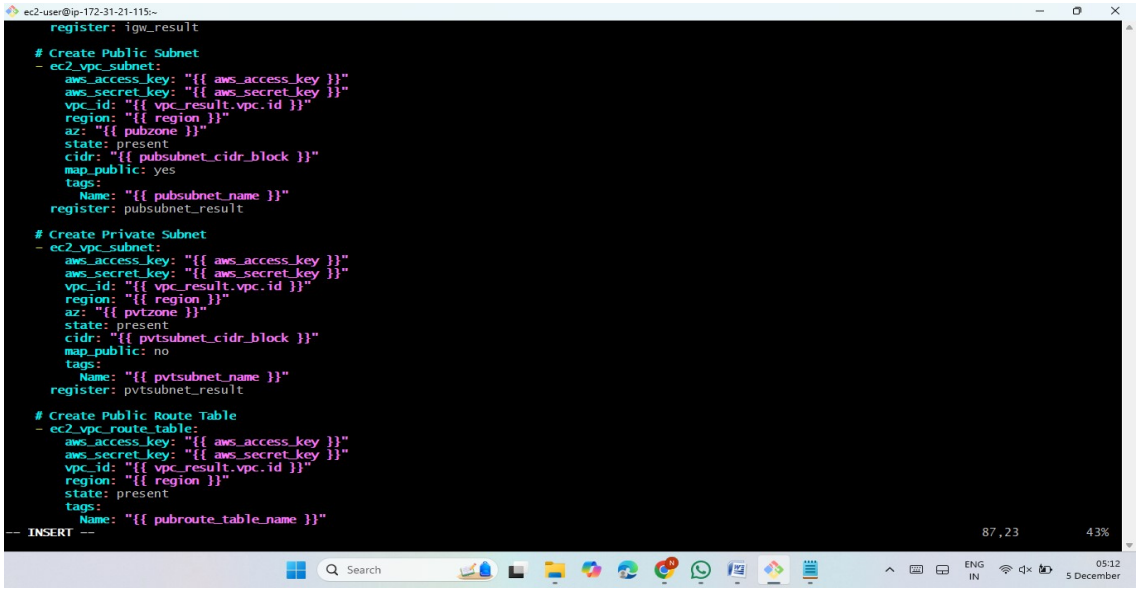
- Now create a file for host the static and dynamic application with the Ansible Script(yaml).
- Here write a variable file in the same script (Inline).
- First create a file for host 2 Instances, VPC, IGW, Private and Public Subnets, Private and Public Route Tables, Security Group.
- Now write a bash script for host static application in instance1.
- Now write a bash script for host static application in instance2.

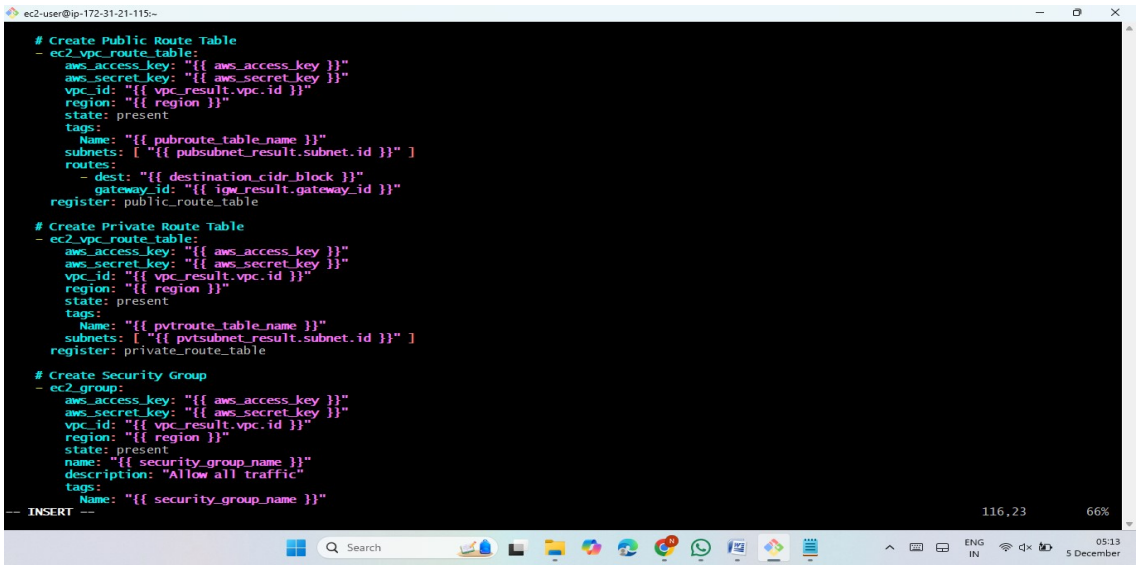


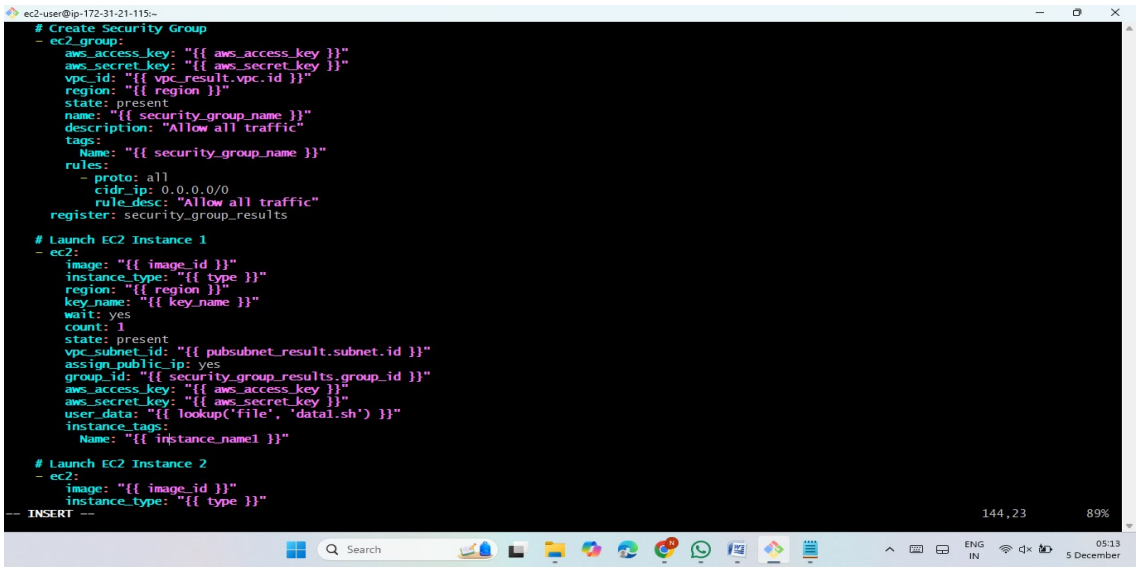

ec2-user@ip-172-31-21-115:~

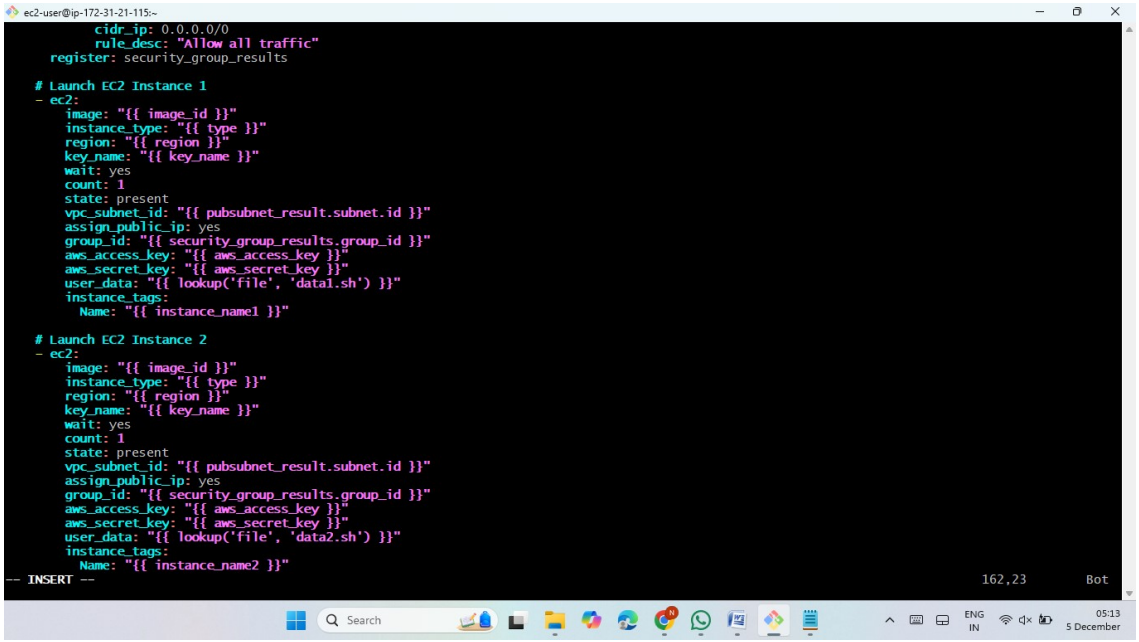
- hosts: localhost
become: yes
gather_facts: false
vars:
aws_access_key: " "
aws_secret_key: " "
title: "YASH"
vpc_name: yash_vpc
igw_name: yash_igw
pubsubnet_name: yash_pub
pvtsubnet_name: yash_pvt
pubroute_table_name: yash_pub_rt
pvtroute_table_name: yash_pvt_rt
security_group_name: yash_sg
vpc_cidr_block: '10.0.0.0/16'
pubsubnet_cidr_block: '10.0.1.0/24'
pvtsubnet_cidr_block: '10.0.2.0/24'
destination_cidr_block: '0.0.0.0/0'
port22_cidr_block: '0.0.0.0/0'
region: "us-east-1"
pubzone: "us-east-1a"
pvtzone: "us-east-1b"
image_id: "ami-0166fe664262f664c"
type: "t2.micro"
key_name: "yash"
instance_name1: yash-instance1
instance_name2: yash-instance2
tasks:
Create VPC
- ec2_vpc_net:
aws_access_key: "{{ aws_access_key }}"
aws_secret_key: "{{ aws_secret_key }}"
cidr_block: "{{ vpc_cidr_block }}"
name: "{{ vpc_name }}"
region: "{{ region }}"
dns_support: yes
-- INSERT --
1,4
Top
Search
ENG
IN
05:11
5 December

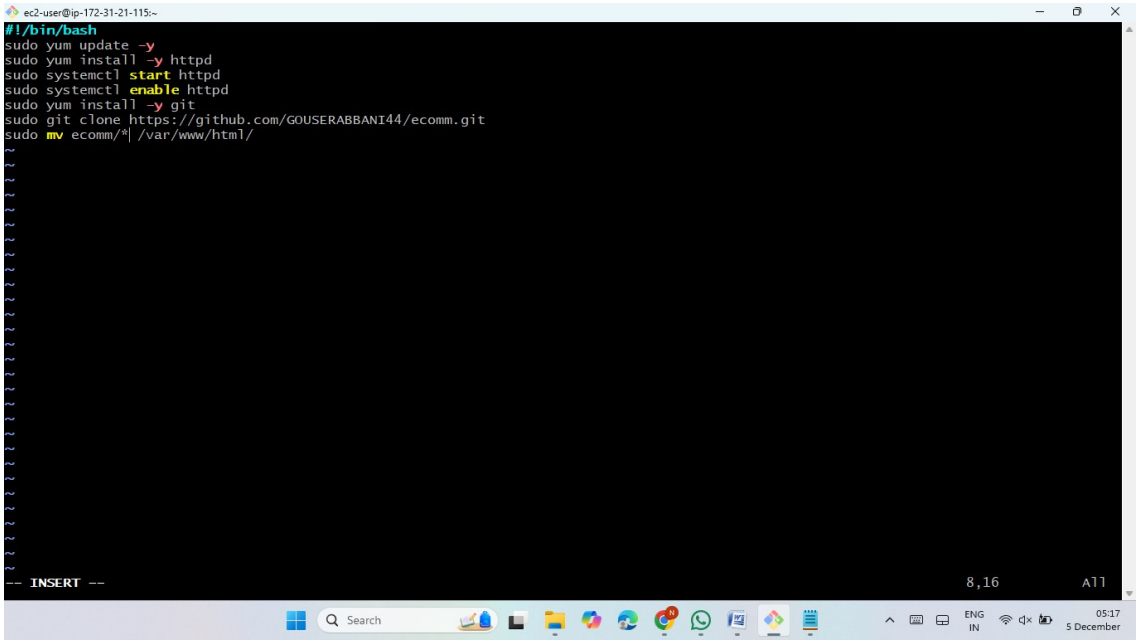

ec2-user@ip-172-31-21-115:~
instance_name2: yash-instance2
tasks:
Create VPC
- ec2_vpc_net:
aws_access_key: "{{ aws_access_key }}"
aws_secret_key: "{{ aws_secret_key }}"
cidr_block: "{{ vpc_cidr_block }}"
name: "{{ vpc_name }}"
region: "{{ region }}"
dns_support: yes
dns_hostnames: yes
tenancy: default
state: present
register: vpc_result
Internet Gateway Creation
- ec2_vpc_igw:
aws_access_key: "{{ aws_access_key }}"
aws_secret_key: "{{ aws_secret_key }}"
vpc_id: "{{ vpc_result.vpc.id }}"
region: "{{ region }}"
state: present
tags:
Name: "{{ igw_name }}"
register: igw_result
Create Public Subnet
- ec2_vpc_subnet:
aws_access_key: "{{ aws_access_key }}"
aws_secret_key: "{{ aws_secret_key }}"
vpc_id: "{{ vpc_result.vpc.id }}"
region: "{{ region }}"
az: "{{ pubzone }}"
state: present
cidr: "{{ pubsubnet_cidr_block }}"
map_public: yes
tags:
Name: "{{ pubsubnet_name }}"
-- INSERT --
62,23
22%
Search
ENG
IN
05:12
5 December

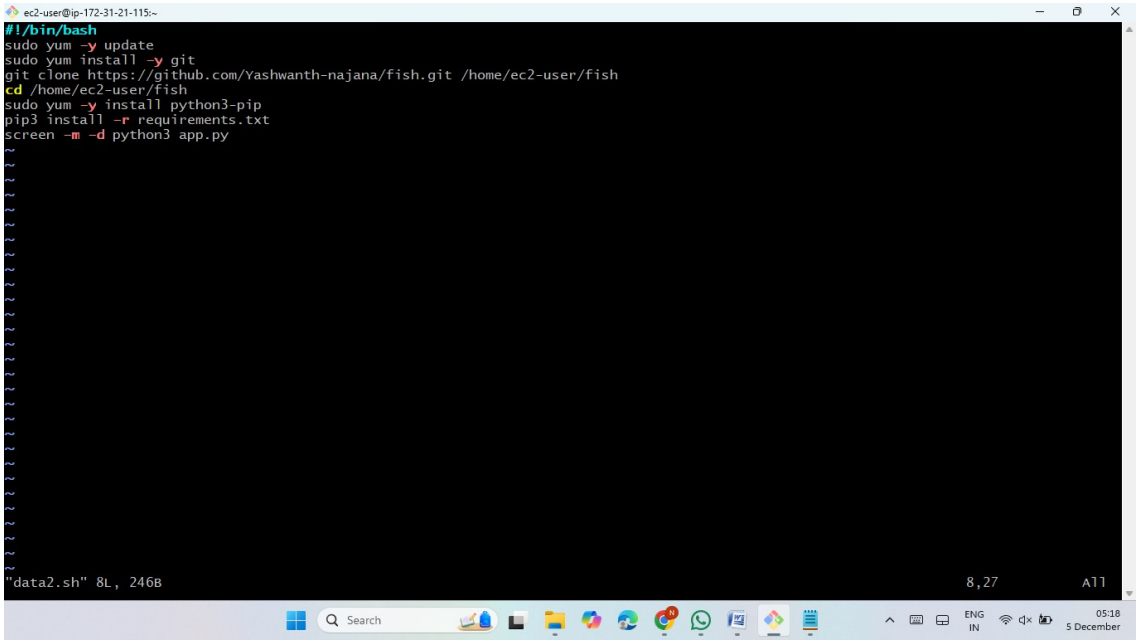

ec2-user@ip-172-31-21-115:~
register: igw_result
Create Public Subnet
- ec2_vpc_subnet:
aws_access_key: "{{ aws_access_key }}"
aws_secret_key: "{{ aws_secret_key }}"
vpc_id: "{{ vpc_result.vpc.id }}"
region: "{{ region }}"
az: "{{ pubzone }}"
state: present
cidr: "{{ pubsubnet_cidr_block }}"
map_public: yes
tags:
Name: "{{ pubsubnet_name }}"
register: pubsubnet_result
Create Private Subnet
- ec2_vpc_subnet:
aws_access_key: "{{ aws_access_key }}"
aws_secret_key: "{{ aws_secret_key }}"
vpc_id: "{{ vpc_result.vpc.id }}"
region: "{{ region }}"
az: "{{ pvtzone }}"
state: present
cidr: "{{ pvtsubnet_cidr_block }}"
map_public: no
tags:
Name: "{{ pvtsubnet_name }}"
register: pvtsubnet_result
Create Public Route Table
- ec2_vpc_route_table:
aws_access_key: "{{ aws_access_key }}"
aws_secret_key: "{{ aws_secret_key }}"
vpc_id: "{{ vpc_result.vpc.id }}"
region: "{{ region }}"
state: present
tags:
Name: "{{ pubroute_table_name }}"
-- INSERT --
87,23
43%
Search
ENG
IN
05:12
5 December


ec2-user@ip-172-31-21-115:~
Create Public Route Table
- ec2_vpc_route_table:
aws_access_key: "{{ aws_access_key }}"
aws_secret_key: "{{ aws_secret_key }}"
vpc_id: "{{ vpc_result.vpc.id }}"
region: "{{ region }}"
state: present
tags:
Name: "{{ pubroute_table_name }}"
subnets: ["{{ pubsubnet_result.subnet.id }}"]
routes:
- dest: "{{ destination_cidr_block }}"
gateway_id: "{{ igw_result.gateway_id }}"
register: public_route_table
Create Private Route Table
- ec2_vpc_route_table:
aws_access_key: "{{ aws_access_key }}"
aws_secret_key: "{{ aws_secret_key }}"
vpc_id: "{{ vpc_result.vpc.id }}"
region: "{{ region }}"
state: present
tags:
Name: "{{ pvtroute_table_name }}"
subnets: ["{{ pvtsubnet_result.subnet.id }}"]
register: private_route_table
Create Security Group
- ec2_group:
aws_access_key: "{{ aws_access_key }}"
aws_secret_key: "{{ aws_secret_key }}"
vpc_id: "{{ vpc_result.vpc.id }}"
region: "{{ region }}"
state: present
name: "{{ security_group_name }}"
description: "Allow all traffic"
tags:
Name: "{{ security_group_name }}"
-- INSERT --
116,23
66%
Search
ENG
IN
05:13
5 December

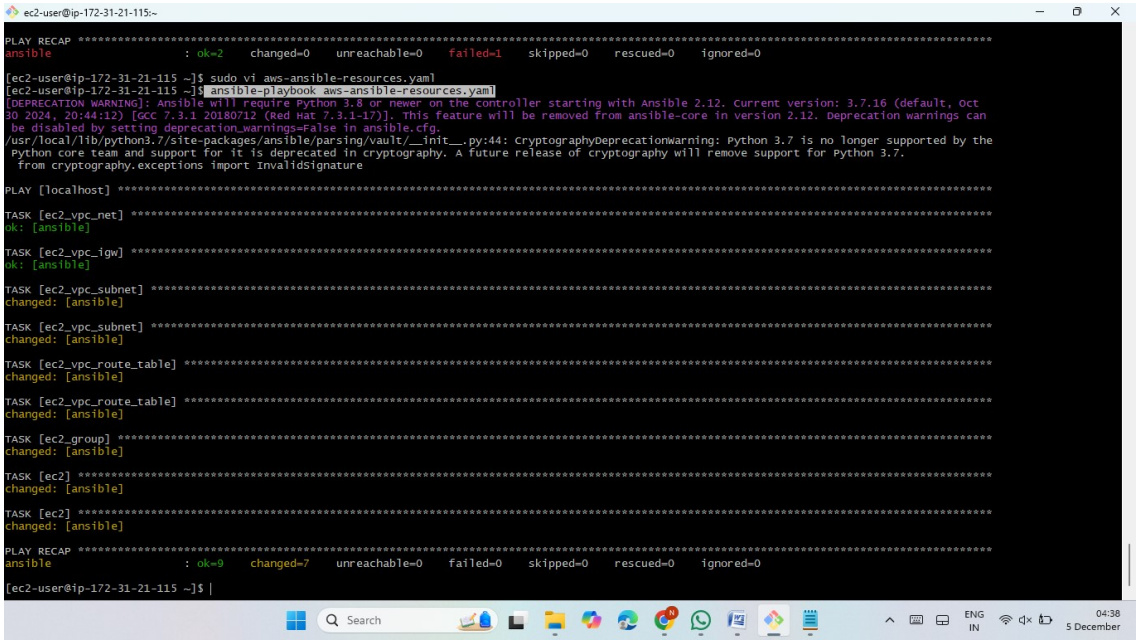

ec2-user@ip-172-31-21-115:~
Create Security Group
- ec2_group:
aws_access_key: "{{ aws_access_key }}"
aws_secret_key: "{{ aws_secret_key }}"
vpc_id: "{{ vpc_result.vpc.id }}"
region: "{{ region }}"
state: present
name: "{{ security_group_name }}"
description: "Allow all traffic"
tags:
Name: "{{ security_group_name }}"
rules:
- proto: all
cidr_ip: 0.0.0.0/0
rule_desc: "Allow all traffic"
register: security_group_results
Launch EC2 Instance 1
- ec2:
image: "{{ image_id }}"
instance_type: "{{ type }}"
region: "{{ region }}"
key_name: "{{ key_name }}"
wait: yes
count: 1
state: present
vpc_subnet_id: "{{ pubsubnet_result.subnet.id }}"
assign_public_ip: yes
group_id: "{{ security_group_results.group_id }}"
aws_access_key: "{{ aws_access_key }}"
aws_secret_key: "{{ aws_secret_key }}"
user_data: "{{ lookup('file', 'data1.sh') }}"
instance_tags:
Name: "{{ instance_name1 }}"
Launch EC2 Instance 2
- ec2:
image: "{{ image_id }}"
instance_type: "{{ type }}"
-- INSERT --
144,23
89%
Search
ENG
IN
05:13
5 December


ec2-user@ip-172-31-21-115:~
cidr_ip: 0.0.0.0/0
rule_desc: "Allow all traffic"
register: security_group_results
Launch EC2 Instance 1
- ec2:
image: "{{ image_id }}"
instance_type: "{{ type }}"
region: "{{ region }}"
key_name: "{{ key_name }}"
wait: yes
count: 1
state: present
vpc_subnet_id: "{{ pubsubnet_result.subnet.id }}"
assign_public_ip: yes
group_id: "{{ security_group_results.group_id }}"
aws_access_key: "{{ aws_access_key }}"
aws_secret_key: "{{ aws_secret_key }}"
user_data: "{{ lookup('file', 'data1.sh') }}"
instance_tags:
Name: "{{ instance_name1 }}"
Launch EC2 Instance 2
- ec2:
image: "{{ image_id }}"
instance_type: "{{ type }}"
region: "{{ region }}"
key_name: "{{ key_name }}"
wait: yes
count: 1
state: present
vpc_subnet_id: "{{ pubsubnet_result.subnet.id }}"
assign_public_ip: yes
group_id: "{{ security_group_results.group_id }}"
aws_access_key: "{{ aws_access_key }}"
aws_secret_key: "{{ aws_secret_key }}"
user_data: "{{ lookup('file', 'data2.sh') }}"
instance_tags:
Name: "{{ instance_name2 }}"
-- INSERT --
162,23
Bot
Search
ENG
IN
05:13
5 December

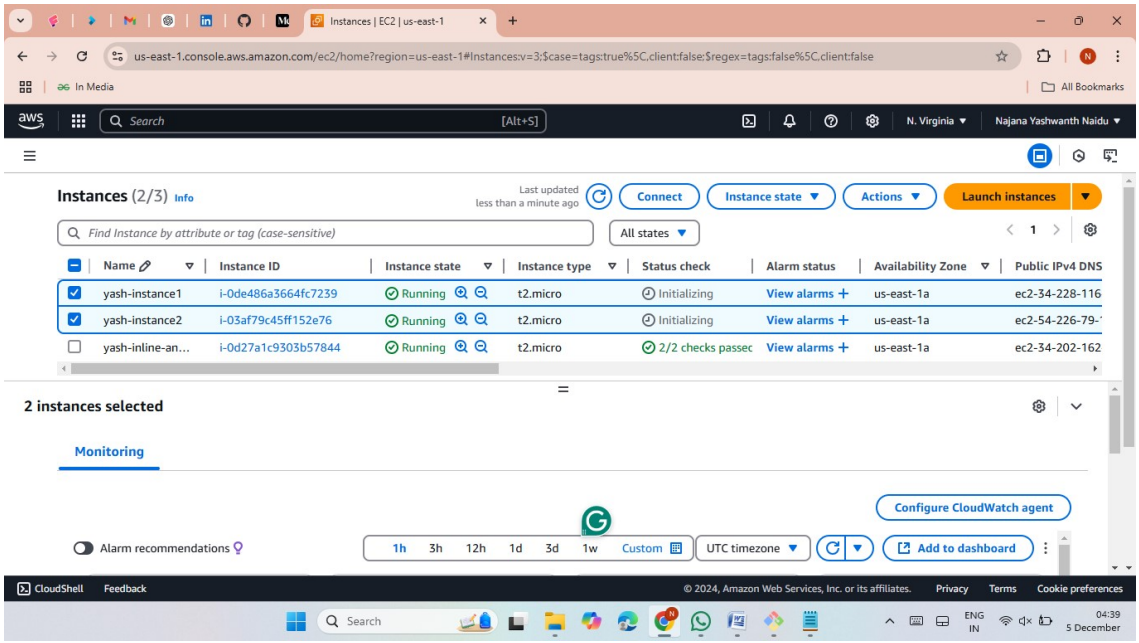

ec2-user@ip-172-31-21-115:~
#!/bin/bash
sudo yum update -y
sudo yum install -y httpd
sudo systemctl start httpd
sudo systemctl enable httpd
sudo yum install -y git
sudo git clone https://github.com/GOUSERABBANI44/ecomm.git
sudo mv ecomm/* /var/www/html/
-- INSERT --
8,16
All
Search
ENG
IN
05:17
5 December

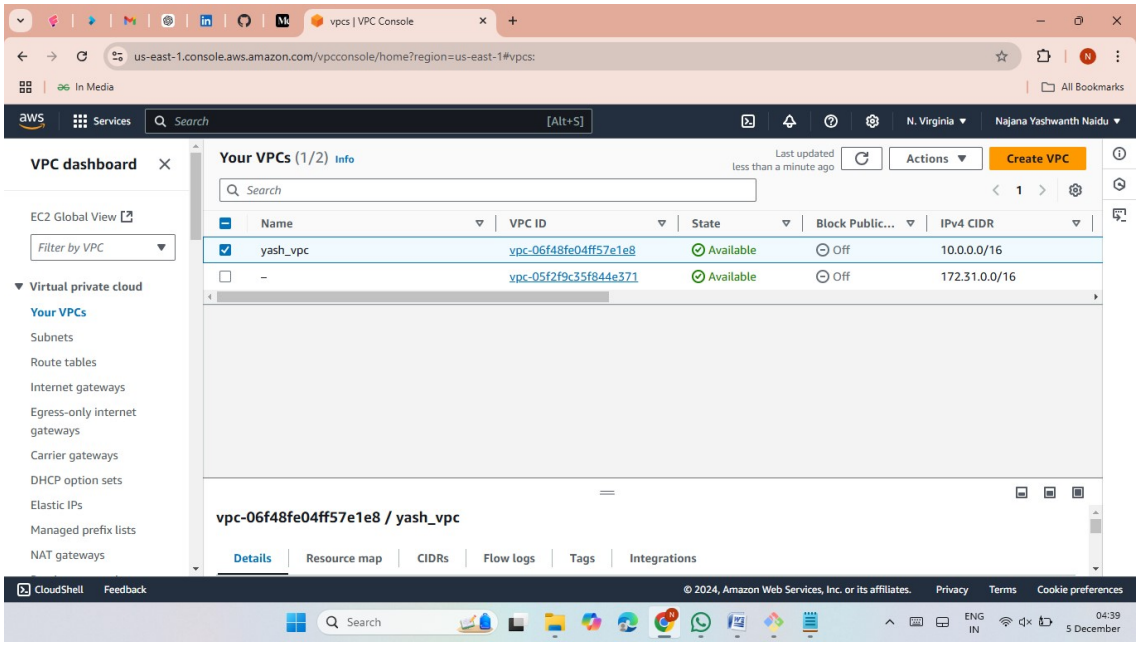


- Now run the AWS resources with the command.
  Ansible-playbook <name of file>

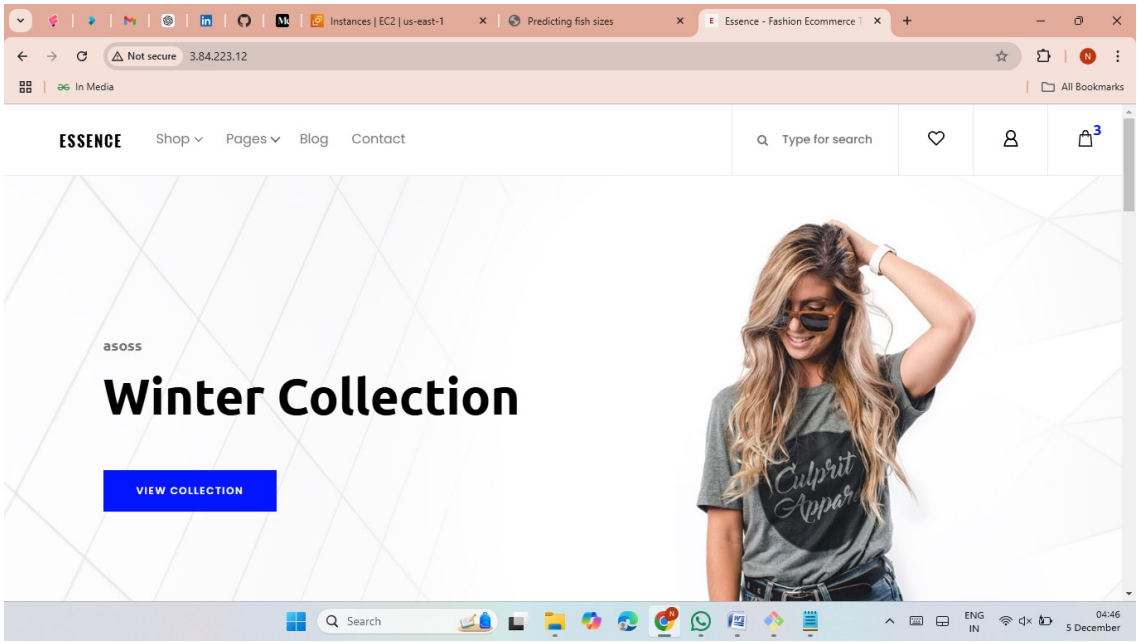


- Here the resources was created successfully.
- Here the 2 instances was launched.
- Here the VPC, IGW, Subnets, Route Tables and Security group was created.

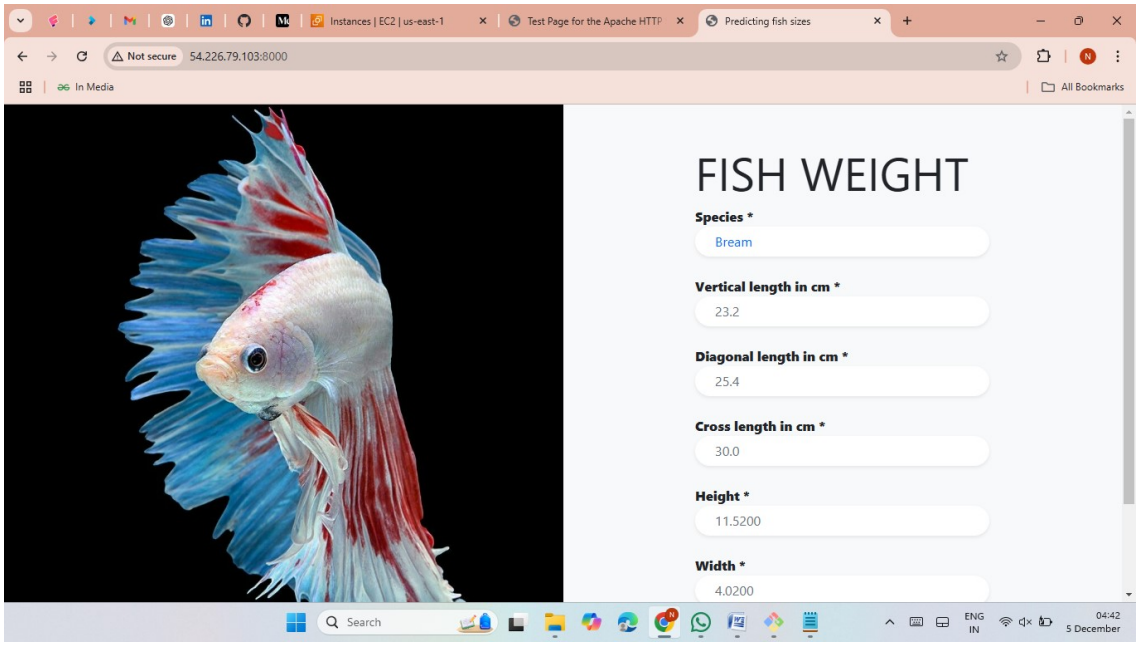




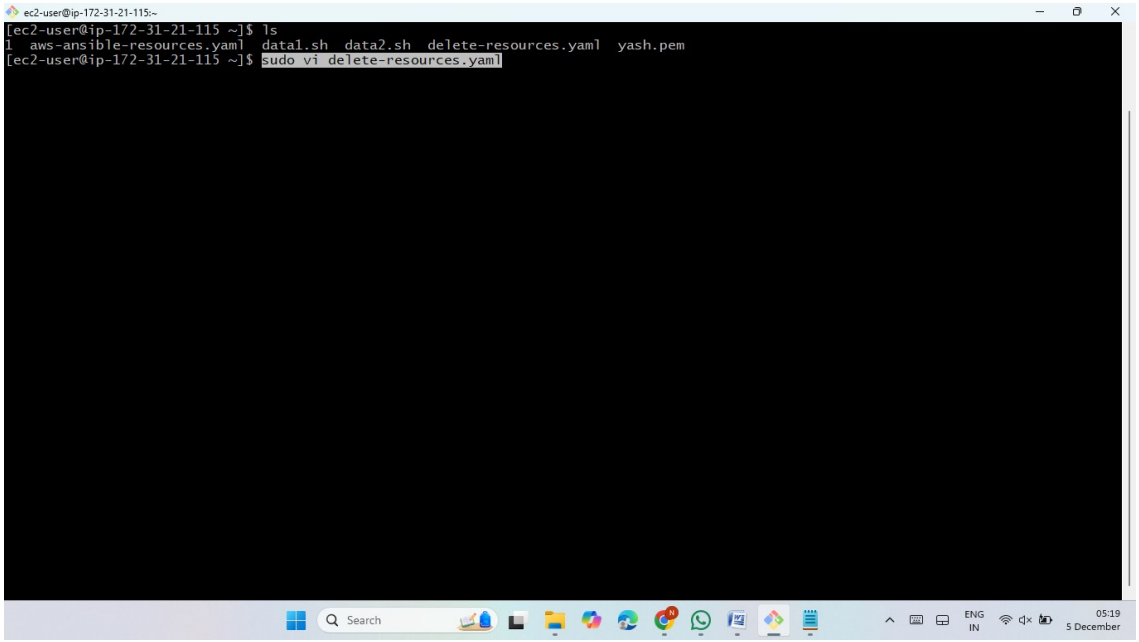
- Now copy the Instance 1 Public IP then Past it on Google along with port 80.
- The static ecomm application was hosted successfully.

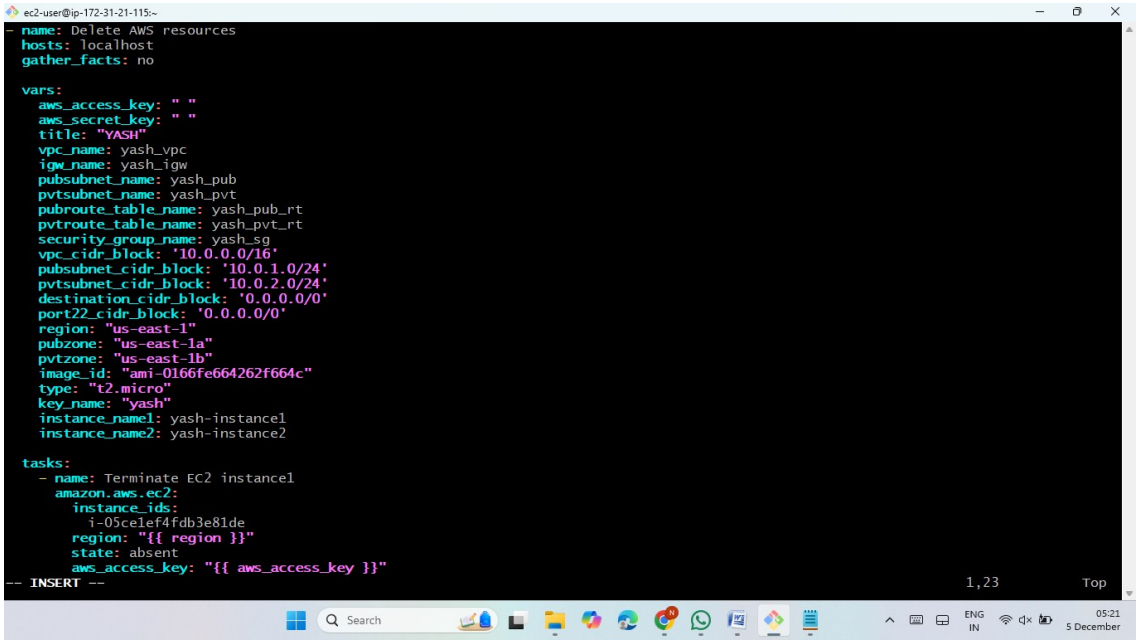


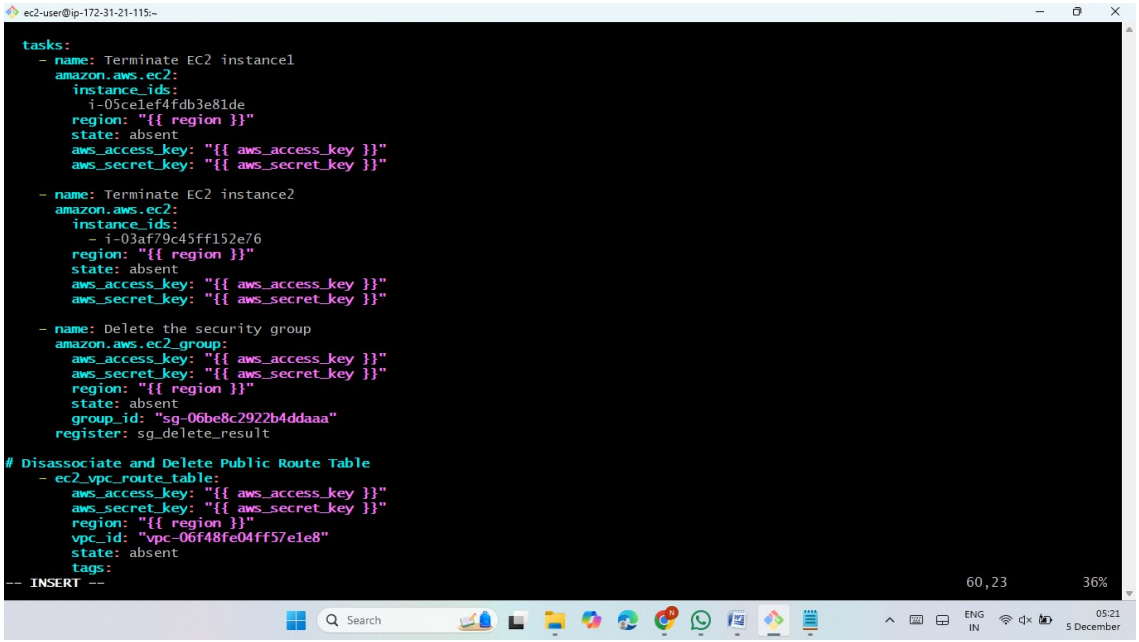
- Now copy the Instance2 Public IP then Past it on Google along with port 8000.
- The dynamic python application was hosted successfully.

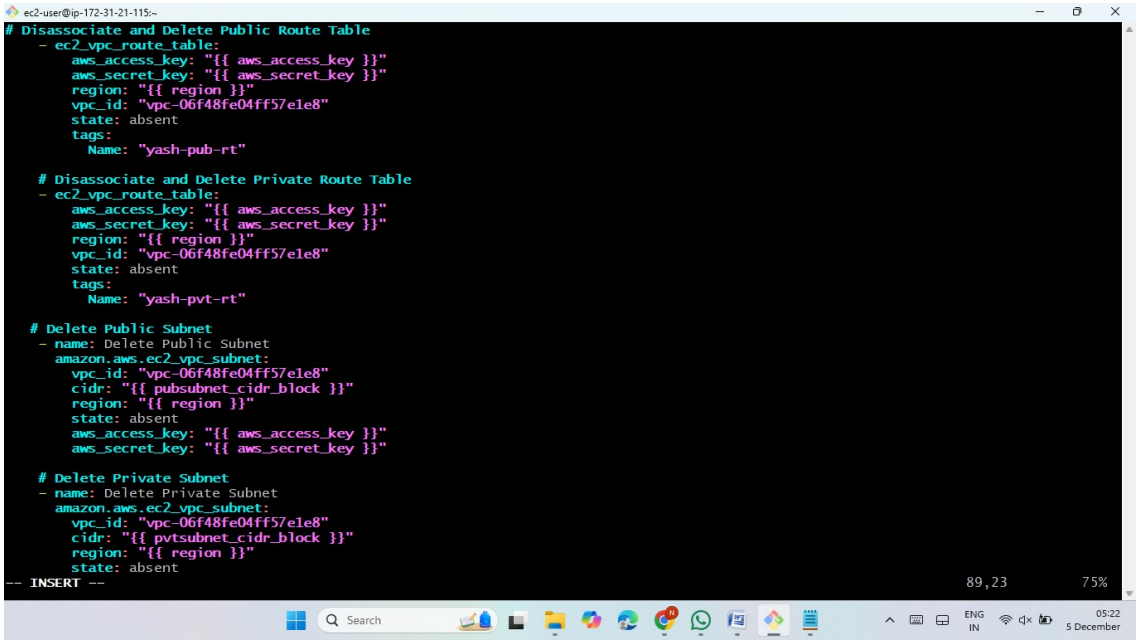


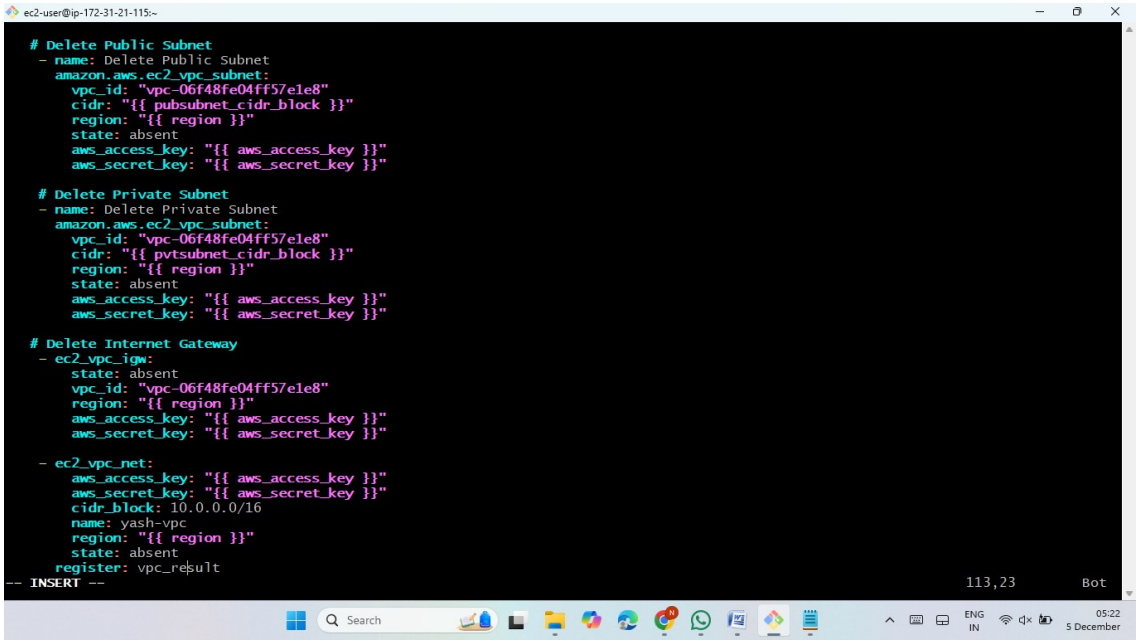
- Now create file for delete the created resources.
- Now run the file with the command **ansible-playbook <name of the file>.**

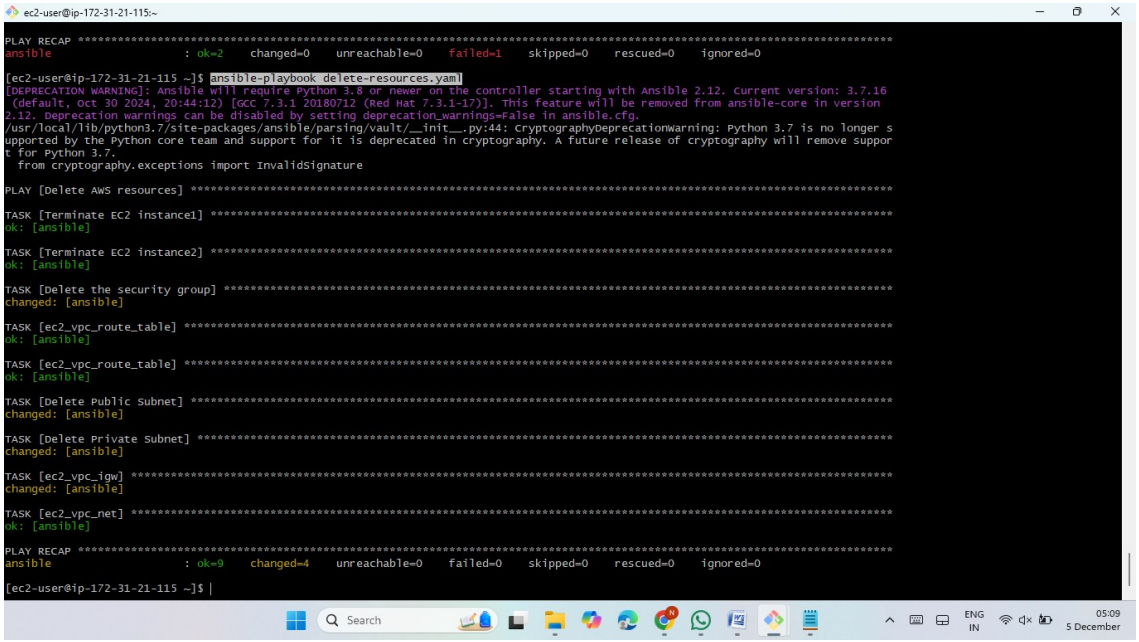


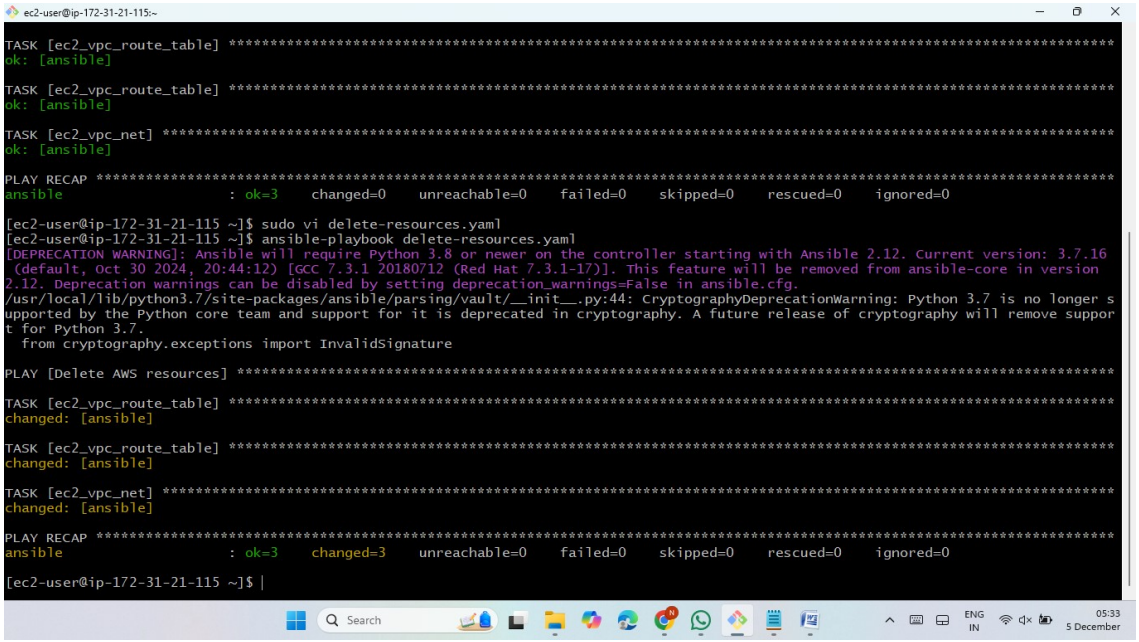



ec2-user@ip-172-31-21-115:~
tasks:
- name: Terminate EC2 instance1
amazon.aws.ec2:
instance_ids:
i-05ce1ef4fdb3e81de
region: "{{ region }}"
state: absent
aws_access_key: "{{ aws_access_key }}"
aws_secret_key: "{{ aws_secret_key }}"
- name: Terminate EC2 instance2
amazon.aws.ec2:
instance_ids:
- i-03af79c45ff152e76
region: "{{ region }}"
state: absent
aws_access_key: "{{ aws_access_key }}"
aws_secret_key: "{{ aws_secret_key }}"
- name: Delete the security group
amazon.aws.ec2_group:
aws_access_key: "{{ aws_access_key }}"
aws_secret_key: "{{ aws_secret_key }}"
region: "{{ region }}"
state: absent
group_id: "sg-06be8c2922b4ddaaa"
register: sg_delete_result
Disassociate and Delete Public Route Table
- ec2_vpc_route_table:
aws_access_key: "{{ aws_access_key }}"
aws_secret_key: "{{ aws_secret_key }}"
region: "{{ region }}"
vpc_id: "vpc-06f48fe04ff57e1e8"
state: absent
tags:
-- INSERT --
60,23
36%
Search
ENG
IN
05:21
5 December

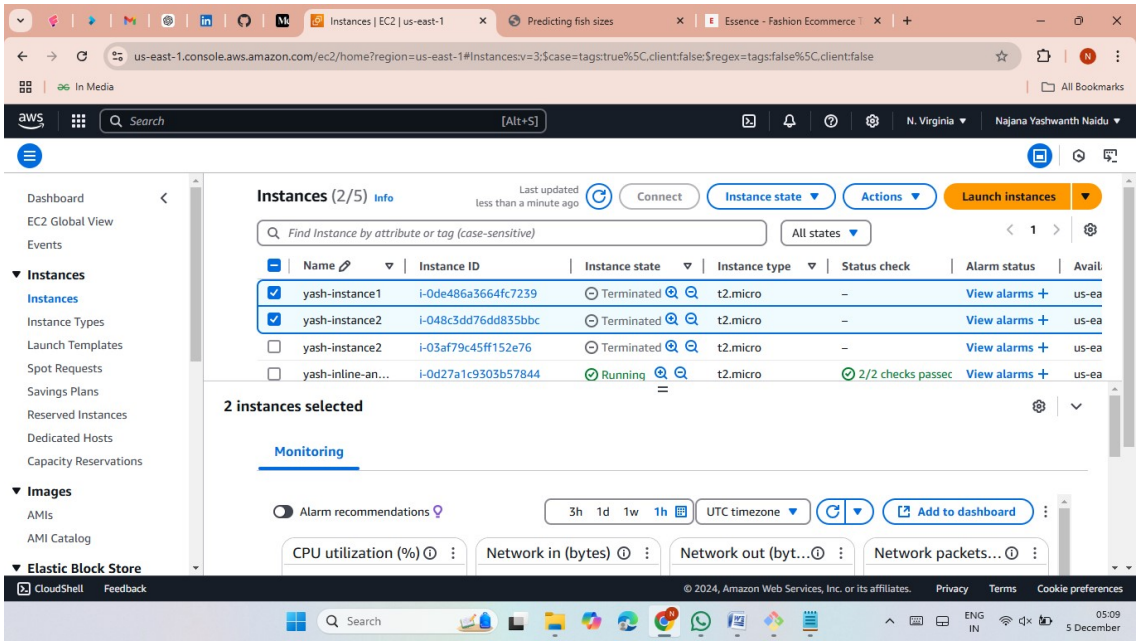

ec2-user@ip-172-31-21-115:~
Disassociate and Delete Public Route Table
- ec2_vpc_route_table:
aws_access_key: "{{ aws_access_key }}"
aws_secret_key: "{{ aws_secret_key }}"
region: "{{ region }}"
vpc_id: "vpc-06f48fe04ff57e1e8"
state: absent
tags:
Name: "yash-pub-rt"
Disassociate and Delete Private Route Table
- ec2_vpc_route_table:
aws_access_key: "{{ aws_access_key }}"
aws_secret_key: "{{ aws_secret_key }}"
region: "{{ region }}"
vpc_id: "vpc-06f48fe04ff57e1e8"
state: absent
tags:
Name: "yash-pvt-rt"
Delete Public Subnet
- name: Delete Public Subnet
amazon.aws.ec2_vpc_subnet:
vpc_id: "vpc-06f48fe04ff57e1e8"
cidr: "{{ pubsubnet_cidr_block }}"
region: "{{ region }}"
state: absent
aws_access_key: "{{ aws_access_key }}"
aws_secret_key: "{{ aws_secret_key }}"
Delete Private Subnet
- name: Delete Private Subnet
amazon.aws.ec2_vpc_subnet:
vpc_id: "vpc-06f48fe04ff57e1e8"
cidr: "{{ pvtsubnet_cidr_block }}"
region: "{{ region }}"
state: absent
-- INSERT --
89,23
75%
Search
ENG
IN
05:22
5 December


ec2-user@ip-172-31-21-115:~
Delete Public Subnet
- name: Delete Public Subnet
amazon.aws.ec2_vpc_subnet:
vpc_id: "vpc-06f48fe04ff57e1e8"
cidr: "{{ pubsubnet_cidr_block }}"
region: "{{ region }}"
state: absent
aws_access_key: "{{ aws_access_key }}"
aws_secret_key: "{{ aws_secret_key }}"
Delete Private Subnet
- name: Delete Private Subnet
amazon.aws.ec2_vpc_subnet:
vpc_id: "vpc-06f48fe04ff57e1e8"
cidr: "{{ pvtsubnet_cidr_block }}"
region: "{{ region }}"
state: absent
aws_access_key: "{{ aws_access_key }}"
aws_secret_key: "{{ aws_secret_key }}"
Delete Internet Gateway
- ec2_vpc_igw:
state: absent
vpc_id: "vpc-06f48fe04ff57e1e8"
region: "{{ region }}"
aws_access_key: "{{ aws_access_key }}"
aws_secret_key: "{{ aws_secret_key }}"
- ec2_vpc_net:
aws_access_key: "{{ aws_access_key }}"
aws_secret_key: "{{ aws_secret_key }}"
cidr_block: 10.0.0.0/16
name: yash-vpc
region: "{{ region }}"
state: absent
register: vpc_result
-- INSERT --
113,23 Bot
Search
ENG IN
05:22 5 December

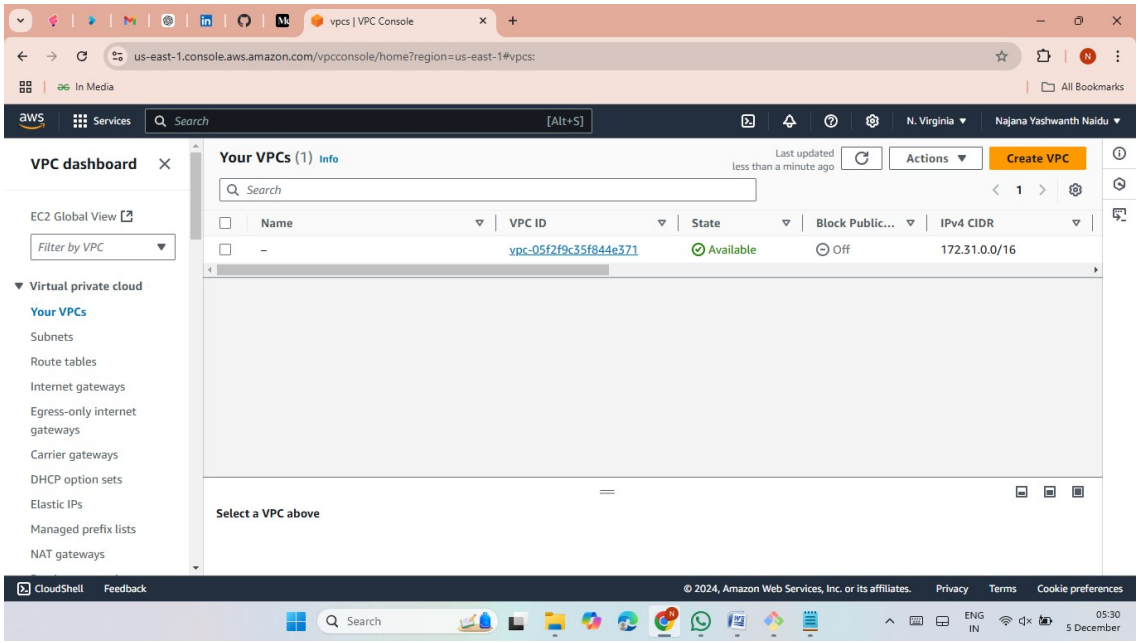

ec2-user@ip-172-31-21-115:~
PLAY RECAP
ansible : ok=2 changed=0 unreachable=0 failed=1 skipped=0 rescued=0 ignored=0
[ec2-user@ip-172-31-21-115 ~]$ ansible-playbook delete-resources.yaml
[DEPRECATION WARNING]: Ansible will require Python 3.8 or newer on the controller starting with Ansible 2.12. Current version: 3.7.16 (default, Oct 30 2024, 20:44:12) [GCC 7.3.1 20180712 (Red Hat 7.3.1-17)]. This feature will be removed from ansible-core in version 2.12. Deprecation warnings can be disabled by setting deprecation_warnings=False in ansible.cfg.
/usr/local/lib/python3.7/site-packages/ansible/parsing/vault/__init__.py:44: CryptographyDeprecationWarning: Python 3.7 is no longer supported by the Python core team and support for it is deprecated in cryptography. A future release of cryptography will remove support for Python 3.7.
from cryptography.exceptions import InvalidSignature
PLAY [Delete AWS resources]
TASK [Terminate EC2 instance1]
ok: [ansible]
TASK [Terminate EC2 instance2]
ok: [ansible]
TASK [Delete the security group]
changed: [ansible]
TASK [ec2_vpc_route_table]
ok: [ansible]
TASK [ec2_vpc_route_table]
ok: [ansible]
TASK [Delete Public Subnet]
changed: [ansible]
TASK [Delete Private Subnet]
changed: [ansible]
TASK [ec2_vpc_igw]
changed: [ansible]
TASK [ec2_vpc_net]
ok: [ansible]
PLAY RECAP
ansible : ok=9 changed=4 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0
[ec2-user@ip-172-31-21-115 ~]$
Search
ENG IN
05:09 5 December



- Here the created resources was deleted successfully.
- Here the instances was deleted.

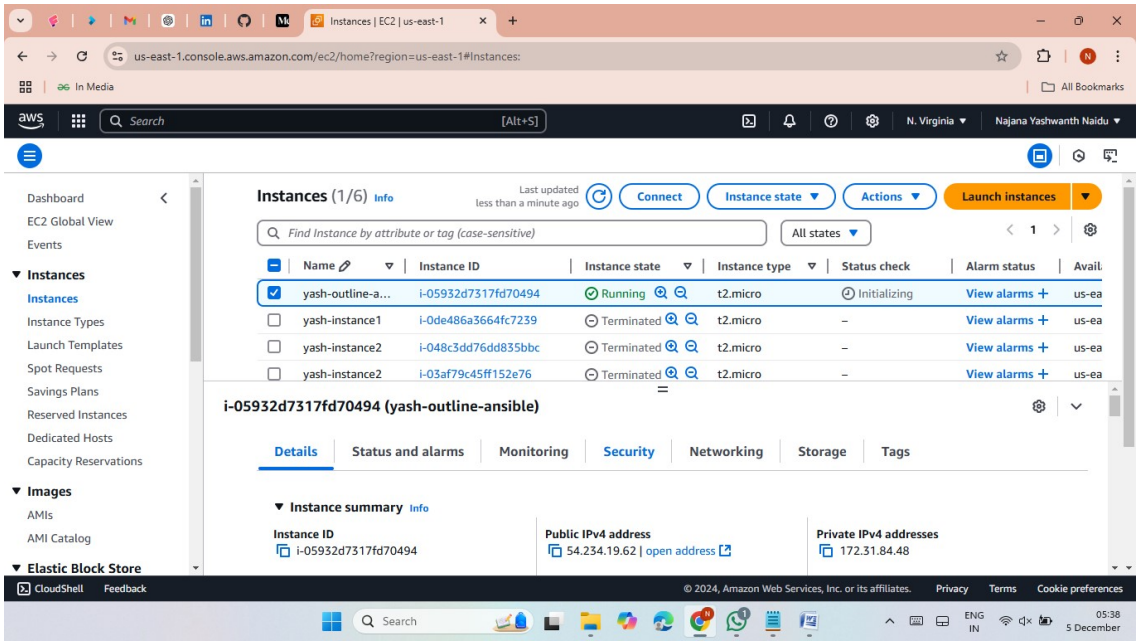


- Here the VPC was deleted.

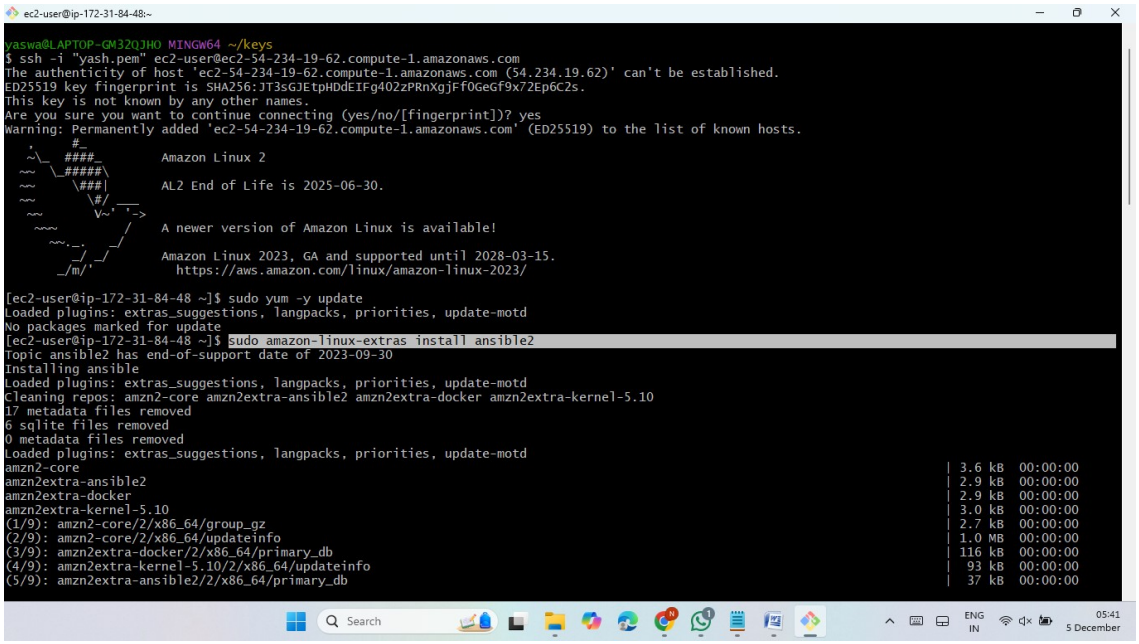


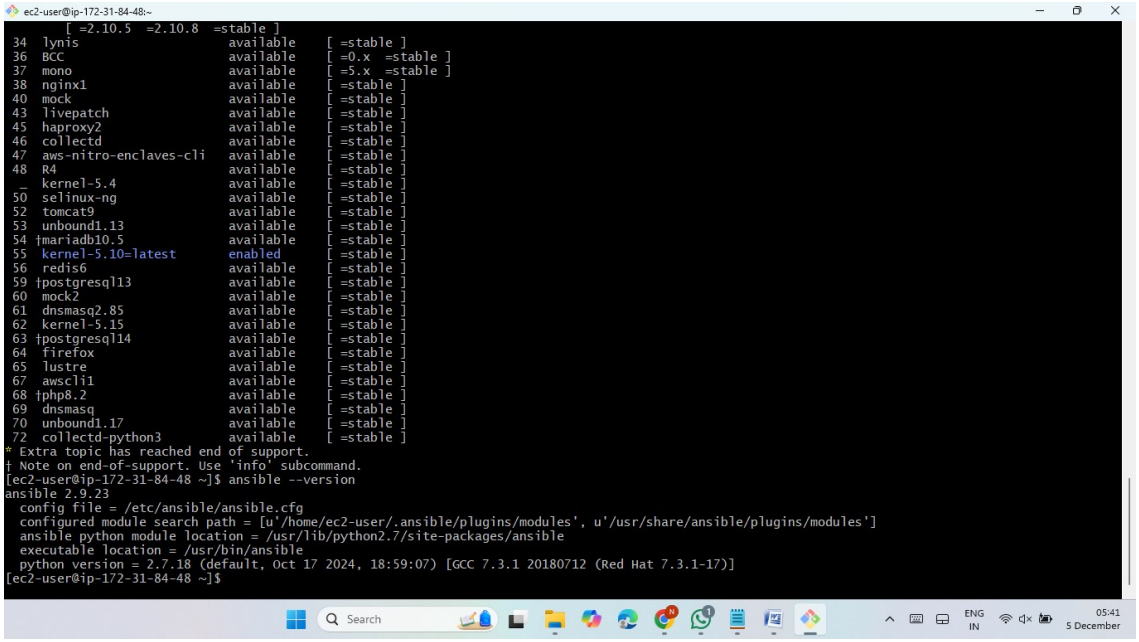
METHOD-3 (OUTLINE)

- Launch an Instances along with required ports.

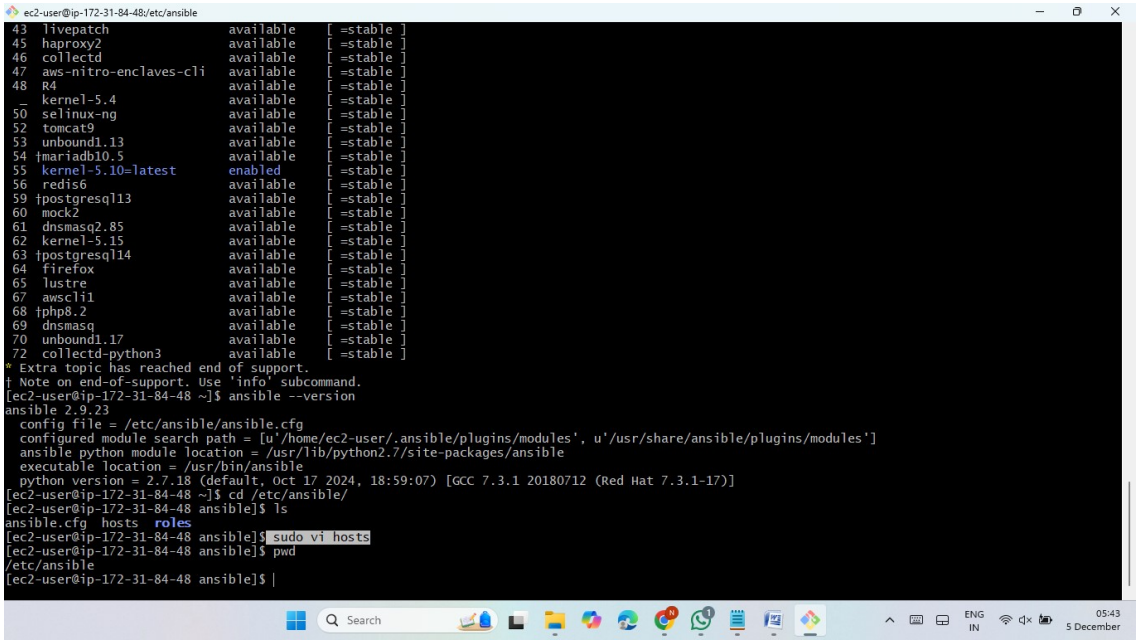


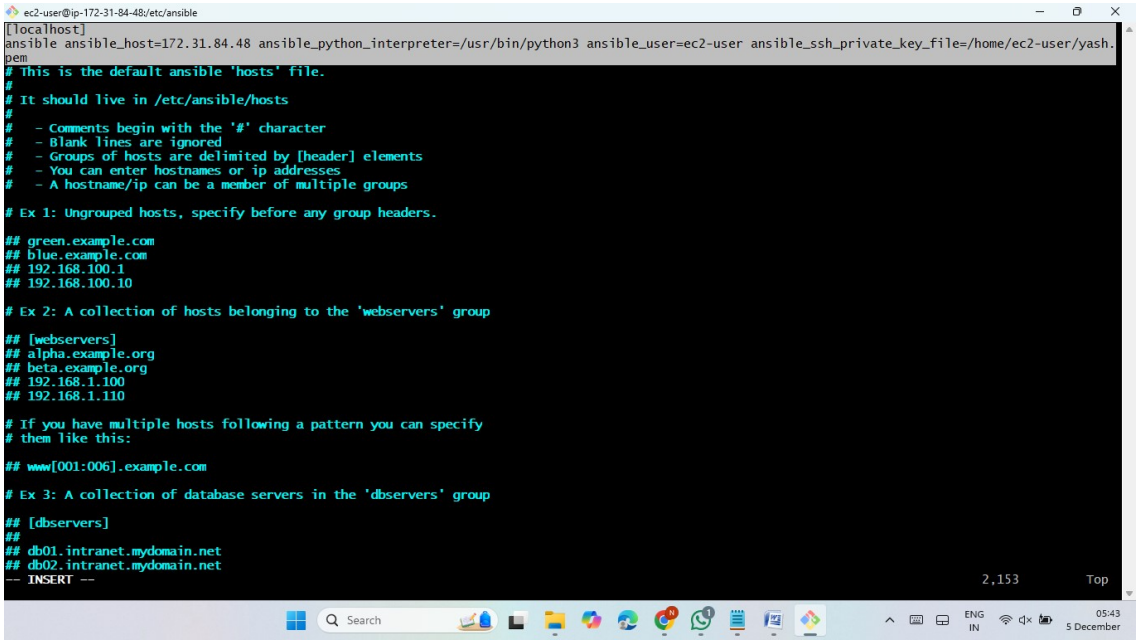
- Now connect to GitBash then install ansible2 with the command.
  Sudo amazon-linux-extras install ansible2



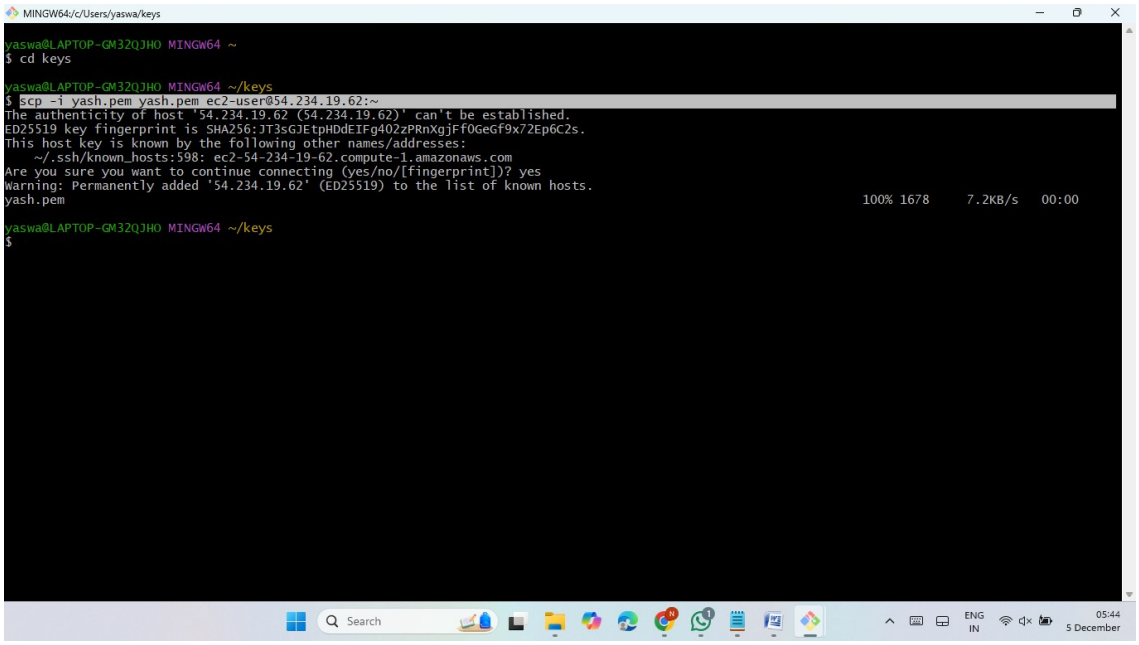


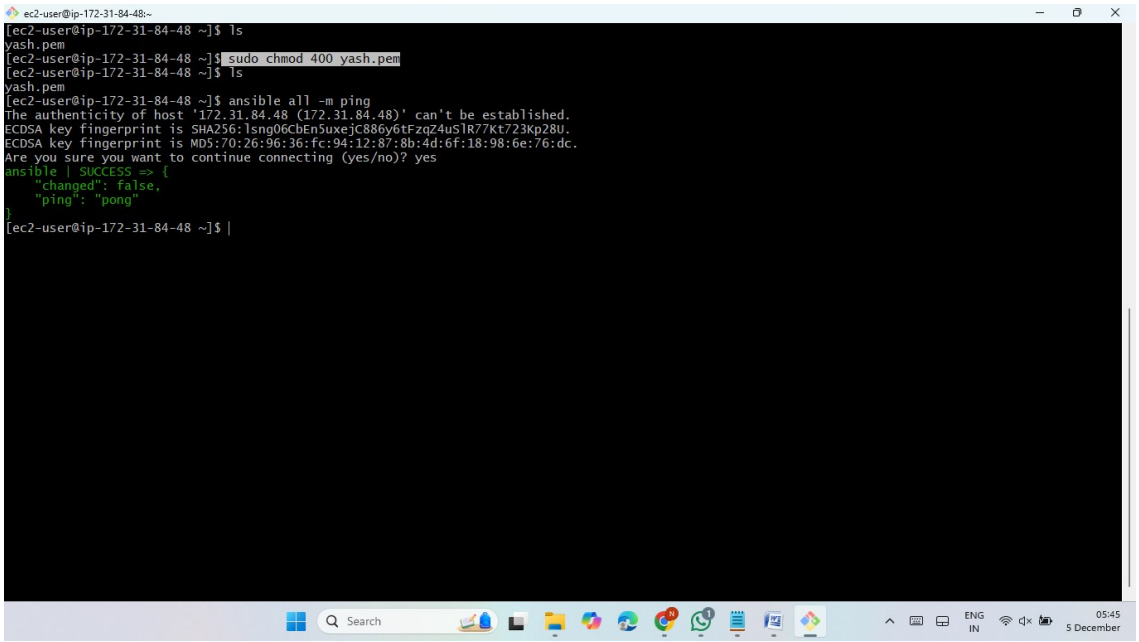
- Now Edit host file (path= /etc/ansible)
  [localhost]
  ansible ansible_host=10.0.49.164 ansible_python_interpreter=/usr/bin/python3
  ansible_user=ec2-user ansible_ssh_private_key_file=/home/ec2-user/yash.pem



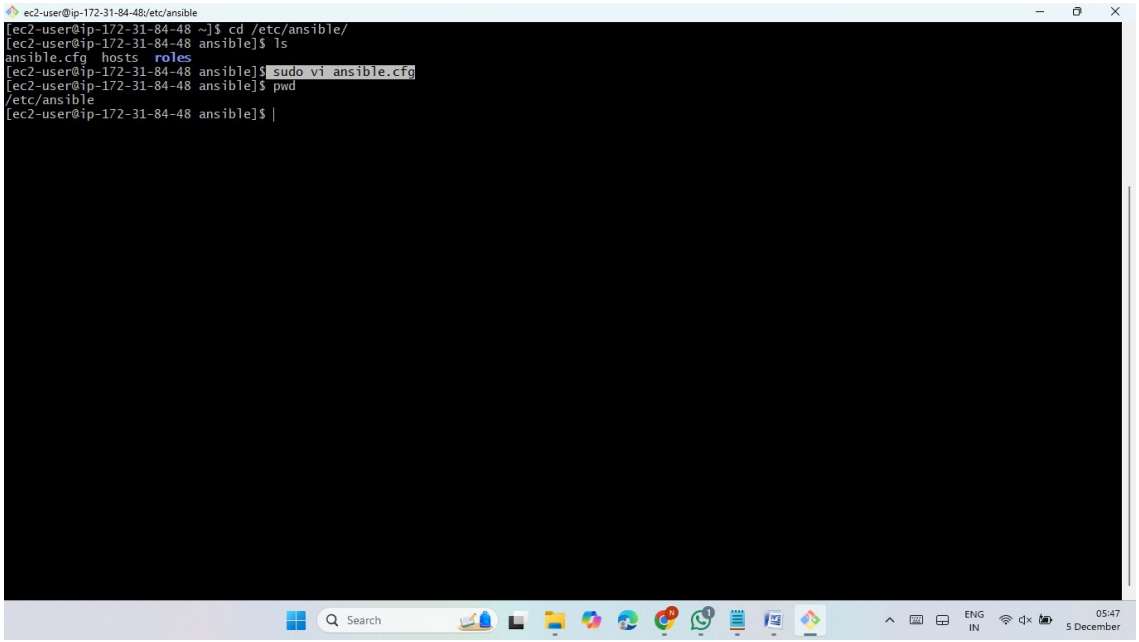


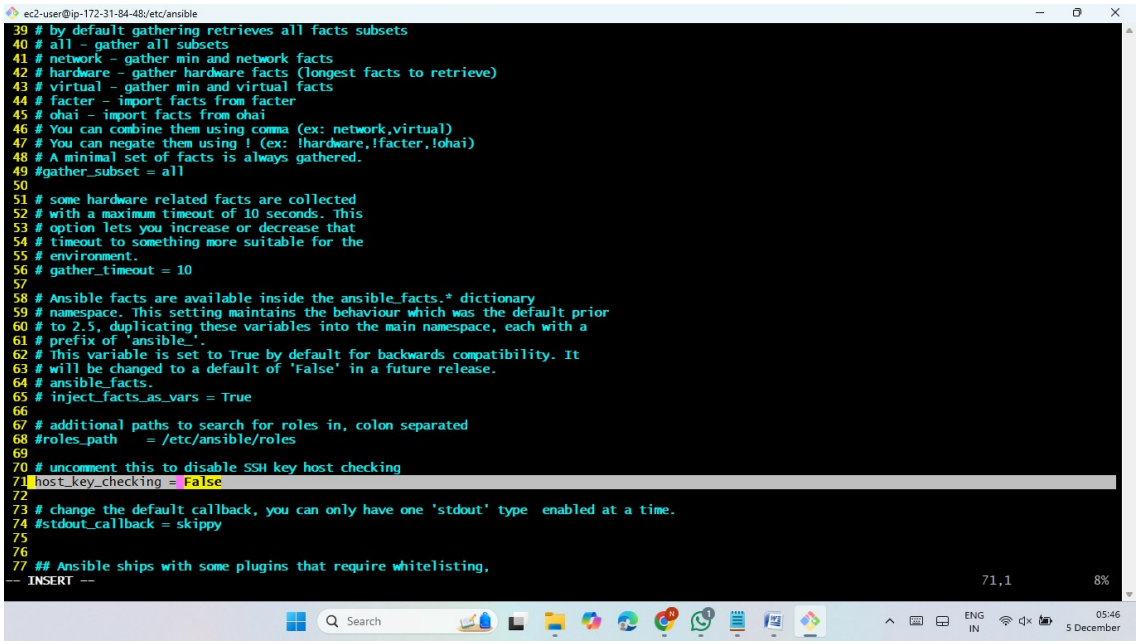
- Now open another terminal then copy the private key form local to instances.



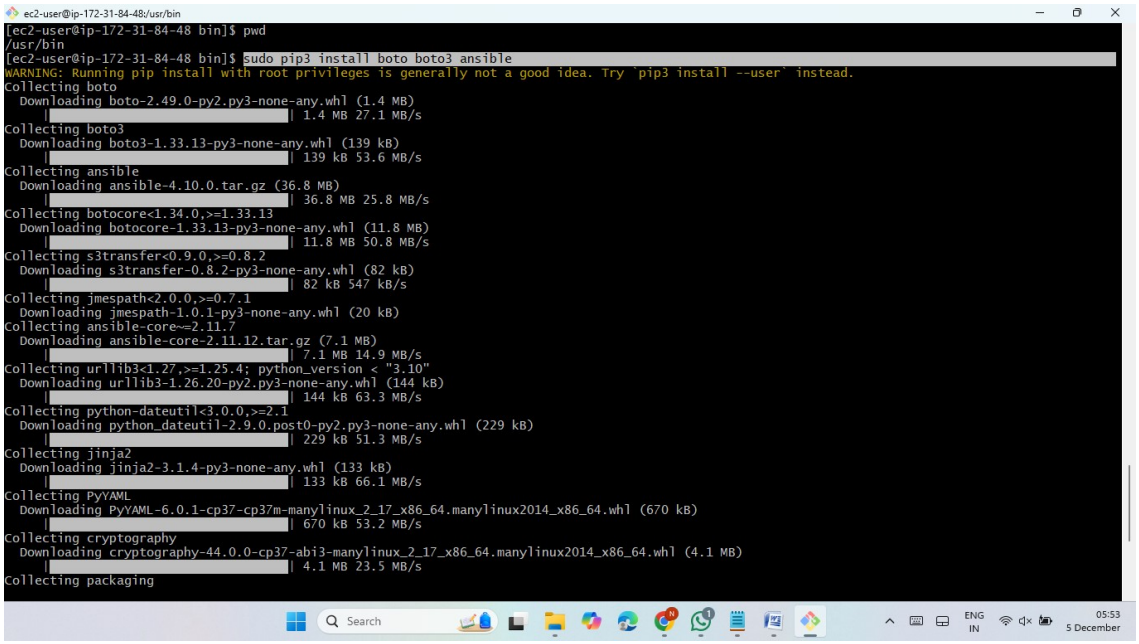


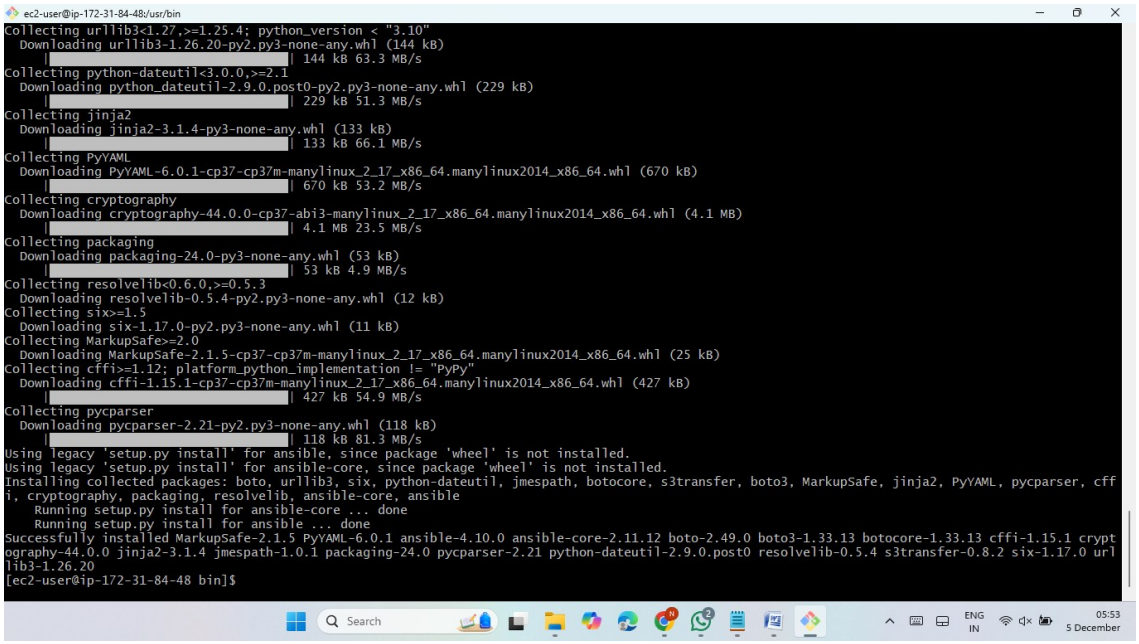
- Here the private key was copied then give permission to the key file.
- Now go to ansible.cfg file (path= /ect/ansible)
- Uncomment the 71 th line in the ansible configuration file.



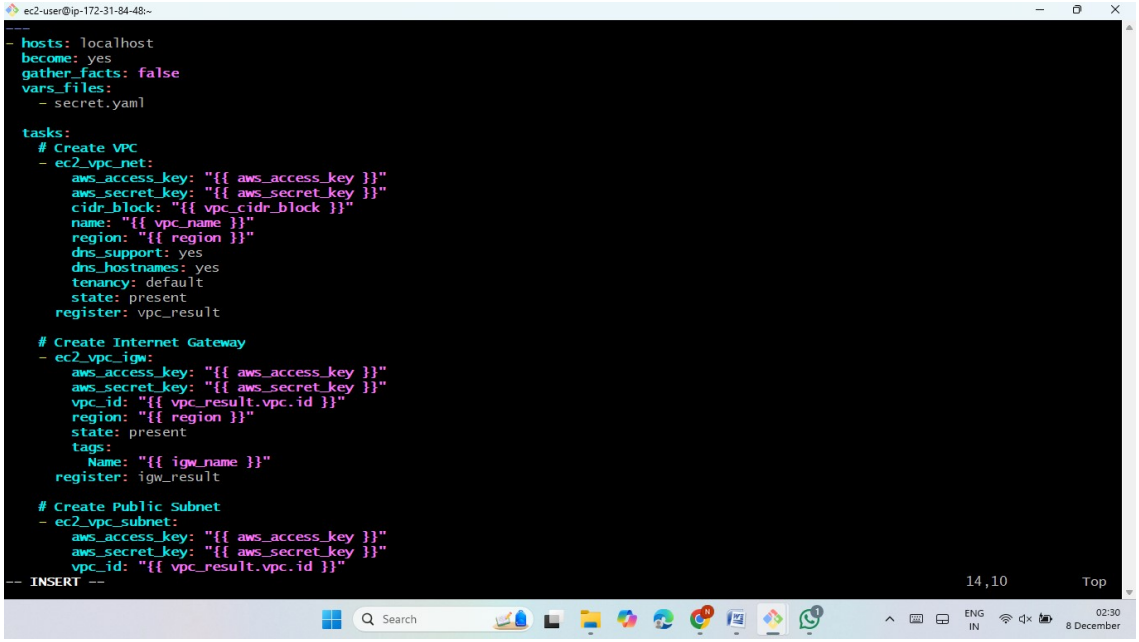


- Now go to /usr/bin/ path then install boto, boto3, ansible with the command.
  Sudo pip3 install boto boto3 ansible.

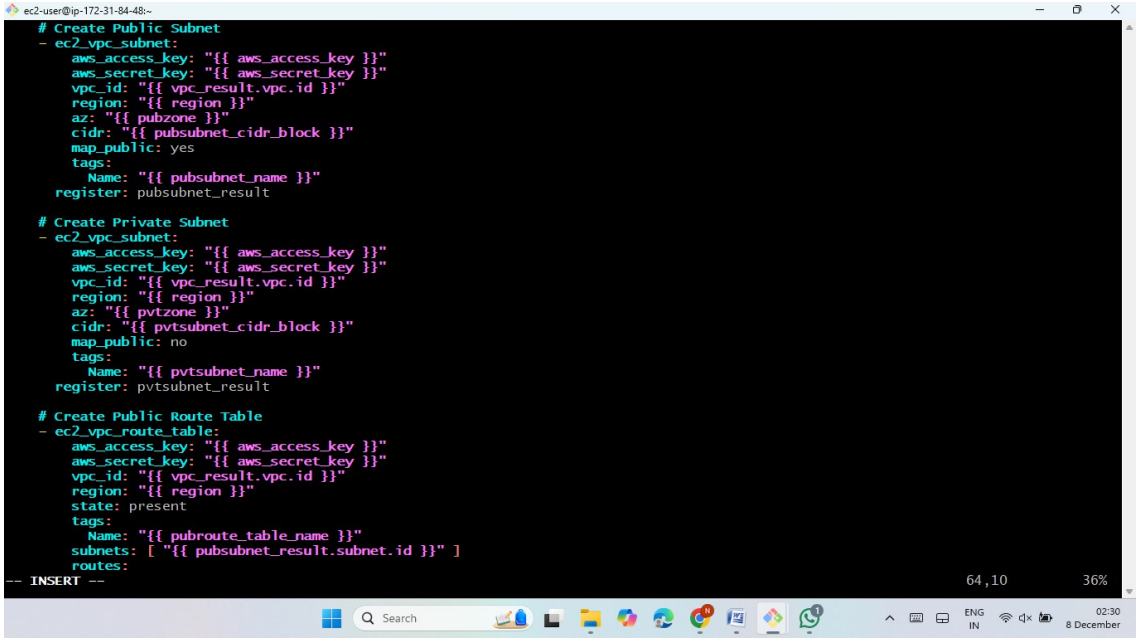




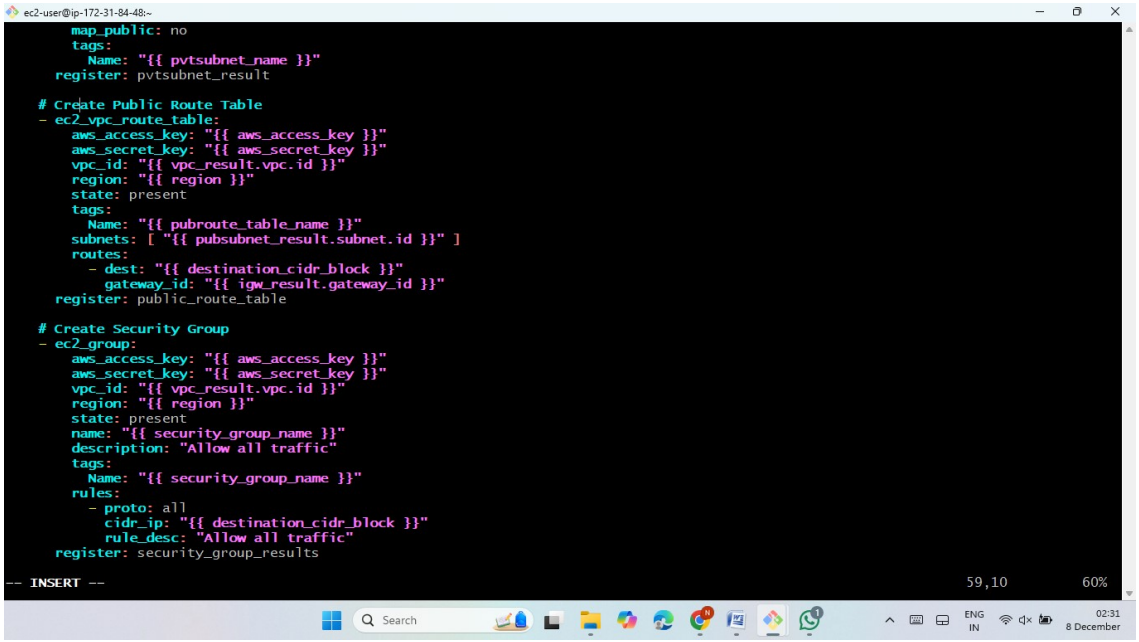
- Now create a file for host the static and dynamic application with the Ansible Script(yaml).
- First create a file for host 2 Instances, VPC, IGW, Private and Public Subnets, Private and Public Route Tables, Security Group.
- Also create variable file(outline).
- Now write a bash script for host static application in instance1.
- Now write a bash script for host static application in instance2.

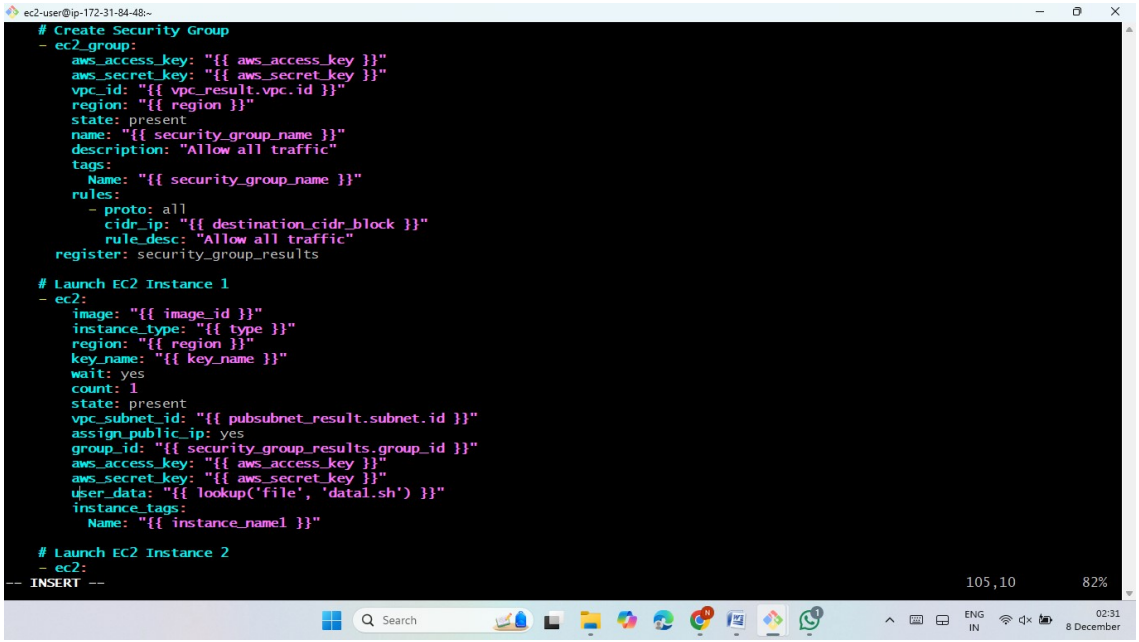


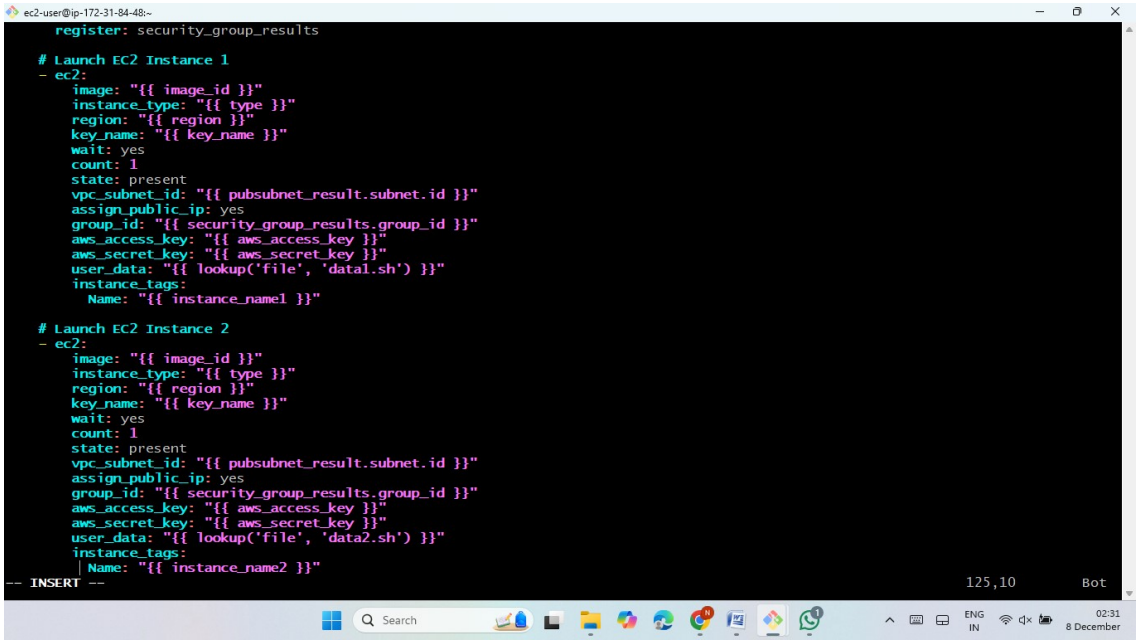
```
ec2-user@ip-172-31-84-48:~
---
- hosts: localhost
  become: yes
  gather_facts: false
  vars_files:
    - secret.yaml

  tasks:
    # Create VPC
    - ec2_vpc_net:
        aws_access_key: "{{ aws_access_key }}"
        aws_secret_key: "{{ aws_secret_key }}"
        cidr_block: "{{ vpc_cidr_block }}"
        name: "{{ vpc_name }}"
        region: "{{ region }}"
        dns_support: yes
        dns_hostnames: yes
        tenancy: default
        state: present
      register: vpc_result

    # Create Internet Gateway
    - ec2_vpc_igw:
        aws_access_key: "{{ aws_access_key }}"
        aws_secret_key: "{{ aws_secret_key }}"
        vpc_id: "{{ vpc_result.vpc.id }}"
        region: "{{ region }}"
        state: present
        tags:
          Name: "{{ igw_name }}"
      register: igw_result

    # Create Public Subnet
    - ec2_vpc_subnet:
        aws_access_key: "{{ aws_access_key }}"
        aws_secret_key: "{{ aws_secret_key }}"
        vpc_id: "{{ vpc_result.vpc.id }}"
-- INSERT --                                                    14,10         Top
```

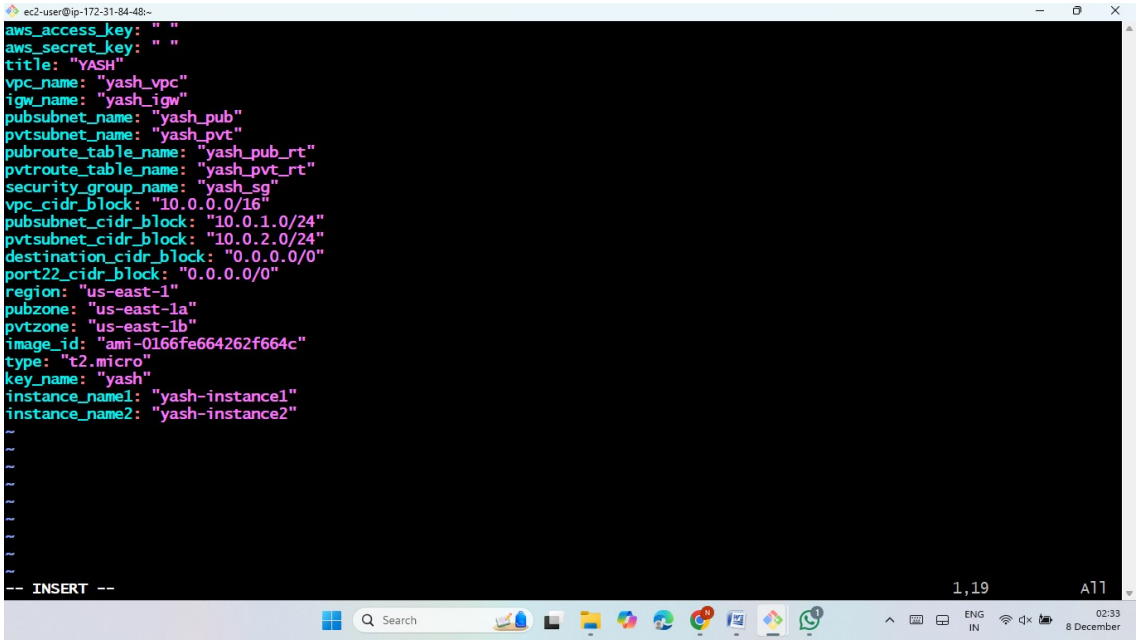


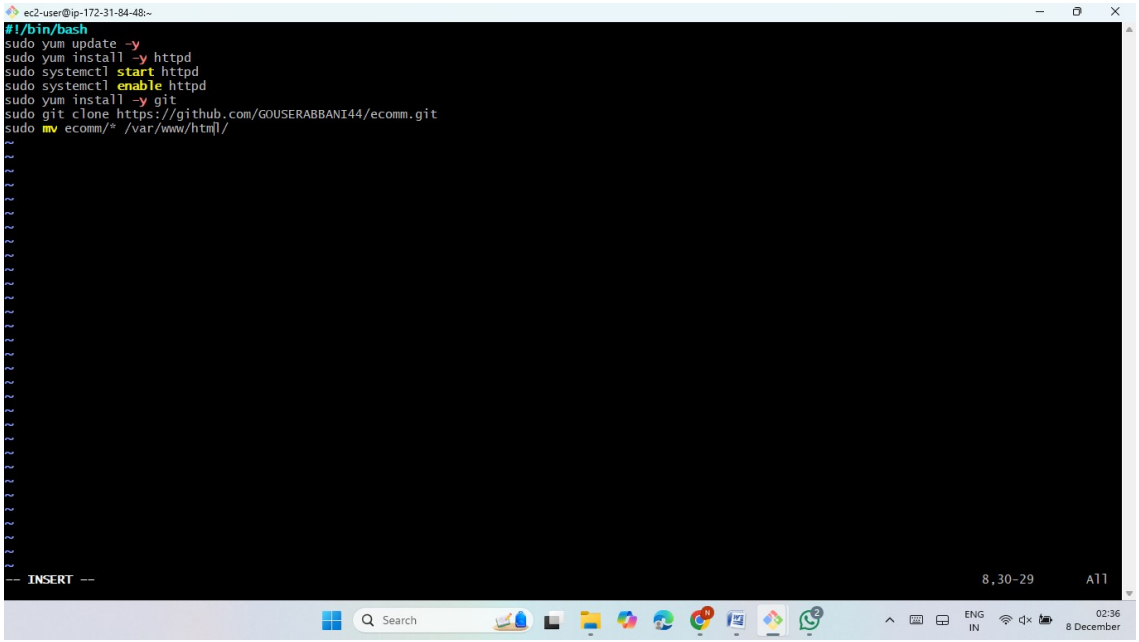
```
ec2-user@ip-172-31-84-48:~
    # Create Public Subnet
    - ec2_vpc_subnet:
        aws_access_key: "{{ aws_access_key }}"
        aws_secret_key: "{{ aws_secret_key }}"
        vpc_id: "{{ vpc_result.vpc.id }}"
        region: "{{ region }}"
        az: "{{ pubzone }}"
        cidr: "{{ pubsubnet_cidr_block }}"
        map_public: yes
        tags:
          Name: "{{ pubsubnet_name }}"
      register: pubsubnet_result

    # Create Private Subnet
    - ec2_vpc_subnet:
        aws_access_key: "{{ aws_access_key }}"
        aws_secret_key: "{{ aws_secret_key }}"
        vpc_id: "{{ vpc_result.vpc.id }}"
        region: "{{ region }}"
        az: "{{ pvtzone }}"
        cidr: "{{ pvtsubnet_cidr_block }}"
        map_public: no
        tags:
          Name: "{{ pvtsubnet_name }}"
      register: pvtsubnet_result

    # Create Public Route Table
    - ec2_vpc_route_table:
        aws_access_key: "{{ aws_access_key }}"
        aws_secret_key: "{{ aws_secret_key }}"
        vpc_id: "{{ vpc_result.vpc.id }}"
        region: "{{ region }}"
        state: present
        tags:
          Name: "{{ pubroute_table_name }}"
        subnets: [ "{{ pubsubnet_result.subnet.id }}" ]
        routes:
-- INSERT --                                                    64,10         36%
```

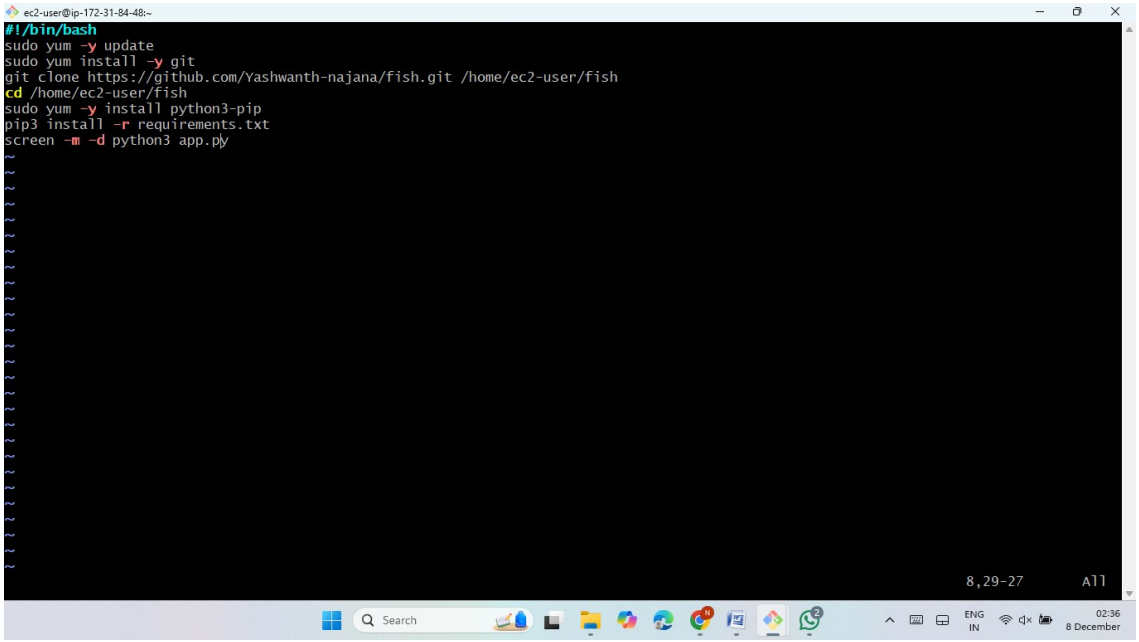

ec2-user@ip-172-31-84-48:~
map_public: no
tags:
Name: "{{ pvtsubnet_name }}"
register: pvtsubnet_result
Create Public Route Table
- ec2_vpc_route_table:
aws_access_key: "{{ aws_access_key }}"
aws_secret_key: "{{ aws_secret_key }}"
vpc_id: "{{ vpc_result.vpc.id }}"
region: "{{ region }}"
state: present
tags:
Name: "{{ pubroute_table_name }}"
subnets: ["{{ pubsubnet_result.subnet.id }}"]
routes:
- dest: "{{ destination_cidr_block }}"
gateway_id: "{{ igw_result.gateway_id }}"
register: public_route_table
Create Security Group
- ec2_group:
aws_access_key: "{{ aws_access_key }}"
aws_secret_key: "{{ aws_secret_key }}"
vpc_id: "{{ vpc_result.vpc.id }}"
region: "{{ region }}"
state: present
name: "{{ security_group_name }}"
description: "Allow all traffic"
tags:
Name: "{{ security_group_name }}"
rules:
- proto: all
cidr_ip: "{{ destination_cidr_block }}"
rule_desc: "Allow all traffic"
register: security_group_results
-- INSERT --
59,10
60%
Search
ENG
IN
02:31
8 December


ec2-user@ip-172-31-84-48:~
Create Security Group
- ec2_group:
aws_access_key: "{{ aws_access_key }}"
aws_secret_key: "{{ aws_secret_key }}"
vpc_id: "{{ vpc_result.vpc.id }}"
region: "{{ region }}"
state: present
name: "{{ security_group_name }}"
description: "Allow all traffic"
tags:
Name: "{{ security_group_name }}"
rules:
- proto: all
cidr_ip: "{{ destination_cidr_block }}"
rule_desc: "Allow all traffic"
register: security_group_results
Launch EC2 Instance 1
- ec2:
image: "{{ image_id }}"
instance_type: "{{ type }}"
region: "{{ region }}"
key_name: "{{ key_name }}"
wait: yes
count: 1
state: present
vpc_subnet_id: "{{ pubsubnet_result.subnet.id }}"
assign_public_ip: yes
group_id: "{{ security_group_results.group_id }}"
aws_access_key: "{{ aws_access_key }}"
aws_secret_key: "{{ aws_secret_key }}"
user_data: "{{ lookup('file', 'data1.sh') }}"
instance_tags:
Name: "{{ instance_name1 }}"
Launch EC2 Instance 2
- ec2:
-- INSERT --
105,10
82%
Search
ENG
IN
02:31
8 December

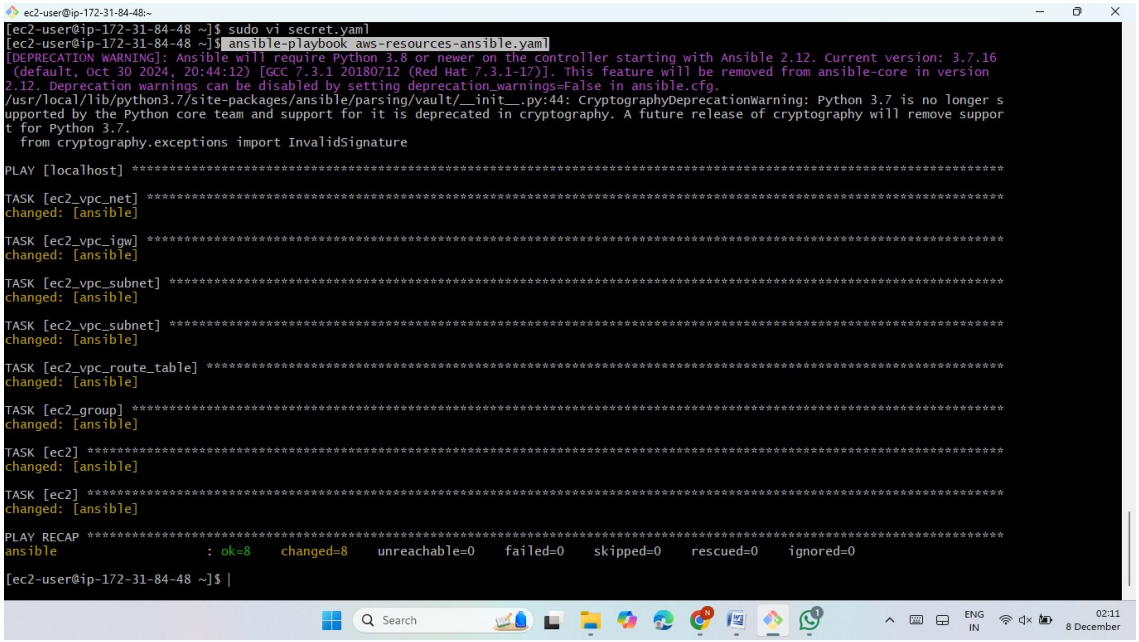

ec2-user@ip-172-31-84-48:~
register: security_group_results
Launch EC2 Instance 1
- ec2:
image: "{{ image_id }}"
instance_type: "{{ type }}"
region: "{{ region }}"
key_name: "{{ key_name }}"
wait: yes
count: 1
state: present
vpc_subnet_id: "{{ pubsubnet_result.subnet.id }}"
assign_public_ip: yes
group_id: "{{ security_group_results.group_id }}"
aws_access_key: "{{ aws_access_key }}"
aws_secret_key: "{{ aws_secret_key }}"
user_data: "{{ lookup('file', 'data1.sh') }}"
instance_tags:
Name: "{{ instance_name1 }}"
Launch EC2 Instance 2
- ec2:
image: "{{ image_id }}"
instance_type: "{{ type }}"
region: "{{ region }}"
key_name: "{{ key_name }}"
wait: yes
count: 1
state: present
vpc_subnet_id: "{{ pubsubnet_result.subnet.id }}"
assign_public_ip: yes
group_id: "{{ security_group_results.group_id }}"
aws_access_key: "{{ aws_access_key }}"
aws_secret_key: "{{ aws_secret_key }}"
user_data: "{{ lookup('file', 'data2.sh') }}"
instance_tags:
Name: "{{ instance_name2 }}"
-- INSERT --
125,10
Bot
Search
ENG
IN
02:31
8 December


ec2-user@ip-172-31-84-48:~
aws_access_key: " "
aws_secret_key: " "
title: "YASH"
vpc_name: "yash_vpc"
igw_name: "yash_igw"
pubsubnet_name: "yash_pub"
pvtsubnet_name: "yash_pvt"
pubroute_table_name: "yash_pub_rt"
pvtroute_table_name: "yash_pvt_rt"
security_group_name: "yash_sg"
vpc_cidr_block: "10.0.0.0/16"
pubsubnet_cidr_block: "10.0.1.0/24"
pvtsubnet_cidr_block: "10.0.2.0/24"
destination_cidr_block: "0.0.0.0/0"
port22_cidr_block: "0.0.0.0/0"
region: "us-east-1"
pubzone: "us-east-1a"
pvtzone: "us-east-1b"
image_id: "ami-0166fe664262f664c"
type: "t2.micro"
key_name: "yash"
instance_name1: "yash-instance1"
instance_name2: "yash-instance2"
-- INSERT --
1,19
All
Search
ENG
IN
02:33
8 December

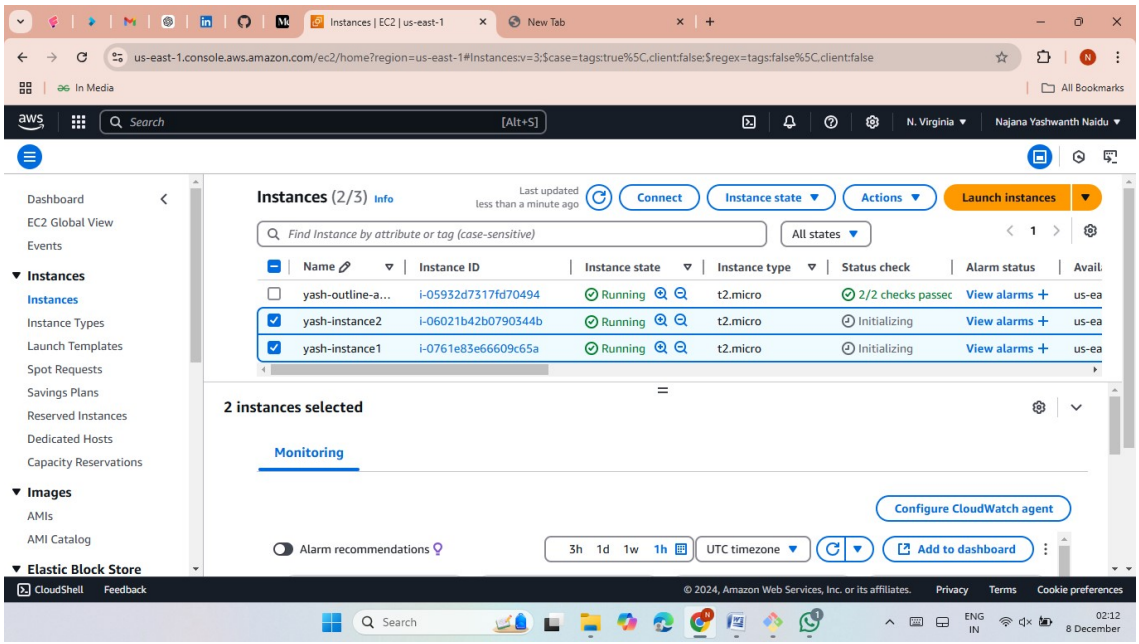

ec2-user@ip-172-31-84-48:~
#!/bin/bash
sudo yum update -y
sudo yum install -y httpd
sudo systemctl start httpd
sudo systemctl enable httpd
sudo yum install -y git
sudo git clone https://github.com/GOUSERABBANI44/ecomm.git
sudo mv ecomm/* /var/www/html/
-- INSERT --
8,30-29
All
Search
ENG
IN
02:36
8 December


ec2-user@ip-172-31-84-48:~
#!/bin/bash
sudo yum -y update
sudo yum install -y git
git clone https://github.com/Yashwanth-najana/fish.git /home/ec2-user/fish
cd /home/ec2-user/fish
sudo yum -y install python3-pip
pip3 install -r requirements.txt
screen -m -d python3 app.py
8,29-27
All
Search
ENG
IN
02:36
8 December

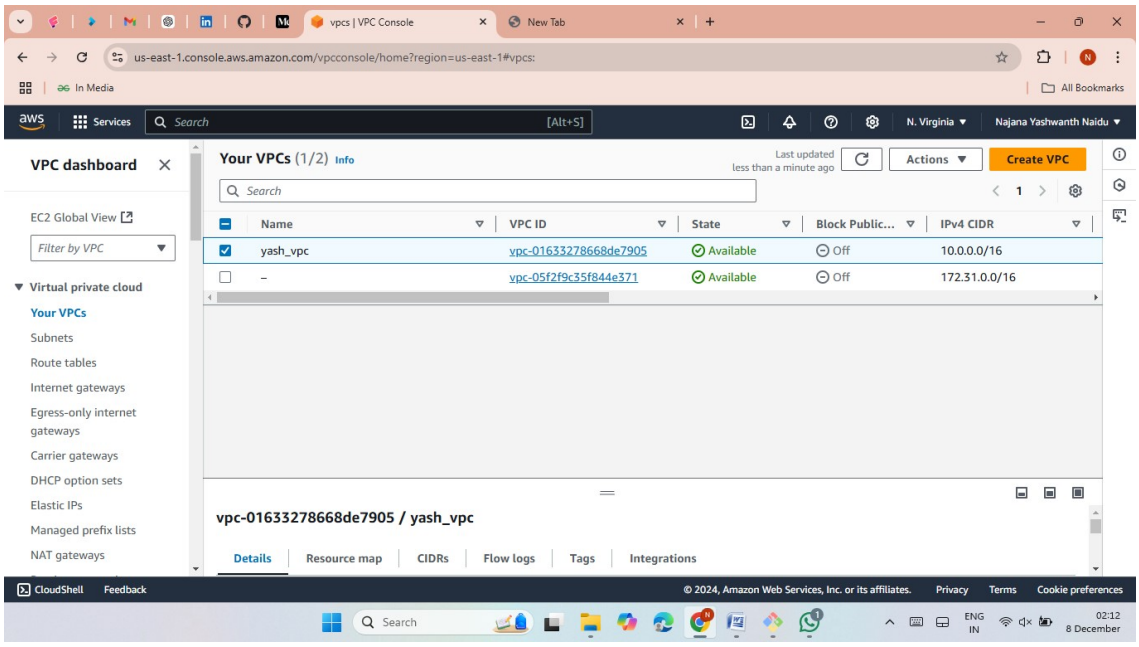
- Now run the AWS resources with the command.
  Ansible-playbook <name of file>
- Here the resources was created successfully.



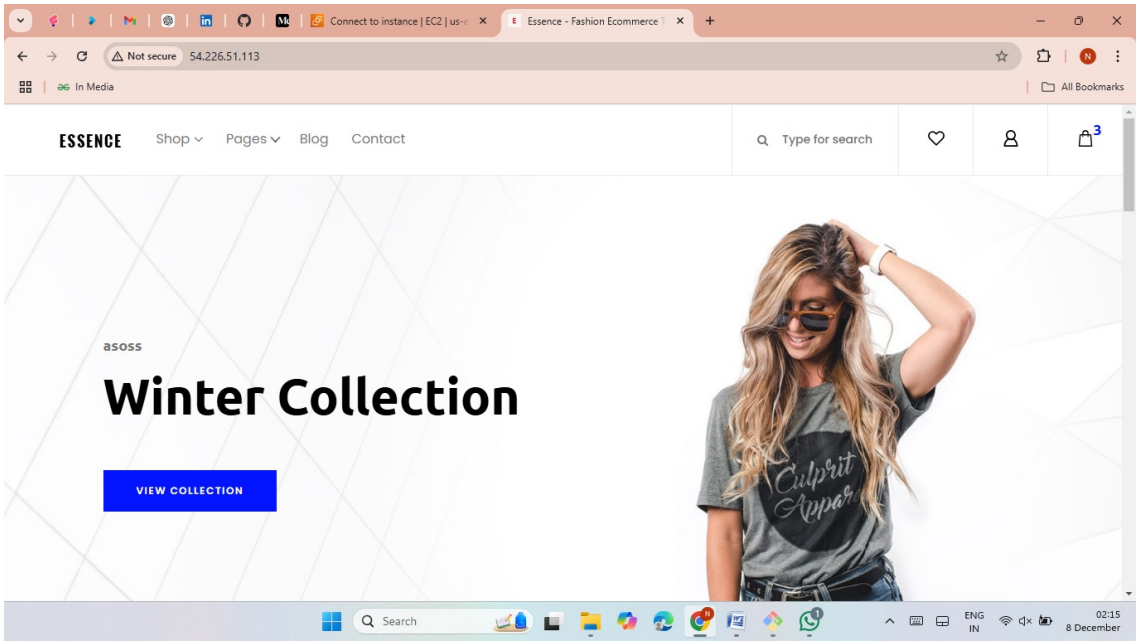
- Here the 2 instances was launched.
- Here the VPC, IGW, Subnets, Route Tables and Security group was created.



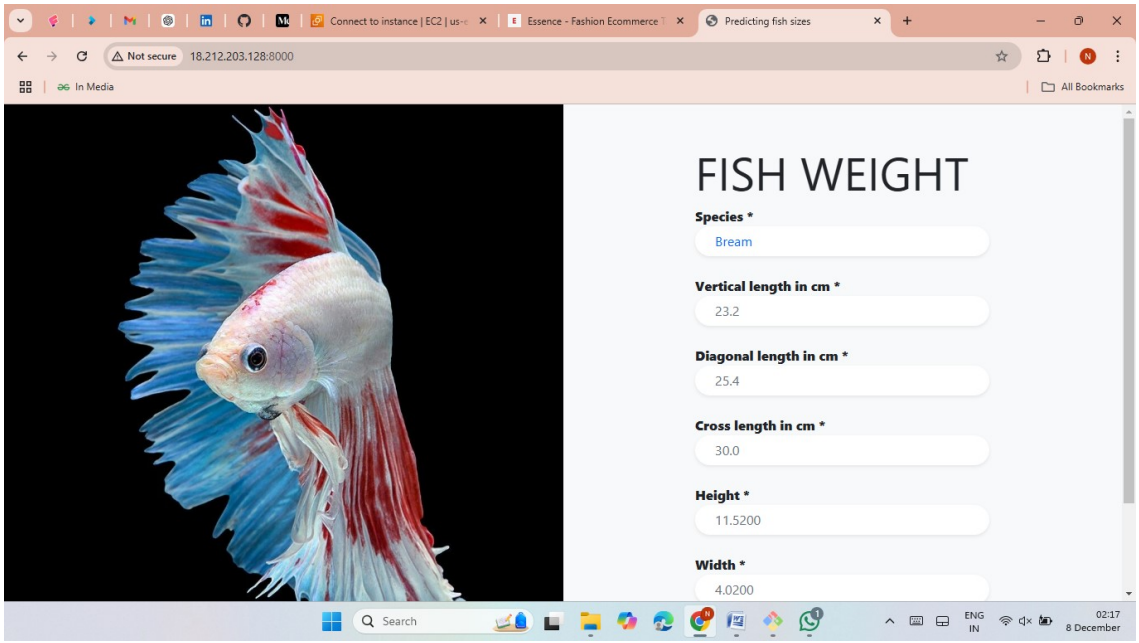
- Here the VPC was created.



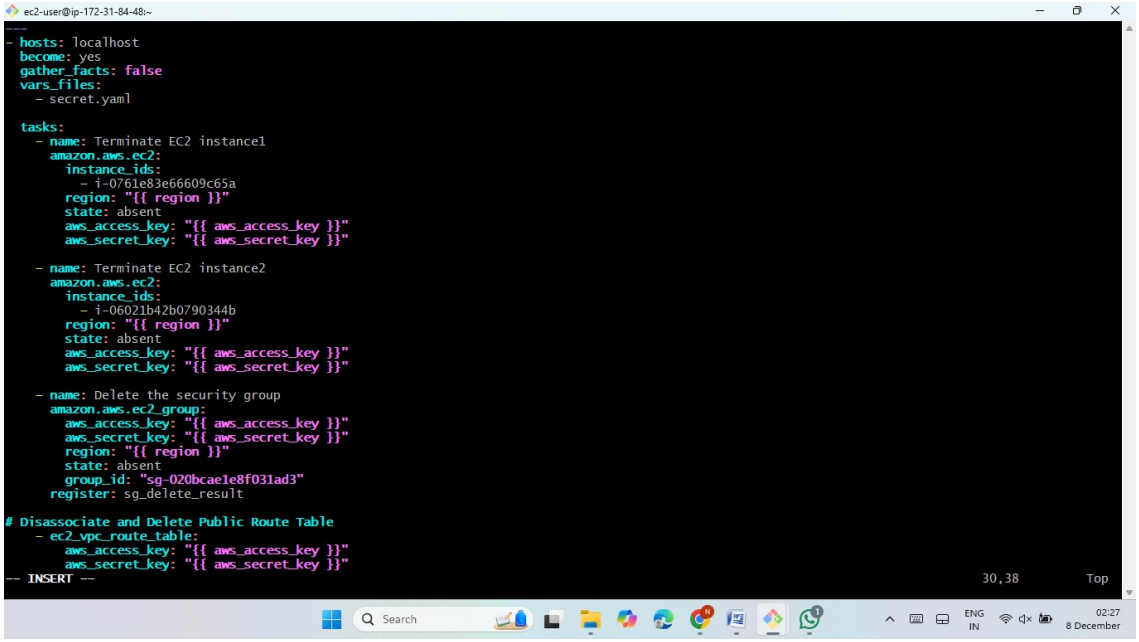
- Now copy the Instance 1 Public IP then Past it on Google along with port 80.
- The static ecomm application was hosted successfully.

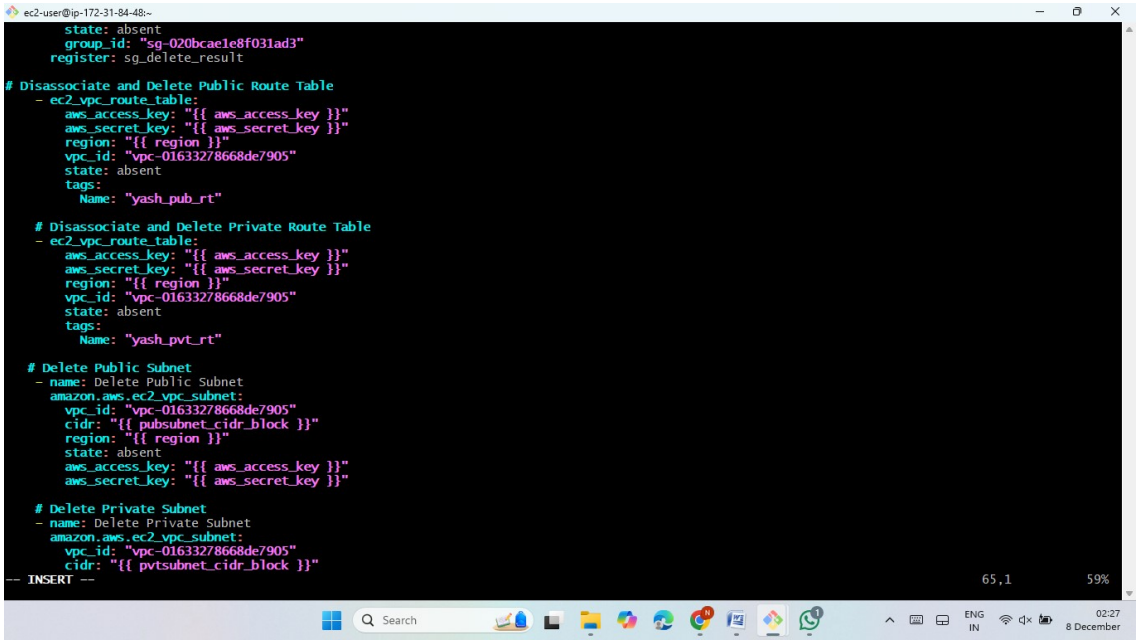


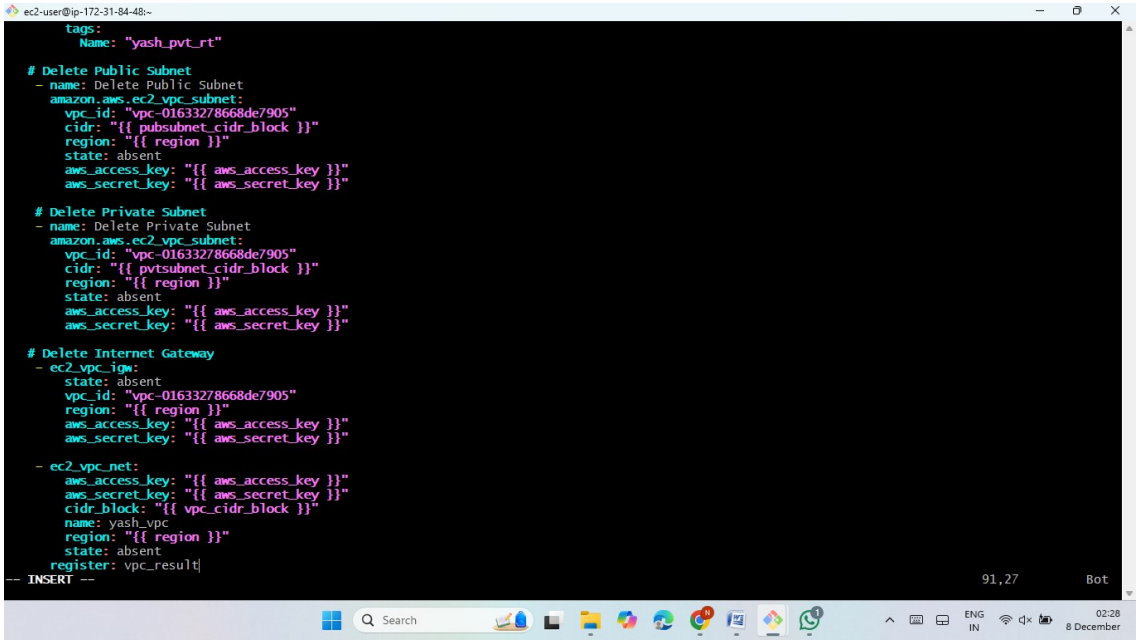
- Now copy the Instance2 Public IP then Past it on Google along with port 8000.
- The dynamic python application was hosted successfully.



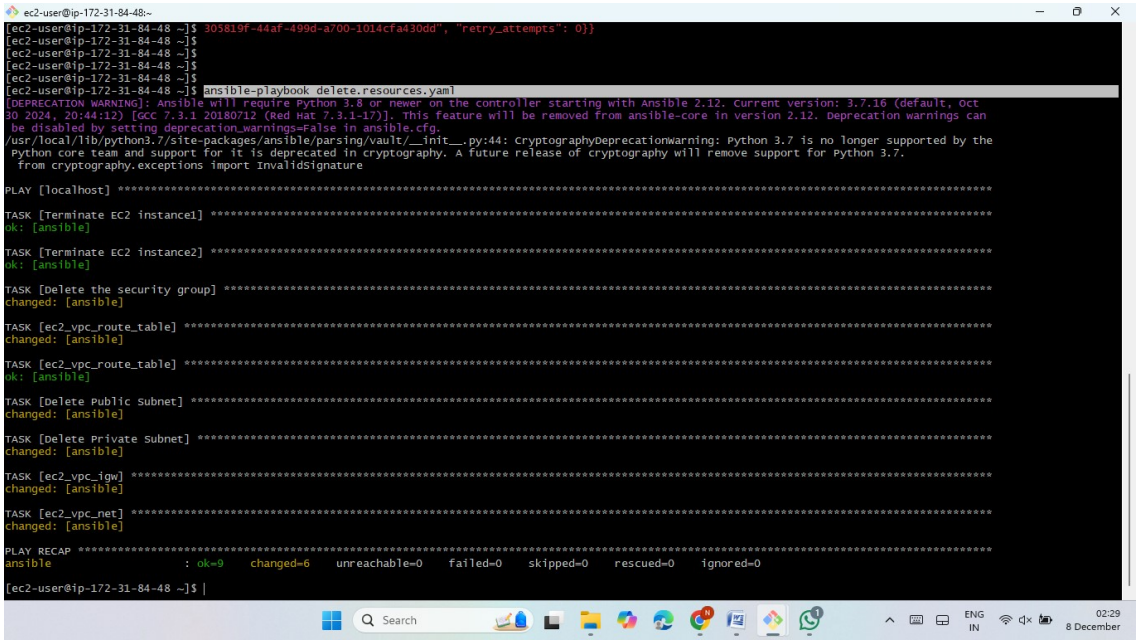
- Now create file for delete the created resources.



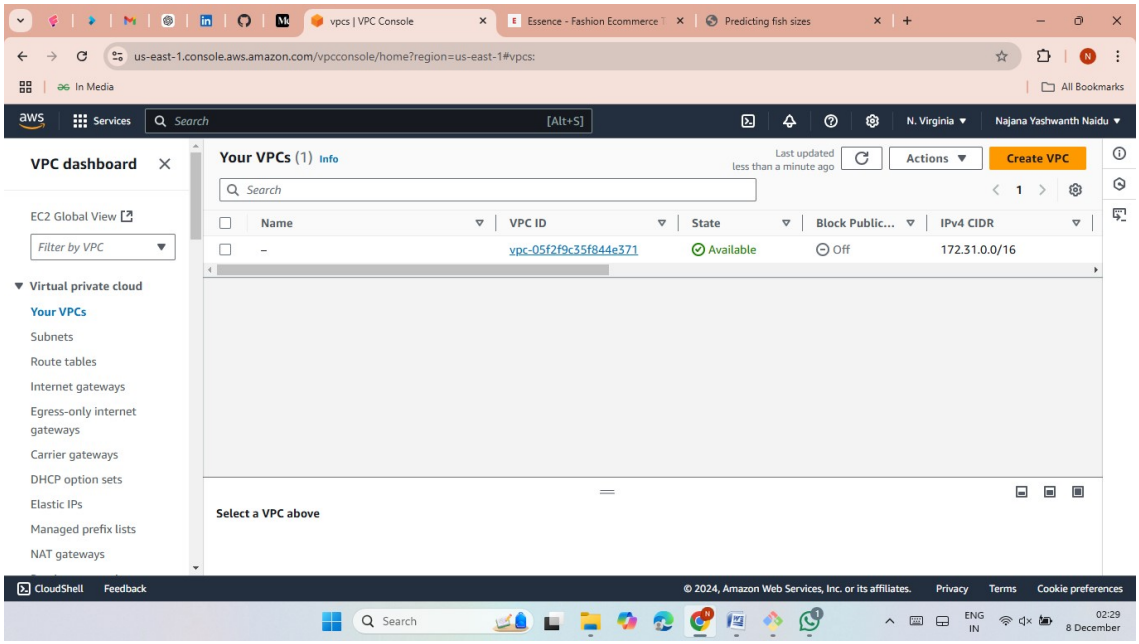

ec2-user@ip-172-31-84-48:~
state: absent
group_id: "sg-020bcae1e8f031ad3"
register: sg_delete_result
Disassociate and Delete Public Route Table
- ec2_vpc_route_table:
aws_access_key: "{{ aws_access_key }}"
aws_secret_key: "{{ aws_secret_key }}"
region: "{{ region }}"
vpc_id: "vpc-01633278668de7905"
state: absent
tags:
Name: "yash_pub_rt"
Disassociate and Delete Private Route Table
- ec2_vpc_route_table:
aws_access_key: "{{ aws_access_key }}"
aws_secret_key: "{{ aws_secret_key }}"
region: "{{ region }}"
vpc_id: "vpc-01633278668de7905"
state: absent
tags:
Name: "yash_pvt_rt"
Delete Public Subnet
- name: Delete Public Subnet
amazon.aws.ec2_vpc_subnet:
vpc_id: "vpc-01633278668de7905"
cidr: "{{ pubsubnet_cidr_block }}"
region: "{{ region }}"
state: absent
aws_access_key: "{{ aws_access_key }}"
aws_secret_key: "{{ aws_secret_key }}"
Delete Private Subnet
- name: Delete Private Subnet
amazon.aws.ec2_vpc_subnet:
vpc_id: "vpc-01633278668de7905"
cidr: "{{ pvtsubnet_cidr_block }}"
-- INSERT --
65,1
59%
Search
ENG
IN
02:27
8 December

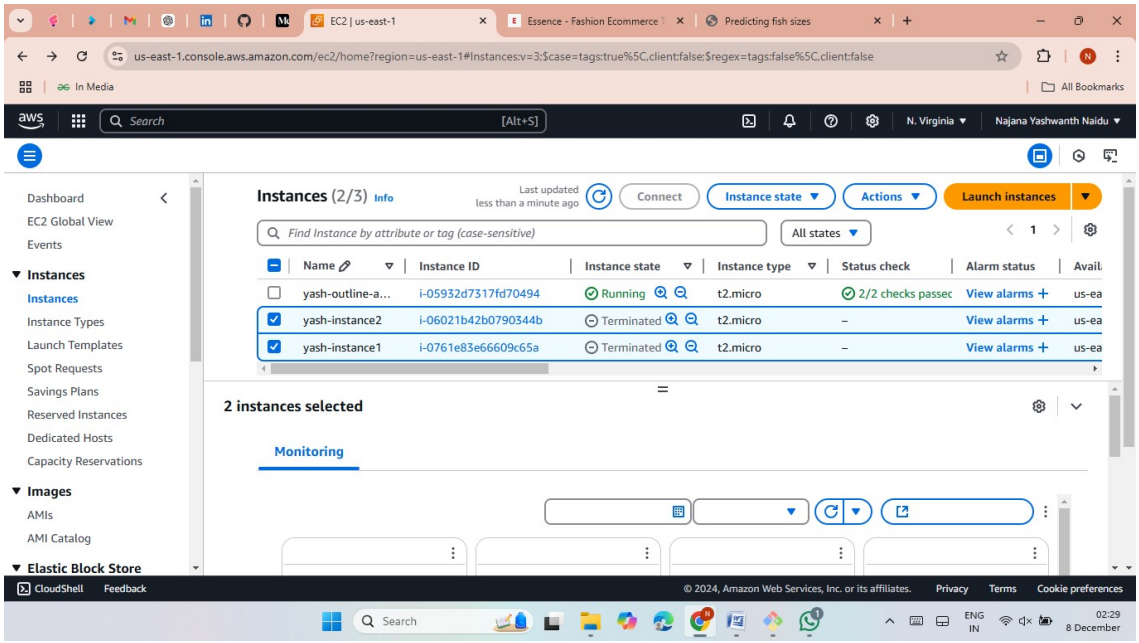

ec2-user@ip-172-31-84-48:~
tags:
Name: "yash_pvt_rt"
Delete Public Subnet
- name: Delete Public Subnet
amazon.aws.ec2_vpc_subnet:
vpc_id: "vpc-01633278668de7905"
cidr: "{{ pubsubnet_cidr_block }}"
region: "{{ region }}"
state: absent
aws_access_key: "{{ aws_access_key }}"
aws_secret_key: "{{ aws_secret_key }}"
Delete Private Subnet
- name: Delete Private Subnet
amazon.aws.ec2_vpc_subnet:
vpc_id: "vpc-01633278668de7905"
cidr: "{{ pvtsubnet_cidr_block }}"
region: "{{ region }}"
state: absent
aws_access_key: "{{ aws_access_key }}"
aws_secret_key: "{{ aws_secret_key }}"
Delete Internet Gateway
- ec2_vpc_igw:
state: absent
vpc_id: "vpc-01633278668de7905"
region: "{{ region }}"
aws_access_key: "{{ aws_access_key }}"
aws_secret_key: "{{ aws_secret_key }}"
- ec2_vpc_net:
aws_access_key: "{{ aws_access_key }}"
aws_secret_key: "{{ aws_secret_key }}"
cidr_block: "{{ vpc_cidr_block }}"
name: yash_vpc
region: "{{ region }}"
state: absent
register: vpc_result
-- INSERT --
91,27
Bot
Search
ENG
IN
02:28
8 December

- Now run the file with the command **ansible-playbook <name of the file>.**



- Here the created resources was deleted successfully.
- Here the instances and VPC was deleted.




EC2 | us-east-1
Essence - Fashion Ecommerce
Predicting fish sizes
us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#Instances:v=3;$case=tags:true%5C,client:false;$regex=tags:false%5C,client:false
aws
Search
[Alt+S]
N. Virginia
Najana Yashwanth Naidu
Dashboard
EC2 Global View
Events
Instances
Instances
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances
Dedicated Hosts
Capacity Reservations
Images
AMIs
AMI Catalog
Elastic Block Store
Instances (2/3) Info
Last updated less than a minute ago
Connect
Instance state
Actions
Launch instances
Find Instance by attribute or tag (case-sensitive)
All states
Name
Instance ID
Instance state
Instance type
Status check
Alarm status
yash-outline-a...
i-05932d7317fd70494
Running
t2.micro
2/2 checks passed
View alarms
yash-instance2
i-06021b42b0790344b
Terminated
t2.micro
View alarms
yash-instance1
i-0761e83e66609c65a
Terminated
t2.micro
View alarms
2 instances selected
Monitoring
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
Search
ENG
IN
02:29
8 December