

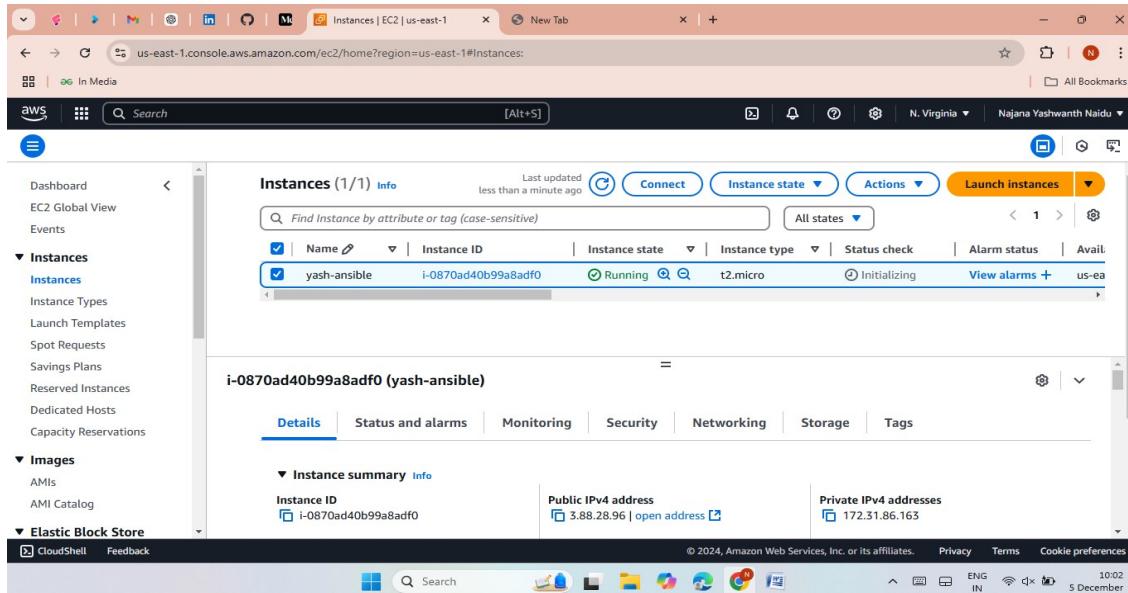
PROJECT

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

```
ec2-user@ip-172-31-86-163:~$ This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-3-88-28-96.compute-1.amazonaws.com' (ED25519) to the list of known hosts.

# Amazon Linux 2
AL2 End of Life is 2025-06-30.
A newer version of Amazon Linux is available!
Amazon Linux 2023, GA and supported until 2028-03-15.
https://aws.amazon.com/linux/amazon-linux-2023/

[ec2-user@ip-172-31-86-163 ~]$ sudo yum -y update
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
No packages marked for update
[ec2-user@ip-172-31-86-163 ~]$ sudo amazon-linux-extras install ansible2
Topic ansible2 has end-of-support date of 2023-09-30
Installing ansible
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Cleaning repos: amzn2-core amzn2extra-ansible2 amzn2extra-docker amzn2extra-kernel-5.10
17 metadata files removed
6 sqlite files removed
0 metadata files removed
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core                                         3.6 kB 00:00:00
amzn2extra-ansible2                                2.9 kB 00:00:00
amzn2extra-docker                                  2.9 kB 00:00:00
amzn2extra-kernel-5.10                             3.0 kB 00:00:00
(1/9): amzn2-core/2/x86_64/group.gz                2.7 kB 00:00:00
(2/9): amzn2-core/2/x86_64/updateinfo              1.0 MB 00:00:00
(3/9): amzn2extra-docker/2/x86_64/primary_db       116 kB 00:00:00
(4/9): amzn2extra-kernel-5.10/2/x86_64/updateinfo   93 kB 00:00:00
(5/9): amzn2extra-ansible2/2/x86_64/updateinfo      4.7 kB 00:00:00
(6/9): amzn2extra-ansible2/2/x86_64/primary_db      37 kB 00:00:00
```

```

ec2-user@ip-172-31-86-163:~-
38 nginx1      available  [ =stable ]
40 mock        available  [ =stable ]
43 livepatch   available  [ =stable ]
45 haproxy2    available  [ =stable ]
46 collectd   available  [ =stable ]
47 aws-nitro-enclaves-cli available  [ =stable ]
48 R4          available  [ =stable ]
_ kernel-5.4   available  [ =stable ]
50 selinux-ng  available  [ =stable ]
52 tomcat9    available  [ =stable ]
53 unbound1.13 available  [ =stable ]
54 mariadb10.5 available  [ =stable ]
55 kernel-5.10=latest enabled   [ =stable ]
56 redis6     available  [ =stable ]
59 postgresql13 available  [ =stable ]
60 mock2      available  [ =stable ]
61 dnsmasq2.85 available  [ =stable ]
62 kernel-5.15 available  [ =stable ]
63 postgresql14 available  [ =stable ]
64 firefox     available  [ =stable ]
65 lustre      available  [ =stable ]
67 awscii1    available  [ =stable ]
68 fphp8.2     available  [ =stable ]
69 dnsmasq    available  [ =stable ]
70 unbound1.17 available  [ =stable ]
72 collectd-python3 available  [ =stable ]
* Extra topic has reached end of support.
† Note on end-of-support. Use 'info' subcommand.
[ec2-user@ip-172-31-86-163 ~]$ ansible --version
ansible 2.9.23
  config file = /etc/ansible/ansible.cfg
  configured module search path = ['/home/ec2-user/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python2.7/site-packages/ansible
  executable location = /usr/bin/ansible
  python version = 2.7.18 (default, Oct 17 2024, 18:59:07) [GCC 7.3.1 20180712 (Red Hat 7.3.1-17)]
[ec2-user@ip-172-31-86-163 ~]$
```

➤ 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
```

```

ec2-user@ip-172-31-86-163:/etc/ansible
[localhost]
ansible ansible_host=172.31.86.163 ansible_python_interpreter=/usr/bin/python3 ansible_user=ec2-user ansible_ssh_private_key_file=/home/ec2-user/yash.pem
# This is the default ansible 'hosts' file.
#
# It should live in /etc/ansible/hosts
#
# - Comments begin with the '#' character
# - Blank lines are ignored
# - Groups of hosts are delimited by [header] elements
# - You can enter hostnames or ip addresses
# - A hostname/ip can be a member of multiple groups

# Ex 1: Ungrouped hosts, specify before any group headers.

## green.example.com
## blue.example.com
## 192.168.100.1
## 192.168.100.10

# Ex 2: A collection of hosts belonging to the 'webservers' group

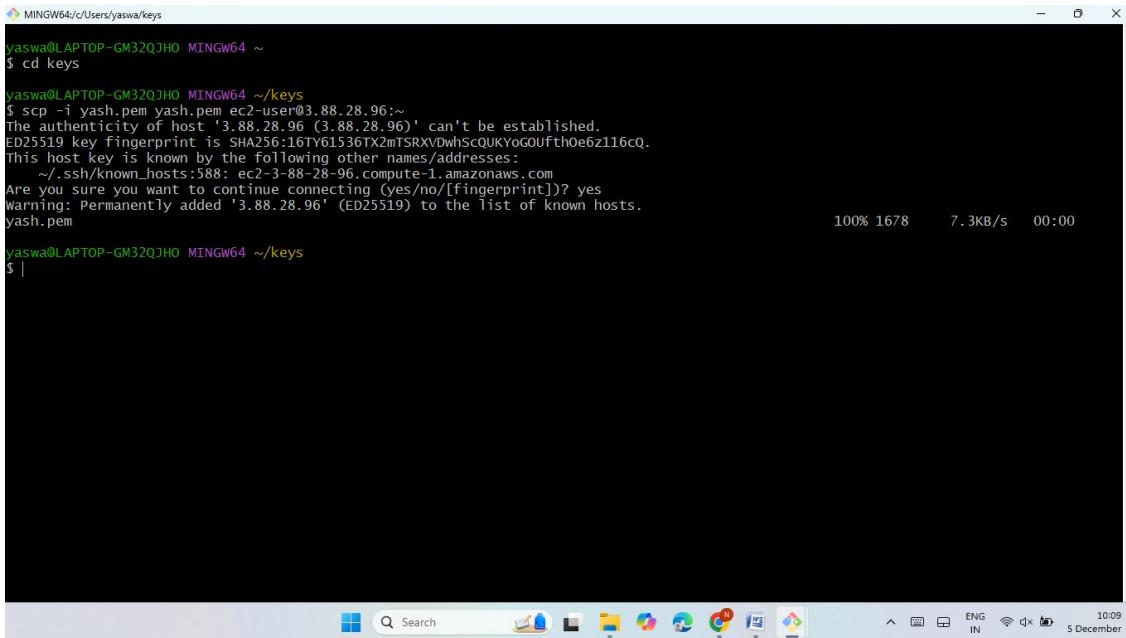
## [webservers]
## alpha.example.org
## beta.example.org
## 192.168.1.100
## 192.168.1.110

# If you have multiple hosts following a pattern you can specify
# them like this:

## www[001:006].example.com

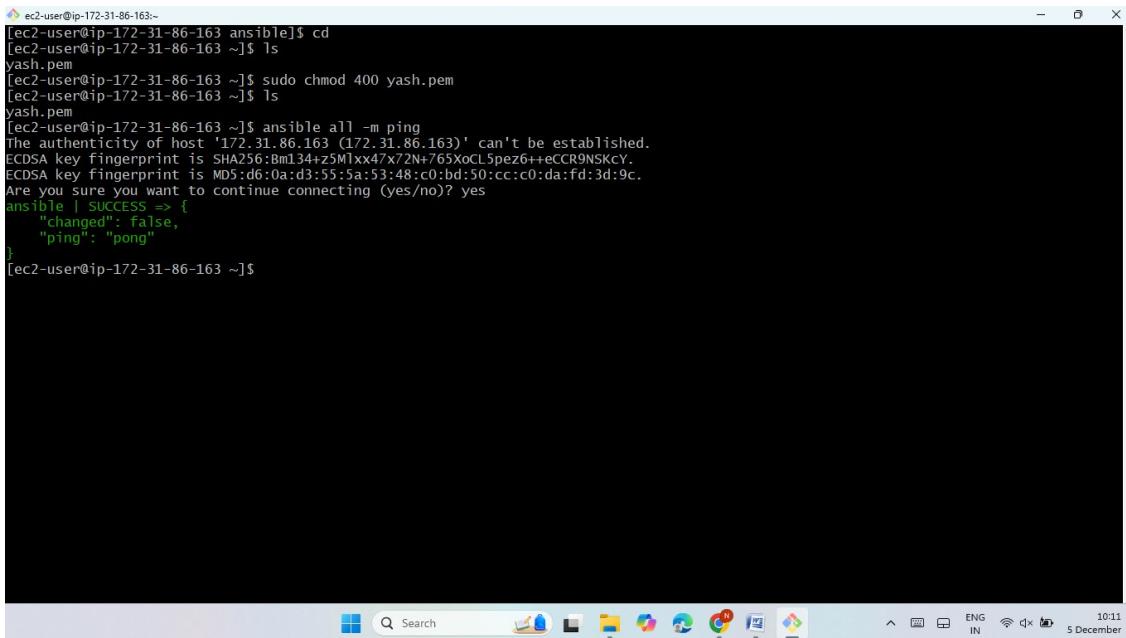
# Ex 3: A collection of database servers in the 'dbservers' group
-- INSERT --
```

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



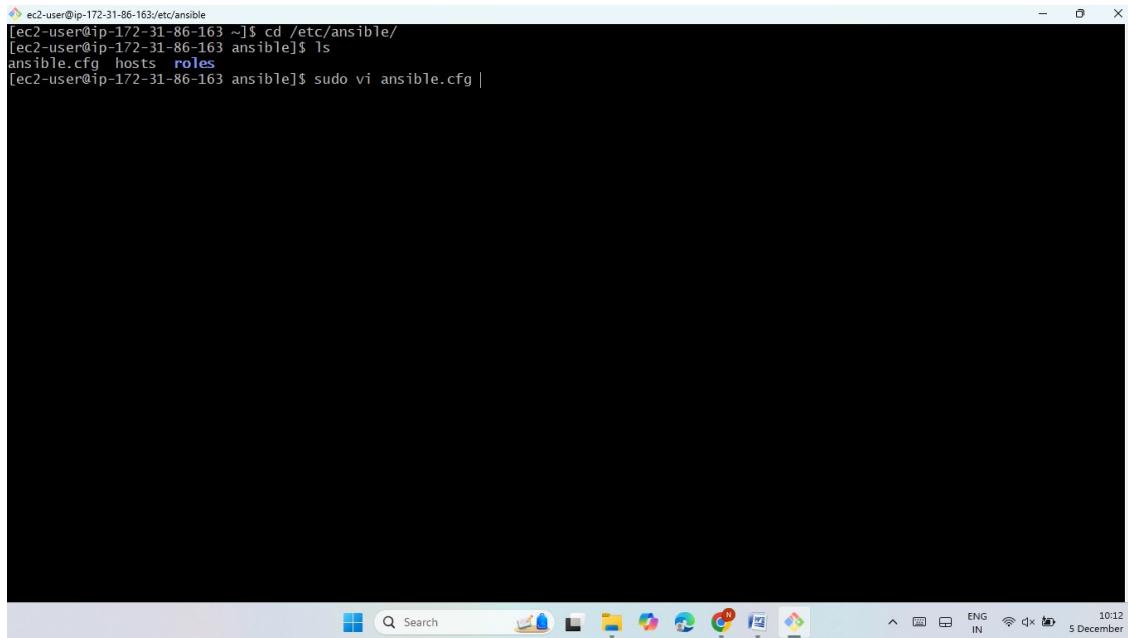
```
MINGW64:/c/Users/yaswa/keys
yaswa@LAPTOP-GM32QJHO MINGW64 ~
$ cd keys
yaswa@LAPTOP-GM32QJHO MINGW64 ~/keys
$ scp -i yash.pem ec2-user@3.88.28.96:~
The authenticity of host '3.88.28.96 (3.88.28.96)' can't be established.
ED25519 key fingerprint is SHA256:16TY61536TX2mTSRXVDwScOUKYoOUpfh0e6z116cQ.
This host key is known by the following other names/addresses:
    ~/.ssh/known_hosts:588: ec2-3-88-28-96.compute-1.amazonaws.com
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '3.88.28.96' (ED25519) to the list of known hosts.
yash.pem                                         100% 1678     7.3KB/s   00:00
yaswa@LAPTOP-GM32QJHO MINGW64 ~/keys
$ |
```

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



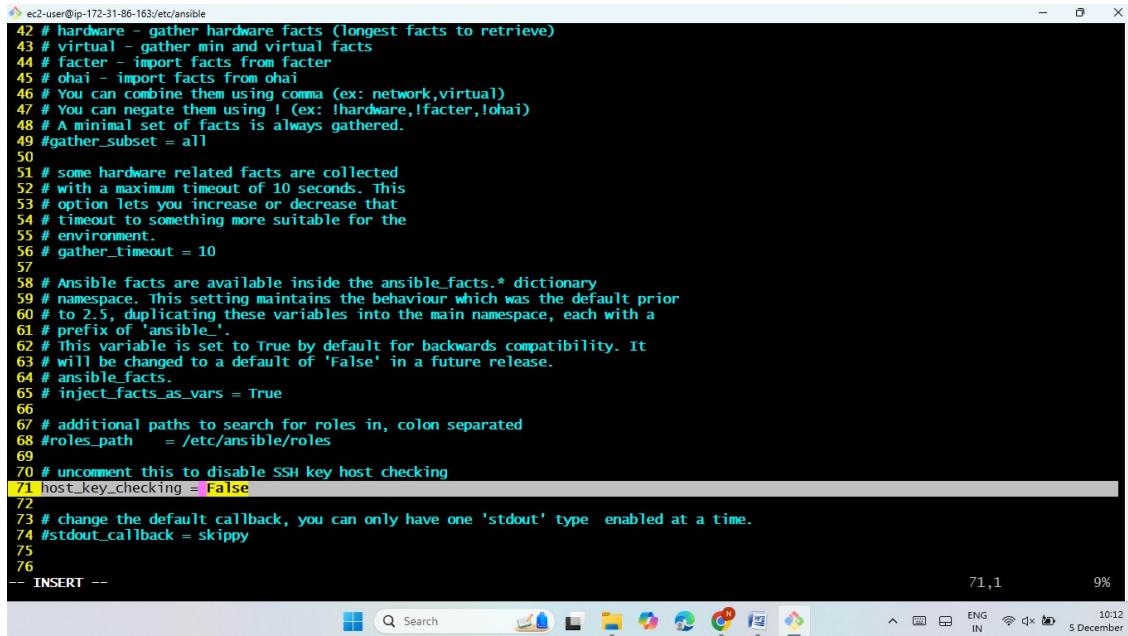
```
[ec2-user@ip-172-31-86-163:~]
[ec2-user@ip-172-31-86-163 ansible]$ cd
[ec2-user@ip-172-31-86-163 ~]$ ls
yash.pem
[ec2-user@ip-172-31-86-163 ~]$ sudo chmod 400 yash.pem
[ec2-user@ip-172-31-86-163 ~]$ ls
yash.pem
[ec2-user@ip-172-31-86-163 ~]$ ansible all -m ping
The authenticity of host '172.31.86.163 (172.31.86.163)' can't be established.
ED25519 key fingerprint is SHA256:8m134+z5M1xx47x72N+765xCL5peZ6++eCCR9NSKcy.
EDDSA key fingerprint is MD5:d6:0a:d3:55:5a:53:48:c0:bd:50:cc:c0:da:fd:3d:9c.
Are you sure you want to continue connecting (yes/no)? yes
ansible | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
[ec2-user@ip-172-31-86-163 ~]$
```

➤ Now go to ansible.cfg file (path= /etc/ansible)



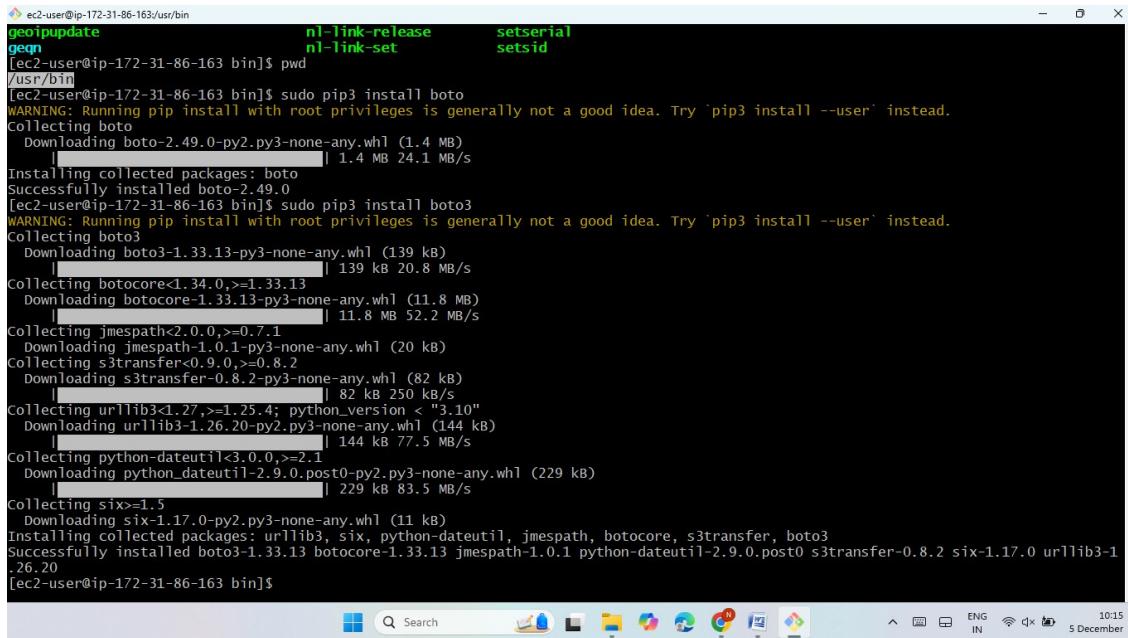
```
ec2-user@ip-172-31-86-163:~$ cd /etc/ansible/
[ec2-user@ip-172-31-86-163 ansible]$ ls
ansible.cfg  hosts  roles
[ec2-user@ip-172-31-86-163 ansible]$ sudo vi ansible.cfg |
```

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

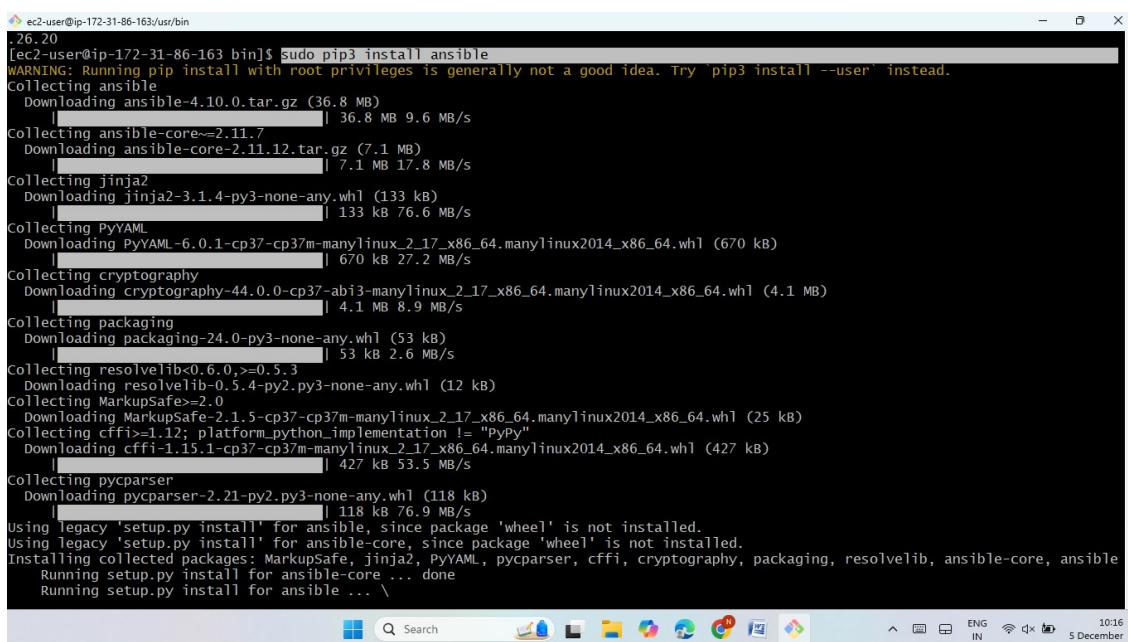


```
42 # hardware - gather hardware facts (longest facts to retrieve)
43 # virtual - gather min and virtual facts
44 # facter - import facts from facter
45 # ohai - import facts from ohai
46 # You can combine them using comma (ex: network,virtual)
47 # You can negate them using ! (ex: !hardware,!facter,!ohai)
48 # A minimal set of facts is always gathered.
49 #gather_subset = all
50
51 # some hardware related facts are collected
52 # with a maximum timeout of 10 seconds. This
53 # option lets you increase or decrease that
54 # timeout to something more suitable for the
55 # environment.
56 # gather_timeout = 10
57
58 # Ansible facts are available inside the ansible_facts.* dictionary
59 # namespace. This setting maintains the behaviour which was the default prior
60 # to 2.5, duplicating these variables into the main namespace, each with a
61 # prefix of 'ansible_'.
62 # This variable is set to True by default for backwards compatibility. It
63 # will be changed to a default of 'False' in a future release.
64 # ansible_facts.
65 # inject_facts_as_vars = True
66
67 # additional paths to search for roles in, colon separated
68 #roles_path    = /etc/ansible/roles
69
70 # uncomment this to disable SSH key host checking
71 host_key_checking = False
72
73 # change the default callback, you can only have one 'stdout' type enabled at a time.
74 #stdout_callback = skippy
75
76
-- INSERT --
```

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



```
ec2-user@ip-172-31-86-163:/usr/bin$ sudo pip3 install boto
WARNING: Running pip install with root privileges is generally not a good idea. Try `pip3 install --user` instead.
Collecting boto
  Downloading boto-2.49.0-py2.py3-none-any.whl (1.4 MB)
    1.4 MB 24.1 MB/s
Installing collected packages: boto
Successfully installed boto-2.49.0
[ec2-user@ip-172-31-86-163 bin]$ sudo pip3 install boto3
WARNING: Running pip install with root privileges is generally not a good idea. Try `pip3 install --user` instead.
Collecting boto3
  Downloading boto3-1.33.13-py3-none-any.whl (139 kB)
    139 kB 20.8 MB/s
Collecting botocore<1.34.0,>=1.33.13
  Downloading botocore-1.33.13-py3-none-any.whl (11.8 MB)
    11.8 MB 52.2 MB/s
Collecting jmespath<2.0.0,>=0.7.1
  Downloading jmespath-1.0.1-py3-none-any.whl (20 kB)
Collecting s3transfer<0.9.0,>=0.8.2
  Downloading s3transfer-0.8.2-py3-none-any.whl (82 kB)
    82 kB 250 kB/s
Collecting urllib3<1.27,>=1.25.4; python_version < "3.10"
  Downloading urllib3-1.26.20-py2.py3-none-any.whl (144 kB)
    144 kB 77.5 MB/s
Collecting python-dateutil<3.0.0,>=2.1
  Downloading python_dateutil-2.9.0.post0-py2.py3-none-any.whl (229 kB)
    229 kB 83.5 MB/s
Collecting six>=1.5
  Downloading six-1.17.0-py2.py3-none-any.whl (11 kB)
Installing collected packages: urllib3, six, python-dateutil, jmespath, botocore, s3transfer, boto3
Successfully installed boto3-1.33.13 botocore-1.33.13 jmespath-1.0.1 python-dateutil-2.9.0.post0 s3transfer-0.8.2 six-1.17.0 urllib3-1.26.20
[ec2-user@ip-172-31-86-163 bin]$
```



```
.26.20
[ec2-user@ip-172-31-86-163 bin]$ sudo pip3 install ansible
WARNING: Running pip install with root privileges is generally not a good idea. Try `pip3 install --user` instead.
Collecting ansible
  Downloading ansible-4.10.0.tar.gz (36.8 MB)
    36.8 MB 9.6 MB/s
Collecting ansible-core==2.11.7
  Downloading ansible-core-2.11.12.tar.gz (7.1 MB)
    7.1 MB 17.8 MB/s
Collecting jinja2
  Downloading jinja2-3.1.4-py3-none-any.whl (133 kB)
    133 kB 76.6 MB/s
Collecting PyYAML
  Downloading PyYAML-6.0.1-cp37-cp37m-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (670 kB)
    670 kB 27.2 MB/s
Collecting cryptography
  Downloading cryptography-44.0.0-cp37abi3-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (4.1 MB)
    4.1 MB 8.9 MB/s
Collecting packaging
  Downloading packaging-24.0-py3-none-any.whl (53 kB)
    53 kB 2.6 MB/s
Collecting resolvelib<0.6.0,>=0.5.3
  Downloading resolvelib-0.5.4-py2.py3-none-any.whl (12 kB)
Collecting MarkupSafe>=2.0
  Downloading MarkupSafe-2.1.5-cp37-cp37m-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (25 kB)
Collecting cffi>=1.12; platform_python_implementation != "PyPy"
  Downloading cffi-1.15.1-cp37-cp37m-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (427 kB)
    427 kB 53.5 MB/s
Collecting pycparser
  Downloading pycparser-2.21-py2.py3-none-any.whl (118 kB)
    118 kB 76.9 MB/s
Using legacy 'setup.py install' for ansible, since package 'wheel' is not installed.
Using legacy 'setup.py install' for ansible-core, since package 'wheel' is not installed.
Installing collected packages: MarkupSafe, jinja2, PyYAML, pycparser, cffi, cryptography, packaging, resolvelib, ansible-core, ansible
  Running setup.py install for ansible-core ... done
  Running setup.py install for 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 --
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 --
```

```
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 --
```

```
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 --
```

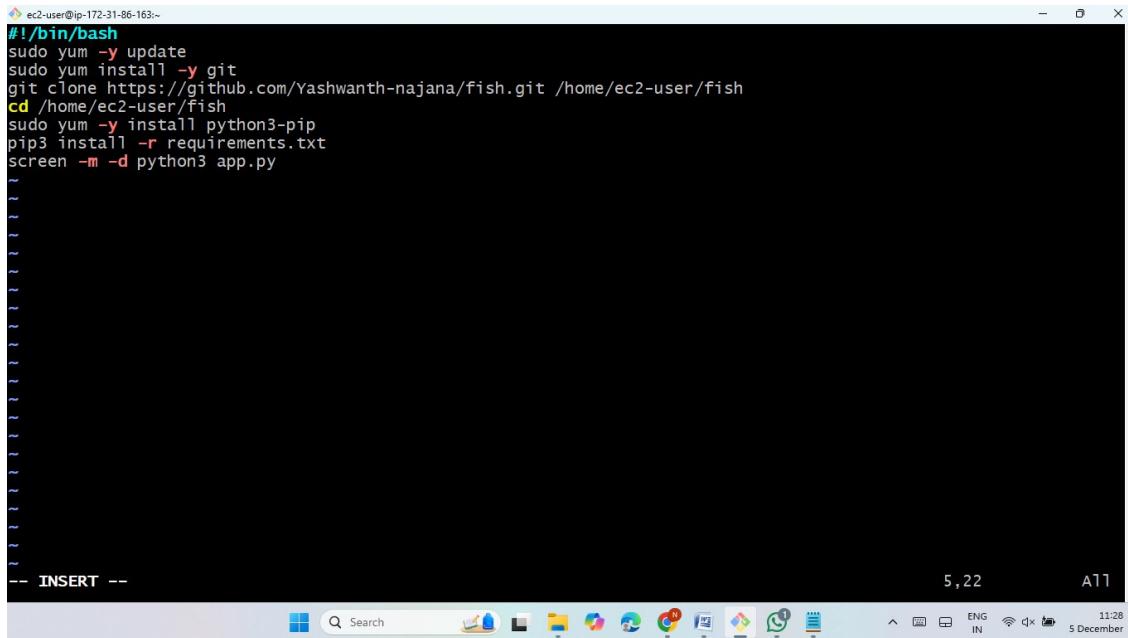
```
ec2-user@ip-172-31-86-163: ~
# Launch EC2 Instance
- amazon.aws.ec2:
  image: ami-0166fe664262f664c
  instance_type: t2.micro
  key_name: "yash"
  region: us-east-1
  wait: yes
  count: 1
  state: present
  vpc_subnet_id: "{{ pubsubnet_result.subnet.id }}"
  assign_public_ip: yes
  group_id: "{{ security_group_results.group_id }}"
  aws_access_key: ""
  aws_secret_key: ""
  user_data: "{{ lookup('file', 'data1.sh') }}"
  instance_tags:
    Name: yash-instance1

# Launch EC2 Instance2
- 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', 'data2.sh') }}"
  instance_tags:
    Name: yash-instance2

-- INSERT --
```

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

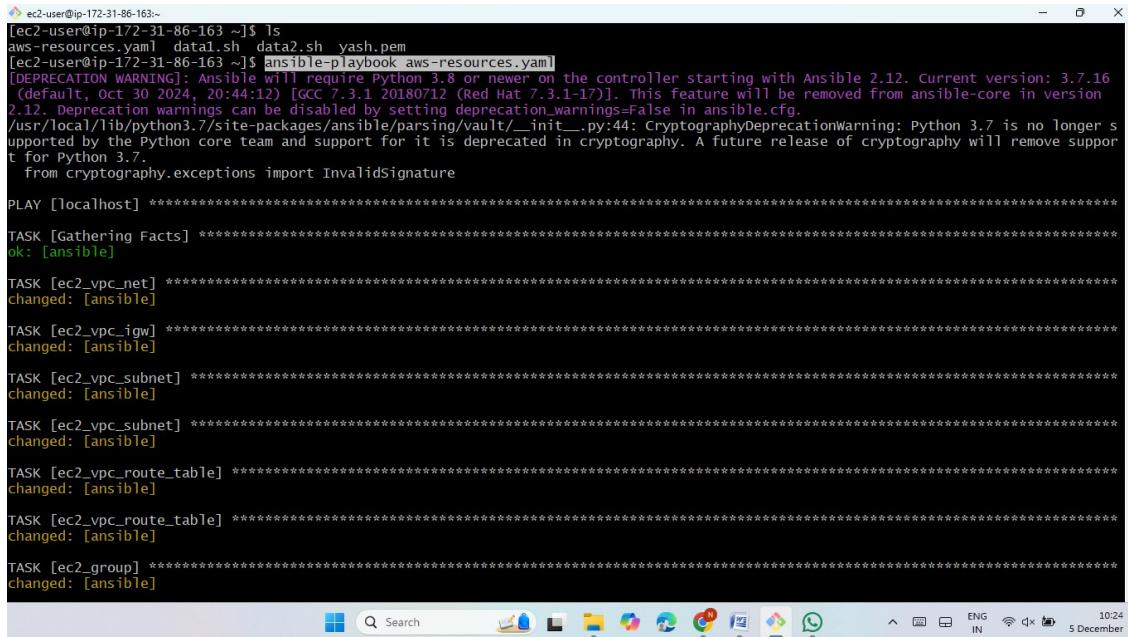
- Now write a bash script for host static application in instance2.



```
ec2-user@ip-172-31-86-163:~#
#!/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
```
-- INSERT --
5,22 A11
11:28 5 December
```

The terminal window shows the execution of a bash script. It starts with `#!/bin/bash` and performs several system updates and installations. It then clones a GitHub repository named 'fish' into the user's home directory. After cloning, it changes to the directory and installs Python 3 via pip. Finally, it runs a Python script named 'app.py' using the screen command.

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



```
ec2-user@ip-172-31-86-163:~$ ls
aws-resources.yaml data1.sh data2.sh yash.pem
[ec2-user@ip-172-31-86-163:~]$ ansible-playbook aws-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 [localhost] ****
TASK [Gathering Facts] ****
ok: [ansible]
TASK [ec2_vpc_net] ****
changed: [ansible]
TASK [ec2_vpc_igw] ****
changed: [ansible]
TASK [ec2_vpc_subnet] ****
changed: [ansible]
TASK [ec2_vpc_subnet] ****
changed: [ansible]
TASK [ec2_vpc_route_table] ****
changed: [ansible]
TASK [ec2_vpc_route_table] ****
changed: [ansible]
TASK [ec2_group] ****
changed: [ansible]
```

The terminal window shows the execution of an Ansible playbook named 'aws-resources.yaml'. The output indicates that various AWS resources (VPC, Internet Gateway, Subnets, Route Tables, and a security group) were successfully created or updated. A deprecation warning is displayed about the Python version requirement.

```

ec2-user@ip-172-31-86-163:~$ PLAY [localhost] *****
TASK [Gathering Facts] *****
ok: [ansible]

TASK [ec2_vpc_net] *****
changed: [ansible]

TASK [ec2_vpc_igw] *****
changed: [ansible]

TASK [ec2_vpc_subnet] *****
changed: [ansible]

TASK [ec2_vpc_subnet] *****
changed: [ansible]

TASK [ec2_vpc_route_table] *****
changed: [ansible]

TASK [ec2_vpc_route_table] *****
changed: [ansible]

TASK [ec2_group] *****
changed: [ansible]

TASK [amazon.aws.ec2] *****
changed: [ansible]

TASK [amazon.aws.ec2] *****
changed: [ansible]

PLAY RECAP *****
ansible : ok=10 changed=9 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0
[ec2-user@ip-172-31-86-163 ~]$
```

➤ Here the 2 instances was launched.

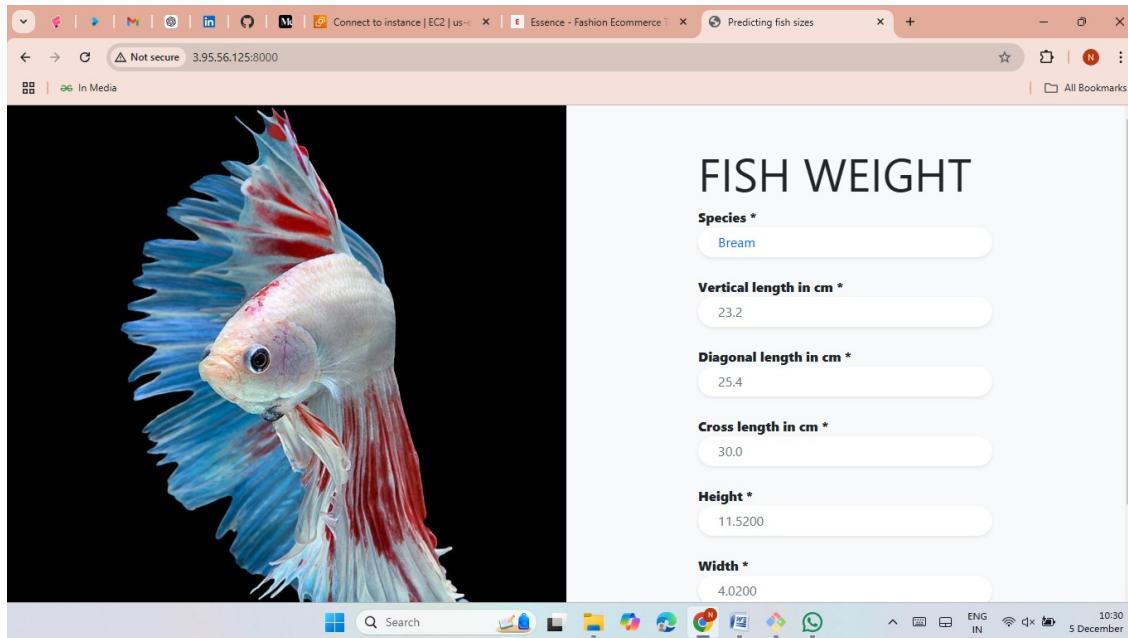
| Name           | Instance ID         | Instance state | Instance type | Status check      | Alarm status                  | Available |
|----------------|---------------------|----------------|---------------|-------------------|-------------------------------|-----------|
| yash-ansible   | i-0870ad40b99a8adf0 | Running        | t2.micro      | 2/2 checks passed | <a href="#">View alarms +</a> | us-ea     |
| yash-instance1 | i-0c732acb5e771b231 | Running        | t2.micro      | Initializing      | <a href="#">View alarms +</a> | us-ea     |
| yash-instance2 | i-0bca9235d6f876ec9 | Running        | t2.micro      | Initializing      | <a href="#">View alarms +</a> | us-ea     |

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

| Name     | VPC ID                | State     | Block Public... | IPv4 CIDR     |
|----------|-----------------------|-----------|-----------------|---------------|
| -        | vpc-05f2f9c35f844e371 | Available | Off             | 172.31.0.0/16 |
| yash-vpc | vpc-0c58fb397f5ce2674 | Available | Off             | 10.0.0.0/16   |

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

- Now copy the Instance2 Public IP then Paste 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-172-31-86-163:~$
- 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 --
```

```
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% 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 11:37 5 December

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

```
ec2-user@ip-172-31-86-163:~$ clear
[ec2-user@ip-172-31-86-163 ~]$ ls
aws-resources.yaml data1.sh data2.sh delete-aws-resources.yaml yash.pem
[ec2-user@ip-172-31-86-163 ~]$ ansible-playbook delete-aws-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] ****
changed: [ansible]
TASK [Terminate EC2 instance2] ****
changed: [ansible]
TASK [Delete the security group] ****
An exception occurred during task execution. To see the full traceback, use -vvv. The error was: botocore.exceptions.ClientError: An error occurred (DependencyViolation) when calling the DeleteSecurityGroup operation: resource sg-04b00f8dec328948d has a dependent object.
fatal: [ansible]: FAILED! => {"boto3_version": "1.33.13", "botocore_version": "1.33.13", "changed": false, "error": {"code": "DependencyViolation", "message": "resource sg-04b00f8dec328948d has a dependent object"}, "msg": "Unable to delete security group ('Description': 'allow all traffic')", "rc": 1}
[{'IpPermissions': [{"IpProtocol": '-1', 'IpRanges': [{"CidrIp": '0.0.0.0/0', 'Description': 'allow all traffic'}]}, {"IpProtocol": '-1', 'IpRanges': [{"CidrIp": '0.0.0.0/0', 'Ipv6Ranges': [{"Ipv6Ranges': [{"CidrIp": '0.0.0.0/0'}]}]}]}, {"IpProtocol": '-1', 'IpRanges': [{"CidrIp": '0.0.0.0/0', 'Ipv6Ranges': [{"Ipv6Ranges': [{"CidrIp": '0.0.0.0/0'}]}]}]}], 'PrefixListIds': [], 'UserIdGroupPairs': []}, {"OwnerId": "637423323663", 'GroupId': 'sg-04b00f8dec328948d', 'PrefixListIds': [], 'UserIdGroupPairs': []}], 'Tags': [{"Key": "Name", "Value": "yash-sg"}], 'VpcId': 'vpc-0c58fb397f5ce2674'}]: An error occurred (DependencyViolation) when calling the DeleteSecurityGroup operation: resource sg-04b00f8dec328948d has a dependent object.
 "response_metadata": {"HTTPHeaders": {"Cache-Control": "no-cache, no-store", "Connection": "close", "Content-Type": "text/xml;charset=UTF-8", "Date": "Thu, 05 Dec 2024 05:22:00 GMT", "Server": "AmazonEC2", "Strict-Transport-Security": "max-age=31536000; includeSubDomains", "Transfer-Encoding": "chunked", "Vary": "accept-encoding", "x-amzn-requestid": "ce25a065-9e2f-4b73-ab03-acde0c2d0f53"}, "RetryAttempts": 0}}

```

```
ec2-user@ip-172-31-86-163:~$ ansible-playbook delete-aws-resources.yaml
, "request_id": "ce25a065-9e2f-4b73-ab03-acde0c2d0f53", "retry_attempts": 0}
PLAY RECAP ****
ansible : ok=2 changed=2 unreachable=0 failed=1 skipped=0 rescued=0 ignored=0
[ec2-user@ip-172-31-86-163 ~]$ ansible-playbook delete-aws-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] ****
changed: [ansible]
TASK [ec2_vpc_route_table] ****
changed: [ansible]
TASK [Delete Public Subnet] ****
changed: [ansible]
TASK [Delete Private Subnet] ****
changed: [ansible]

```

```

t 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] ****
changed: [ansible]
TASK [ec2_vpc_route_table] ****
changed: [ansible]
TASK [Delete Public Subnet] ****
changed: [ansible]
TASK [Delete Private Subnet] ****
changed: [ansible]
TASK [ec2_vpc_igw] ****
changed: [ansible]
TASK [ec2_vpc_net] ****
changed: [ansible]

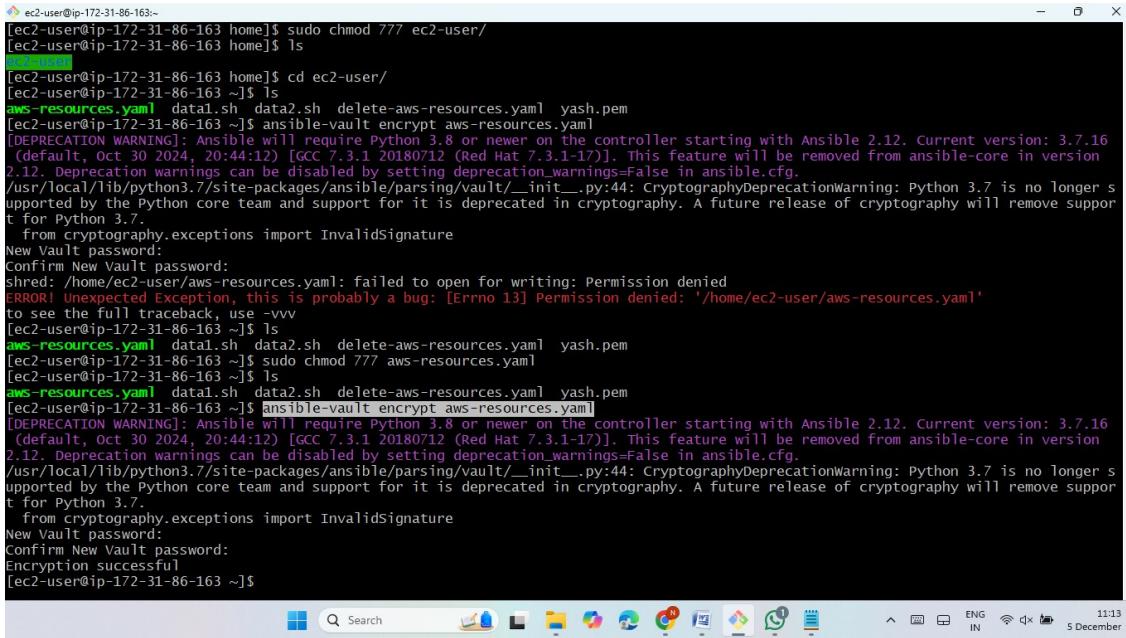
PLAY RECAP ****
ansible : ok=9 changed=7 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0
[ec2-user@ip-172-31-86-163 ~]$ |

```

➤ Here the instances was deleted.

| Name           | Instance ID         | Instance state | Type     | Status check      | Alarm status                | Availability zone |
|----------------|---------------------|----------------|----------|-------------------|-----------------------------|-------------------|
| yash-ansible   | i-0870ad40b99a8adf0 | Running        | t2.micro | 2/2 checks passed | <a href="#">View alarms</a> | us-east-1         |
| yash-instance1 | i-0c732acb5e771b231 | Terminated     | t2.micro | -                 | <a href="#">View alarms</a> | us-east-1         |
| yash-instance2 | i-0bca9235d6f876ec9 | Terminated     | t2.micro | -                 | <a href="#">View alarms</a> | us-east-1         |

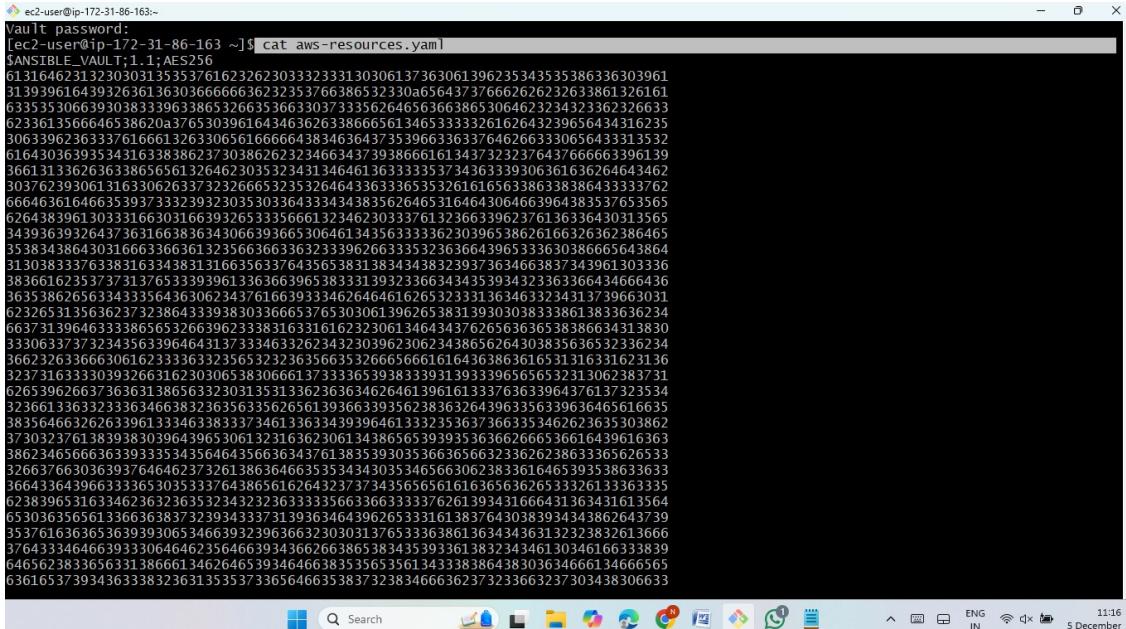
- Now Give 777 permissions to /home/ec2-user/filename.
- Now encrypt the created file with the command.  
**ansible-vault encrypt aws-resources.yml**



```

[ec2-user@ip-172-31-86-163 ~]$ sudo chmod 777 ec2-user/
[ec2-user@ip-172-31-86-163 home]$ ls
[ec2-user@ip-172-31-86-163 ~]$ cd ec2-user/
[ec2-user@ip-172-31-86-163 ~]$ ls
aws-resources.yaml data1.sh data2.sh delete-aws-resources.yaml yash.pem
[ec2-user@ip-172-31-86-163 ~]$ ansible-vault encrypt aws-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
New Vault password:
Confirm New Vault password:
shred: /home/ec2-user/aws-resources.yaml: failed to open for writing: Permission denied
ERROR! Unexpected Exception, this is probably a bug: [Errno 13] Permission denied: '/home/ec2-user/aws-resources.yaml'
to see the full traceback, use -vvv
[ec2-user@ip-172-31-86-163 ~]$ ls
aws-resources.yaml data1.sh data2.sh delete-aws-resources.yaml yash.pem
[ec2-user@ip-172-31-86-163 ~]$ sudo chmod 777 aws-resources.yaml
[ec2-user@ip-172-31-86-163 ~]$ ls
aws-resources.yaml data1.sh data2.sh delete-aws-resources.yaml yash.pem
[ec2-user@ip-172-31-86-163 ~]$ ansible-vault encrypt aws-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
New Vault password:
Confirm New Vault password:
Encryption successful
[ec2-user@ip-172-31-86-163 ~]$
```

- Here the file was encrypted successfully.



```

Vault password:
[ec2-user@ip-172-31-86-163 ~]$ cat aws-resources.yaml
$ANSIBLE_VAULT;1;AES256
613164623132303031355376162326230333233130306137363061396235343535386336303961
313939616439326361363036666663623235376386532330a656437376662626232633861326161
63353506639303833396338653266353663303733562646536638653064623234323362326633
6233613566646538620a376530396164346362633866561346533332616243239656434316235
30633962363337616661326330656166666438834636437353966336337646263633065643313532
6164303639353431633838623730386262323463437393866613437323276437666663396139
366131336236338656561326462303532343134646136333353734363339306361636264643462
303762393061316330626337323266532353264643363336535326161656338633838643333762
66646361646635393733239323053033643343438356264653164643064663964383537653565
6264389613033316630316639326533356661323462303337613236633962376136336430313565
343936393264373631663836340639365306461343563333620396538626166326362386465
353834386430316663366361323566366332339626633352363664396533630386665643864
313038333763381633438313166356337643565383138343438323937363463837343961303336
38366162356326363138656332303135313362363643626461396161333763633964376137323534
3236613363323336346638323635633562656139366339356238363264396335639636465616635
38358626563343335643630623437616639334362323313634633234313739663031
62326531356362373286433938303366653765303061396265383139303038333861383363234
66373196463633865632663962338316316162323061346434376265636358386634313830
333063373732343563396464313733463326234323039623062348656264303853636532336234
3662326336630612333363323565323236563326665661646363863616531316331623136
32373163333039326631623030653830666137333653938333913933396565653213062383731
6265396266736363138656332303135313362363643626461396161333763633964376137323534
3236613363323336346638323635633562656139366339356238363264396335639636465616635
38354663262633961333436338337346133633439646133323536373663353462623635303862
37303237613839383039643965306132316362306134386565393353662666536616439616363
3862346566363393335343564643566363437613835393030536635663233626238633365626533
32663766303639376464623732613863646635353434303534656630623833616465393538633633
36643364396633336530353376438656162643237373435656561616365362653326133363335
62383965316346236326353234323263633566336633335376261393431666431363431613564
653036356561336633638373239343337313936346439626533316138376430383934343862643739
35376136365363939306346639323963663230301376533363861363434363132323832613666
376433464663933306464623564663934366263865383435393361383234346130346166333839
64656238336563313866613462646539346466383535653561343338386438303634666134666565
636165373934363338323631353573365646635383732383466636237303438306633
```

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

```

ec2-user@ip-172-31-86-163:~$ ansible-vault decrypt aws-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
Vault password:
Decryption successful
[ec2-user@ip-172-31-86-163 ~]$

```

- Here the file was decrypted.

```

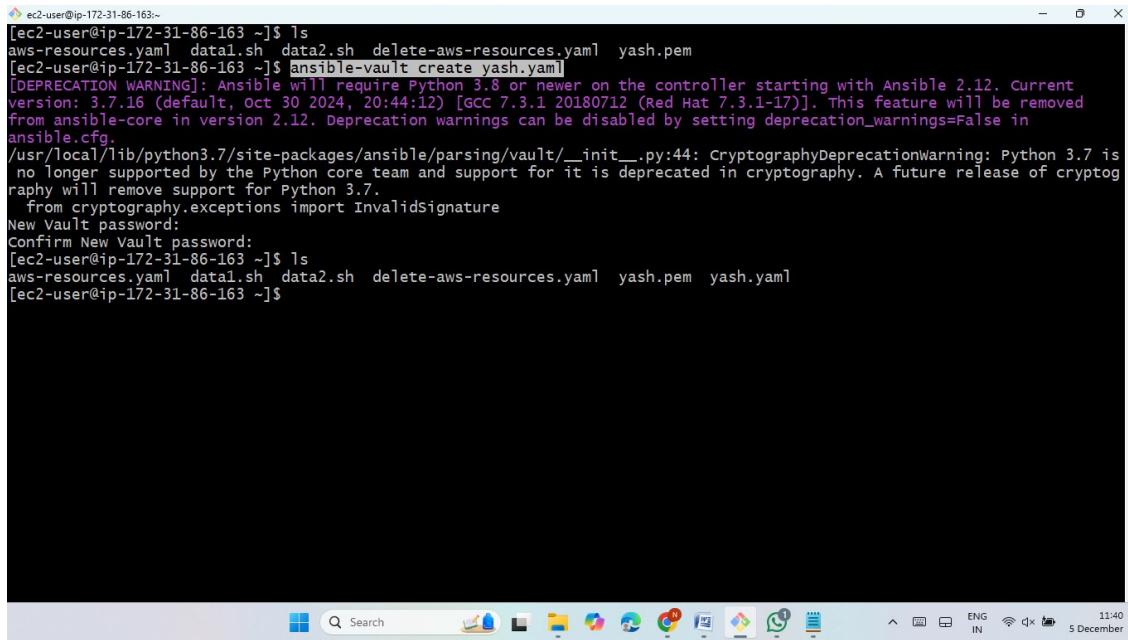
ec2-user@ip-172-31-86-163:~$ cat aws-resources.yaml
VPC Creation

- hosts: localhost
 become: yes
 tasks:
 - ec2_vpc_net:

```

- Now create an encrypt files with the command.

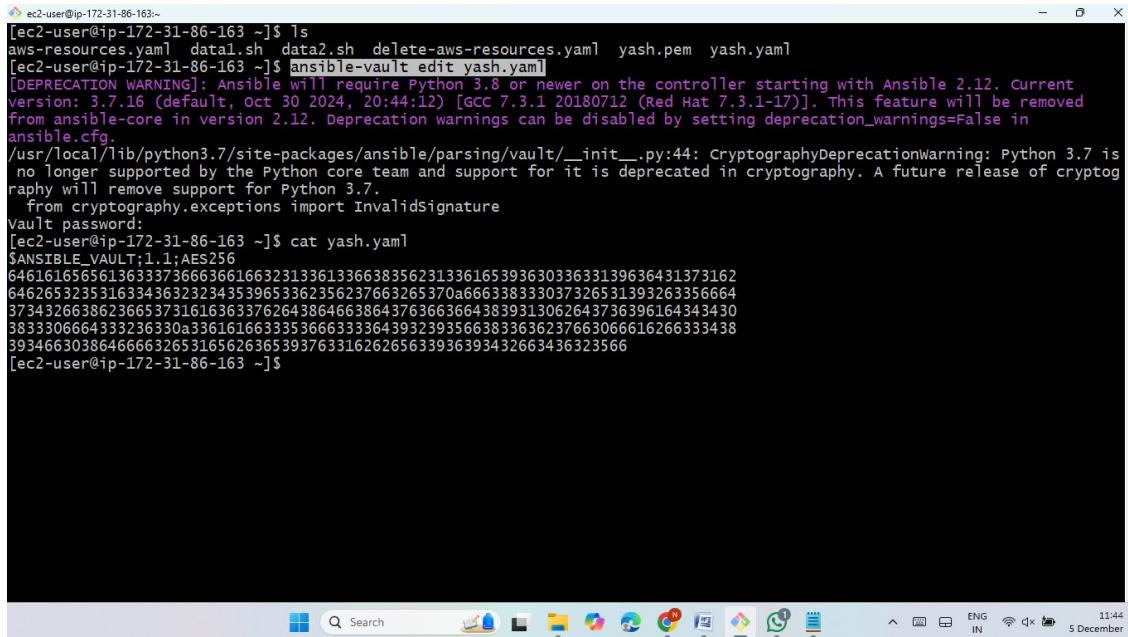
```
ansible-vault create yash.yml
```



```
[ec2-user@ip-172-31-86-163 ~]$ ls
aws-resources.yaml data1.sh data2.sh delete-aws-resources.yaml yash.pem
[ec2-user@ip-172-31-86-163 ~]$ ansible-vault create yash.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 crypto-
graphy will remove support for Python 3.7.
 from cryptography.exceptions import InvalidSignature
New Vault password:
Confirm New Vault password:
[ec2-user@ip-172-31-86-163 ~]$ ls
aws-resources.yaml data1.sh data2.sh delete-aws-resources.yaml yash.pem yash.yaml
[ec2-user@ip-172-31-86-163 ~]$
```

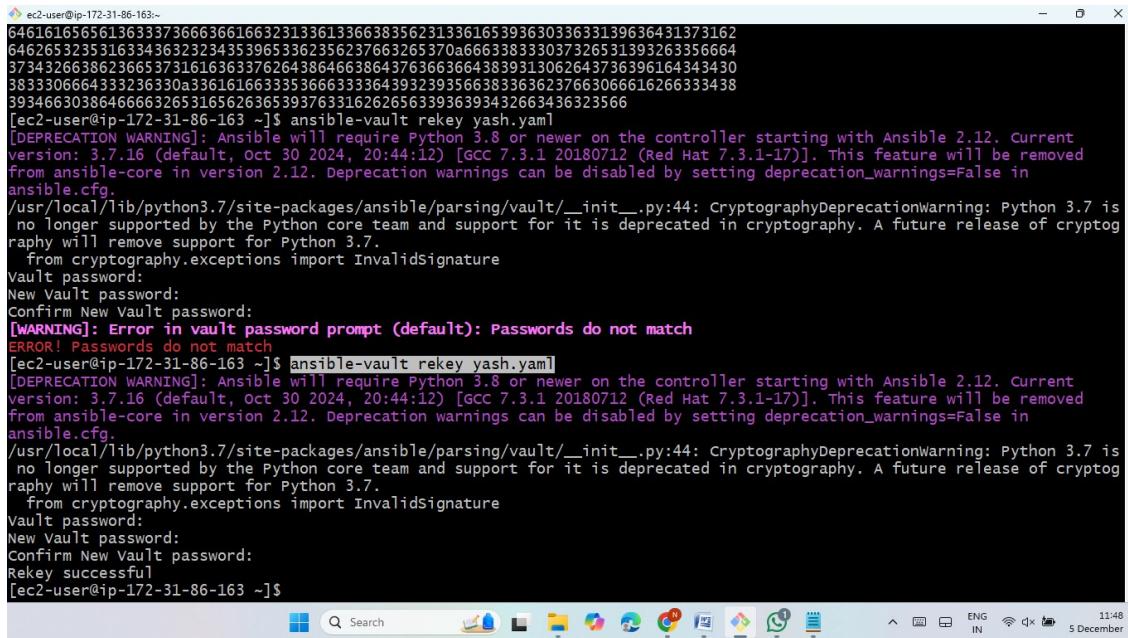
- Now Edit the encrypted file with the command.

```
ansible-vault edit yash.yaml
```



```
[ec2-user@ip-172-31-86-163 ~]$ ls
aws-resources.yaml data1.sh data2.sh delete-aws-resources.yaml yash.pem yash.yaml
[ec2-user@ip-172-31-86-163 ~]$ ansible-vault edit yash.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 crypto-
graphy will remove support for Python 3.7.
 from cryptography.exceptions import InvalidSignature
Vault password:
[ec2-user@ip-172-31-86-163 ~]$ cat yash.yaml
$ANSIBLE_VAULT;1.1;AES256
6461656561363337366636616632313361336638385623133616539363033633139636431373162
6462653235316334363232343539653362356237663265370a666338333037326531393263356664
373432663862366537316136337626438646386437636636643839313062643736396164343430
3833306664333236330a336161663335366633336439323935663833636237663066616266333438
3934663038646666326531656263653937633162626533936393432663436323566
[ec2-user@ip-172-31-86-163 ~]$
```

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



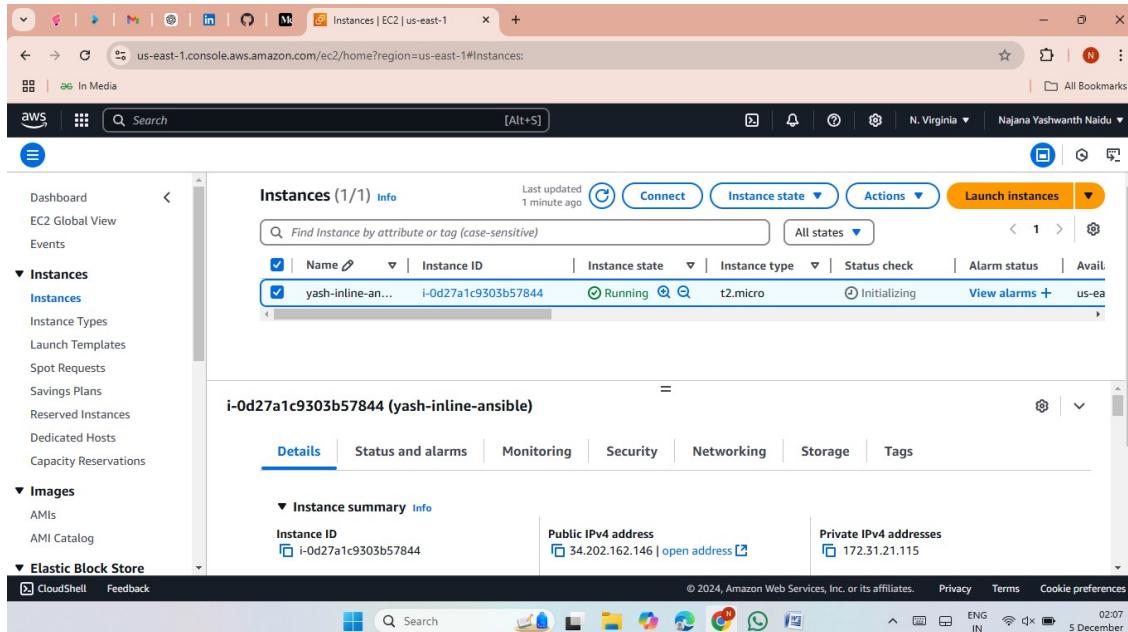
```

ec2-user@ip-172-31-86-163:~$ ansible-vault rekey yash.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
Vault password:
New Vault password:
Confirm New Vault password:
[WARNING]: Error in vault password prompt (default): Passwords do not match
ERROR! Passwords do not match
[ec2-user@ip-172-31-86-163:~]$ ansible-vault rekey yash.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
Vault password:
New Vault password:
Confirm New Vault password:
Rekey successful
[ec2-user@ip-172-31-86-163:~$]

```

## 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**

```
ec2-user@ip-172-31-21-115:~$ ssh -i "yash.pem" ec2-user@ec2-34-202-162-146.compute-1.amazonaws.com
The authenticity of host 'ec2-34-202-162-146.compute-1.amazonaws.com (34.202.162.146)' can't be established.
ED25519 key fingerprint is SHA256:azdMvxyjykIVYjel9twOYPIzhWrEuvthuwr7ZITU.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-34-202-162-146.compute-1.amazonaws.com' (ED25519) to the list of known hosts.

Amazon Linux 2
AL2 End of Life is 2025-06-30.
A newer version of Amazon Linux is available!
Amazon Linux 2023, GA and supported until 2028-03-15.
https://aws.amazon.com/linux/amazon-linux-2023/

[ec2-user@ip-172-31-21-115 ~]$ sudo yum -y update
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
No packages marked for update
[ec2-user@ip-172-31-21-115 ~]$ amazon-linux-extras install ansible2
Topic ansible2 has end-of-support date of 2023-09-30
You lack permissions to write to system configuration. /etc/yum.repos.d/amzn2-extras.repo
[ec2-user@ip-172-31-21-115 ~]$ sudo amazon-linux-extras install ansible2
Topic ansible2 has end-of-support date of 2023-09-30
Installing ansible
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Cleaning repos: amzn2-core amzn2extra-ansible2 amzn2extra-docker amzn2extra-kernel-5.10
1/ metadata files removed
6 sqlite files removed
0 metadata files removed
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core

| 3.6 kB 00:00:00
ENG IN 02:09
5 December
```

```
36 BCC available [=0.x =stable]
37 mono available [=5.x =stable]
38 nginx1x available [=stable]
40 mock available [=stable]
43 livepatch available [=stable]
45 haproxy2 available [=stable]
46 collectd available [=stable]
47 aws-nitro-enclaves-cli available [=stable]
48 R4 available [=stable]
kernel-5.4 available [=stable]
50 selinux-ng available [=stable]
52 tomcat9 available [=stable]
53 unbound1.13 available [=stable]
54 mariadb10.5 available [=stable]
55 kernel-5.10=latest enabled [=stable]
56 redis available [=stable]
59 postgresql13 available [=stable]
60 mock2 available [=stable]
61 dnsmasq_2.85 available [=stable]
62 kernel-5.15 available [=stable]
63 postgresql14 available [=stable]
64 firefox available [=stable]
65 lustre available [=stable]
67 awscli1 available [=stable]
68 tphp8.2 available [=stable]
69 dnsmasq available [=stable]
70 unbound1.17 available [=stable]
72 collectd-python3 available [=stable]
* Extra topic has reached end of support.
† Note on end-of-support. Use 'info' subcommand.
[ec2-user@ip-172-31-21-115 ~]$ ansible --version
ansible 2.9.23
 config file = /etc/ansible/ansible.cfg
 configured module search path = ['~/home/ec2-user/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
 ansible python module location = /usr/lib/python2.7/site-packages/ansible
 executable location = /usr/bin/ansible
 python version = 2.7.18 (default, Oct 17 2024, 18:59:07) [GCC 7.3.1 20180712 (Red Hat 7.3.1-17)]
[ec2-user@ip-172-31-21-115 ~]$
```

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

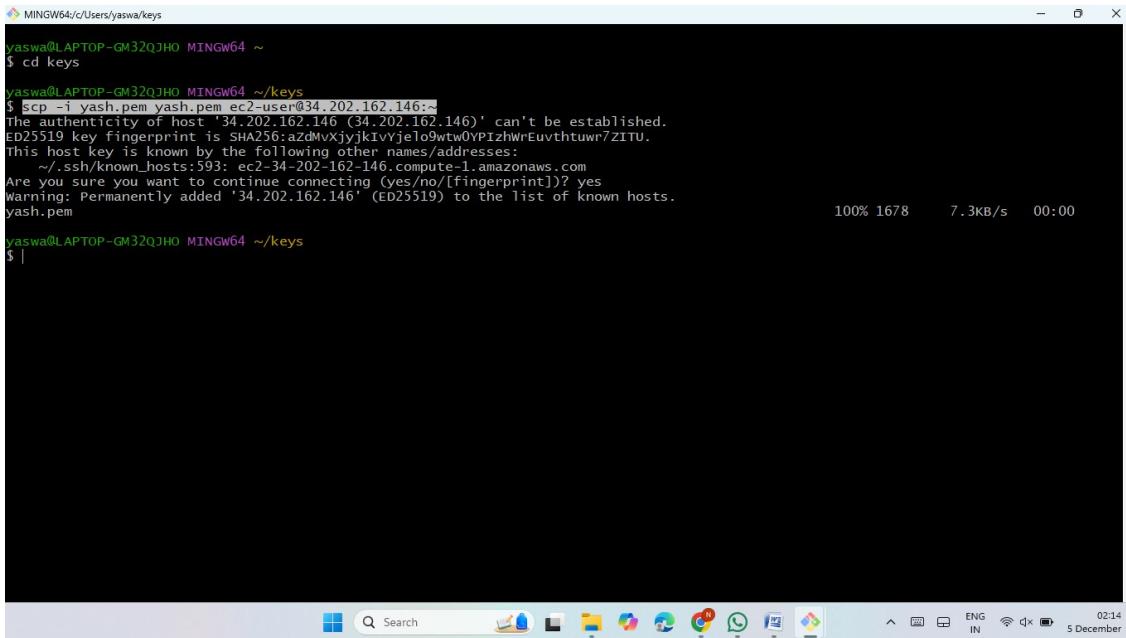
[localhost]

```
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
```

```
ec2-user@ip-172-31-21-115:~$ cd /etc/ansible/
[ec2-user@ip-172-31-21-115 ansible]$ ls
ansible.cfg hosts roles
[ec2-user@ip-172-31-21-115 ansible]$ sudo vi hosts
[ec2-user@ip-172-31-21-115 ansible]$ pwd
/etc/ansible
[ec2-user@ip-172-31-21-115 ansible]$ |
```

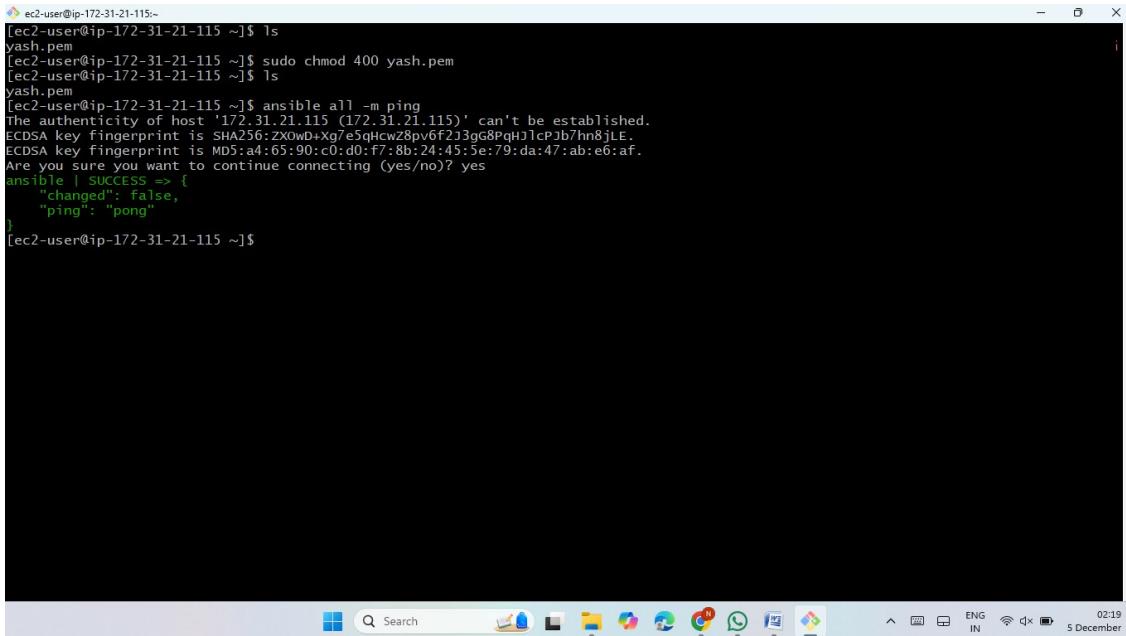
```
[localhost]
ansible ansible_host=172.31.21.115 ansible_python_interpreter=/usr/bin/python3 ansible_user=ec2-user ansible_ssh_private_key_file=/home/ec2-user/yash.pem
This is the default ansible 'hosts' file.
#
It should live in /etc/ansible/hosts
#
- Comments begin with the '#' character
- Blank lines are ignored
- Groups of hosts are delimited by [header] elements
- You can enter hostnames or ip addresses
- A hostname/ip can be a member of multiple groups
#
Ex 1: Ungrouped hosts, specify before any group headers.
green.example.com
blue.example.com
192.168.1.100
192.168.1.10
#
Ex 2: A collection of hosts belonging to the 'webservers' group
[webservers]
alpha.example.org
beta.example.org
192.168.1.100
192.168.1.110
#
If you have multiple hosts following a pattern you can specify
them like this:
www[001:006].example.com
#
Ex 3: A collection of database servers in the 'dbservers' group
[dbservers]
##
```

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



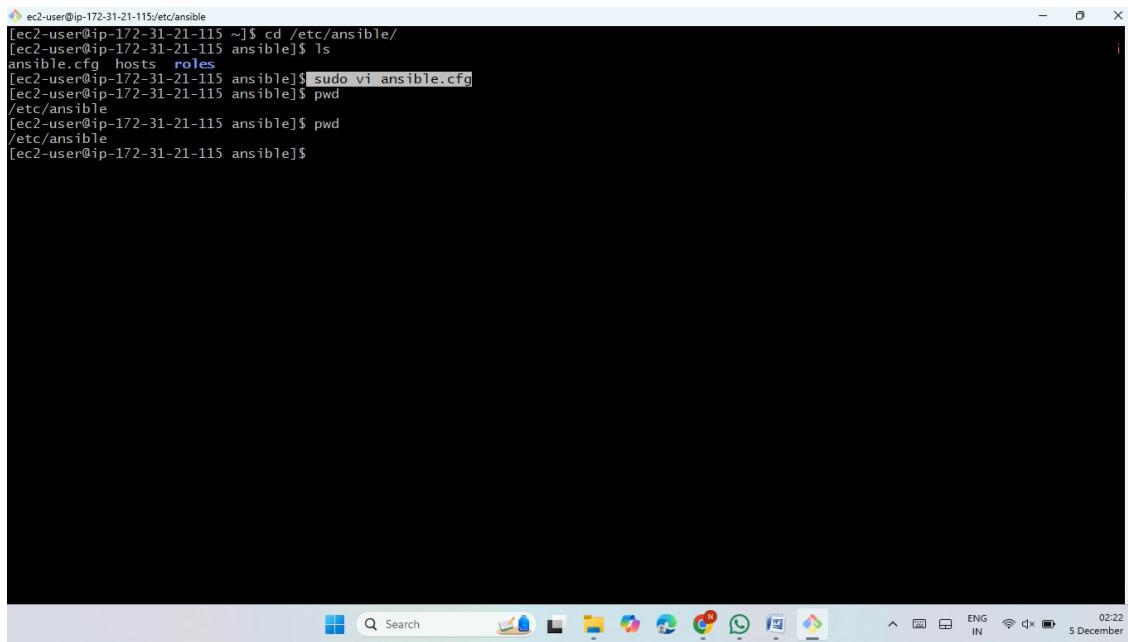
```
MINGW64:/c/Users/yaswa/keys
yaswa@LAPTOP-GM32QJHO MINGW64 ~
$ cd keys
yaswa@LAPTOP-GM32QJHO MINGW64 ~/keys
$ scp -i yash.pem yash.pem ec2-user@34.202.162.146:.
The authenticity of host '34.202.162.146' (34.202.162.146) can't be established.
ED25519 key fingerprint is SHA256:azdMvXjyjkIVYjelo9wtw0YPiZhwrEuvtthuwr7ZITU.
This host key is known by the following other names/addresses:
 ~/.ssh/known_hosts:93: ec2-34-202-162-146.compute-1.amazonaws.com
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '34.202.162.146' (ED25519) to the list of known hosts.
yash.pem 100% 1678 7.3KB/s 00:00
yaswa@LAPTOP-GM32QJHO MINGW64 ~/keys
$ |
```

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

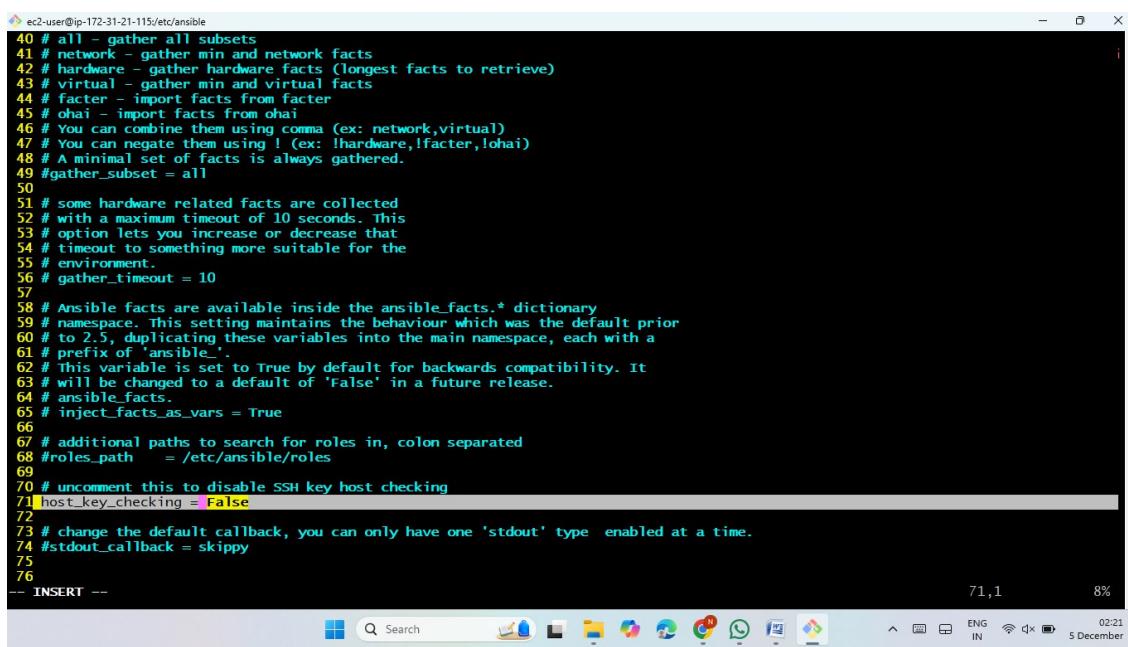


```
ec2-user@ip-172-31-21-115:~
[ec2-user@ip-172-31-21-115 ~]$ ls
yash.pem
[ec2-user@ip-172-31-21-115 ~]$ sudo chmod 400 yash.pem
[ec2-user@ip-172-31-21-115 ~]$ ls
yash.pem
[ec2-user@ip-172-31-21-115 ~]$ ansible all -m ping
The authenticity of host '172.31.21.115 (172.31.21.115)' can't be established.
ED25519 key fingerprint is SHA256:ZX0wD+Xg7e5qHcwZ8pv6f2j3gc8PgHJ1cPjB7hn8jLE.
EDDSA key fingerprint is MD5:a4:65:90:c0:d0:f7:8b:24:45:5e:79:da:47:ab:e6:af.
Are you sure you want to continue connecting (yes/no)? yes
ansible | SUCCESS => {
 "changed": false,
 "ping": "pong"
}
[ec2-user@ip-172-31-21-115 ~]$
```

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

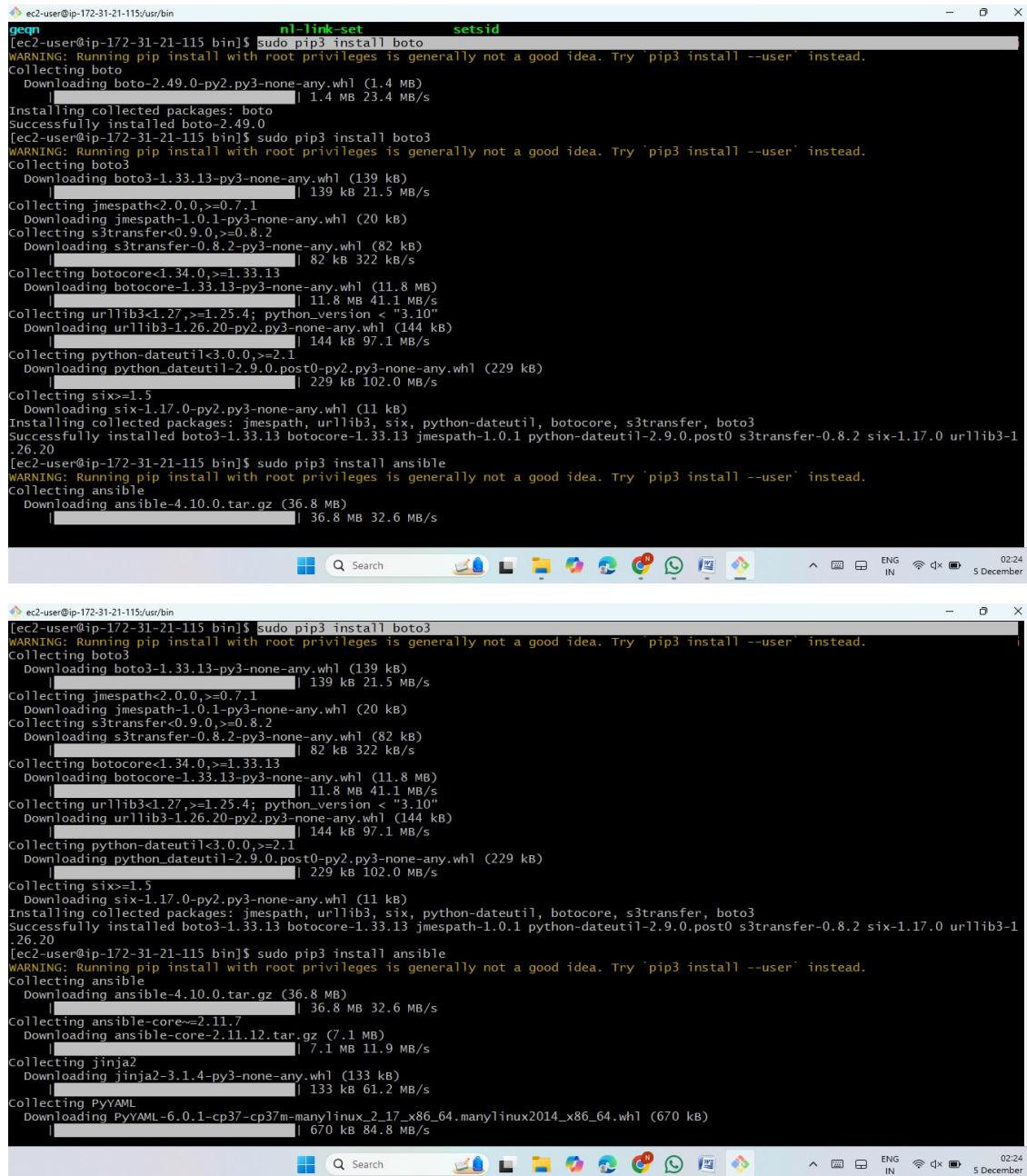


```
[ec2-user@ip-172-31-21-115 ~]$ cd /etc/ansible/
[ec2-user@ip-172-31-21-115 ansible]$ ls
ansible.cfg hosts roles
[ec2-user@ip-172-31-21-115 ansible]$ sudo vi ansible.cfg
[ec2-user@ip-172-31-21-115 ansible]$ pwd
/etc/ansible
[ec2-user@ip-172-31-21-115 ansible]$ pwd
/etc/ansible
[ec2-user@ip-172-31-21-115 ansible]$
```



```
40 # all - gather all subsets
41 # network - gather min and network facts
42 # hardware - gather hardware facts (longest facts to retrieve)
43 # virtual - gather min and virtual facts
44 # facter - import facts from facter
45 # ohai - import facts from ohai
46 # You can combine them using comma (ex: network,virtual)
47 # You can negate them using ! (ex: !hardware,!facter,!ohai)
48 # A minimal set of facts is always gathered.
49 #gather_subset = all
50
51 # some hardware related facts are collected
52 # with a maximum timeout of 10 seconds. This
53 # option lets you increase or decrease that
54 # timeout to something more suitable for the
55 # environment.
56 # gather_timeout = 10
57
58 # Ansible facts are available inside the ansible_facts.* dictionary
59 # namespace. This setting maintains the behaviour which was the default prior
60 # to 2.5, duplicating these variables into the main namespace, each with a
61 # prefix of 'ansible_'.
62 # this variable is set to True by default for backwards compatibility. It
63 # will be changed to a default of 'False' in a future release.
64 # ansible_facts.
65 # inject_facts_as_vars = True
66
67 # additional paths to search for roles in, colon separated
68 #roles_path = /etc/ansible/roles
69
70 # uncomment this to disable SSH key host checking
71 host_key_checking = False
72
73 # change the default callback, you can only have one 'stdout' type enabled at a time.
74 #stdout_callback = skippy
75
76
-- INSERT --
```

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



```
ec2-user@ip-172-31-21-115:~$ sudo pip3 install boto
WARNING: Running pip install with root privileges is generally not a good idea. Try 'pip3 install --user' instead.
Collecting boto
 Downloading boto-2.49.0-py2.py3-none-any.whl (1.4 MB)
[0%|] 1.4 MB 23.4 MB/s
Installing collected packages: boto
Successfully installed boto-2.49.0
[ec2-user@ip-172-31-21-115:~$ sudo pip3 install boto3
WARNING: Running pip install with root privileges is generally not a good idea. Try 'pip3 install --user' instead.
Collecting boto3
 Downloading boto3-1.33.13-py3-none-any.whl (139 kB)
[0%|] 139 kB 21.5 MB/s
Collecting jmespath<2.0.0,>=0.7.1
 Downloading jmespath-1.0.1-py3-none-any.whl (20 kB)
Collecting s3transfer<0.9.0,>=0.8.2
 Downloading s3transfer-0.8.2-py3-none-any.whl (82 kB)
[0%|] 82 kB 322 kB/s
Collecting botocore<1.34.0,>=1.33.13
 Downloading botocore-1.33.13-py3-none-any.whl (11.8 MB)
[0%|] 11.8 MB 41.1 MB/s
Collecting urllib3<1.27,>=1.25.4; python_version < "3.10"
 Downloading urllib3-1.26.20-py2.py3-none-any.whl (144 kB)
[0%|] 144 kB 97.1 MB/s
Collecting python-dateutil<3.0.0,>=2.1
 Downloading python_dateutil-2.9.0.post0-py2.py3-none-any.whl (229 kB)
[0%|] 229 kB 102.0 MB/s
Collecting six>=1.5
 Downloading six-1.17.0-py2.py3-none-any.whl (11 kB)
Installing collected packages: jmespath, urllib3, six, python-dateutil, botocore, s3transfer, boto3
Successfully installed boto3-1.33.13 botocore-1.33.13 jmespath-1.0.1 python-dateutil-2.9.0.post0 s3transfer-0.8.2 six-1.17.0 urllib3-1.26.20
[ec2-user@ip-172-31-21-115:~$ sudo pip3 install ansible
WARNING: Running pip install with root privileges is generally not a good idea. Try 'pip3 install --user' instead.
Collecting ansible
 Downloading ansible-4.10.0.tar.gz (36.8 MB)
[0%|] 36.8 MB 32.6 MB/s

ec2-user@ip-172-31-21-115:~$ sudo pip3 install boto3
WARNING: Running pip install with root privileges is generally not a good idea. Try 'pip3 install --user' instead.
Collecting boto3
 Downloading boto3-1.33.13-py3-none-any.whl (139 kB)
[0%|] 139 kB 21.5 MB/s
Collecting jmespath<2.0.0,>=0.7.1
 Downloading jmespath-1.0.1-py3-none-any.whl (20 kB)
Collecting s3transfer<0.9.0,>=0.8.2
 Downloading s3transfer-0.8.2-py3-none-any.whl (82 kB)
[0%|] 82 kB 322 kB/s
Collecting botocore<1.34.0,>=1.33.13
 Downloading botocore-1.33.13-py3-none-any.whl (11.8 MB)
[0%|] 11.8 MB 41.1 MB/s
Collecting urllib3<1.27,>=1.25.4; python_version < "3.10"
 Downloading urllib3-1.26.20-py2.py3-none-any.whl (144 kB)
[0%|] 144 kB 97.1 MB/s
Collecting python-dateutil<3.0.0,>=2.1
 Downloading python_dateutil-2.9.0.post0-py2.py3-none-any.whl (229 kB)
[0%|] 229 kB 102.0 MB/s
Collecting six>=1.5
 Downloading six-1.17.0-py2.py3-none-any.whl (11 kB)
Installing collected packages: jmespath, urllib3, six, python-dateutil, botocore, s3transfer, boto3
Successfully installed boto3-1.33.13 botocore-1.33.13 jmespath-1.0.1 python-dateutil-2.9.0.post0 s3transfer-0.8.2 six-1.17.0 urllib3-1.26.20
[ec2-user@ip-172-31-21-115:~$ sudo pip3 install ansible
WARNING: Running pip install with root privileges is generally not a good idea. Try 'pip3 install --user' instead.
Collecting ansible
 Downloading ansible-4.10.0.tar.gz (36.8 MB)
[0%|] 36.8 MB 32.6 MB/s
Collecting ansible-core=<2.11.7
 Downloading ansible-core-2.11.12.tar.gz (7.1 MB)
[0%|] 7.1 MB 11.9 MB/s
Collecting jinja2
 Downloading jinja2-3.1.4-py3-none-any.whl (133 kB)
[0%|] 133 kB 61.2 MB/s
Collecting PyYAML
 Downloading PyYAML-6.0.1-cp37-cp37m-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (670 kB)
[0%|] 670 kB 84.8 MB/s
```

```

[ec2-user@ip-172-31-21-115 ~]$ sudo pip3 install ansible
WARNING: Running pip install with root privileges is generally not a good idea. Try 'pip3 install --user' instead.
Collecting ansible
 Downloading ansible-4.10.0.tar.gz (36.8 MB)
 36.8 MB 32.6 MB/s
Collecting ansible-core==2.11.7
 Downloading ansible-core-2.11.12.tar.gz (7.1 MB)
 7.1 MB 11.9 MB/s
Collecting jinja2
 Downloading jinja2-3.1.4-py3-none-any.whl (133 kB)
 133 kB 61.2 MB/s
Collecting PyYAML
 Downloading PyYAML-6.0.1-cp37-cp37m-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (670 kB)
 670 kB 84.8 MB/s
Collecting cryptography
 Downloading cryptography-44.0.0-cp37abi3-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (4.1 MB)
 4.1 MB 17.2 MB/s
Collecting packaging
 Downloading packaging-24.0-py3-none-any.whl (53 kB)
 53 kB 1.5 MB/s
Collecting resolvelib>=0.5.3
 Downloading resolvelib-0.5.4-py2.py3-none-any.whl (12 kB)
Collecting MarkupSafe>=2.0
 Downloading MarkupSafe-2.1.5-cp37-cp37m-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (25 kB)
Collecting cffi>=1.12; platform_python_implementation != "PyPy"
 Downloading cffi-1.15.1-cp37-cp37m-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (427 kB)
 427 kB 56.7 MB/s
Collecting pycparser
 Downloading pycparser-2.21-py2.py3-none-any.whl (118 kB)
 118 kB 91.7 MB/s
Using legacy 'setup.py install' for ansible, since package 'wheel' is not installed.
Using legacy 'setup.py install' for ansible-core, since package 'wheel' is not installed.
Installing collected packages: MarkupSafe, jinja2, PyYAML, pycparser, cffi, cryptography, packaging, resolvelib, ansible-core, ansible
 Running setup.py install for ansible-core ... done
 Running setup.py install for ansible ... done
Successfully installed MarkupSafe-2.1.5 PyYAML-6.0.1 ansible-4.10.0 ansible-core-2.11.12 cffi-1.15.1 cryptography-44.0.0 jinja2-3.1.4
packaging-24.0 pycparser-2.21 resolvelib-0.5.4
[ec2-user@ip-172-31-21-115 ~]$
```

- 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 ~]$ sudo vi aws-ansible-resources.yaml
[ec2-user@ip-172-31-21-115 ~]$ sudo vi data1.sh
[ec2-user@ip-172-31-21-115 ~]$ sudo vi data2.sh
[ec2-user@ip-172-31-21-115 ~]$
```

```
ec2-user@ip-172-31-21-115:~
- hosts: localhost
 become: yes
 gather_facts: false

vars:
 aws_access_key: ""
 aws_secret_key: ""
 title: "ASH"
 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
```

```
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%
05:12 ENG IN 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: "{{ pvzone }}"
 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 --
ec2-user@ip-172-31-21-115:~$

```

```

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

```

```

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 }}"
-- INSERT --
ec2-user@ip-172-31-21-115:~$

```

```
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 --
```

```
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 --
```

- Now run the AWS resources with the command.

## Ansible-playbook <name of file>

```
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 ~]$ sudo vi ansible-resources.yaml
[ec2-user@ip-172-31-21-115 ~]$ ansible-playbook ansible-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 [localhost] ****
TASK [ec2_vpc_net] ****
ok: [ansible]
ok: [ansible]

TASK [ec2_vpc_igw] ****
ok: [ansible]
ok: [ansible]

TASK [ec2_vpc_subnet] ****
changed: [ansible]
changed: [ansible]

TASK [ec2_vpc_subnet] ****
changed: [ansible]
changed: [ansible]

TASK [ec2_vpc_route_table] ****
changed: [ansible]
changed: [ansible]

TASK [ec2_vpc_route_table] ****
changed: [ansible]
changed: [ansible]

TASK [ec2_group] ****
changed: [ansible]

TASK [ec2] ****
changed: [ansible]

TASK [ec2] ****
changed: [ansible]

PLAY RECAP ****
ansible : ok=0 changed=7 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0
[ec2-user@ip-172-31-21-115 ~]$
```

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

The screenshot shows the AWS EC2 Instances page. At the top, there are buttons for Connect, Instance state, Actions, and Launch instances. Below this is a search bar and a dropdown for 'All states'. A table lists three instances:

| Name              | Instance ID         | Instance state | Instance type | Status check      | Alarm status                | Availability Zone | Public IPv4 DNS |
|-------------------|---------------------|----------------|---------------|-------------------|-----------------------------|-------------------|-----------------|
| yash-instance1    | i-0de486a3664fc7239 | Running        | t2.micro      | Initializing      | <a href="#">View alarms</a> | us-east-1a        | ec2-34-228-116  |
| yash-instance2    | i-03af79c45ff152e76 | Running        | t2.micro      | Initializing      | <a href="#">View alarms</a> | us-east-1a        | ec2-54-226-79-  |
| yash-inline-an... | i-0d27a1c9303b57844 | Running        | t2.micro      | 2/2 checks passed | <a href="#">View alarms</a> | us-east-1a        | ec2-34-202-162  |

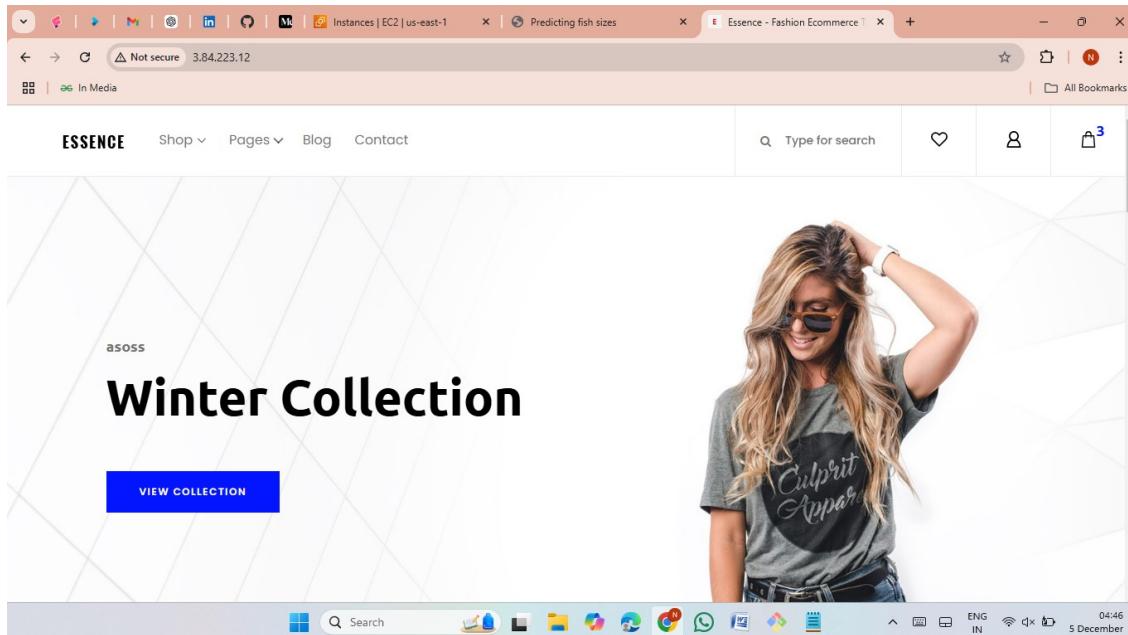
Below the table, it says "2 instances selected". The "Monitoring" tab is selected. At the bottom, there are buttons for CloudShell and Feedback, along with a search bar and a toolbar with various icons.

The screenshot shows the AWS VPC Dashboard. On the left, there's a sidebar with options like EC2 Global View, Filter by VPC, Virtual private cloud, Your VPCs, Subnets, Route tables, Internet gateways, Egress-only internet gateways, Carrier gateways, DHCP option sets, Elastic IPs, Managed prefix lists, and NAT gateways. The main area shows "Your VPCs (1/2) Info" with a table:

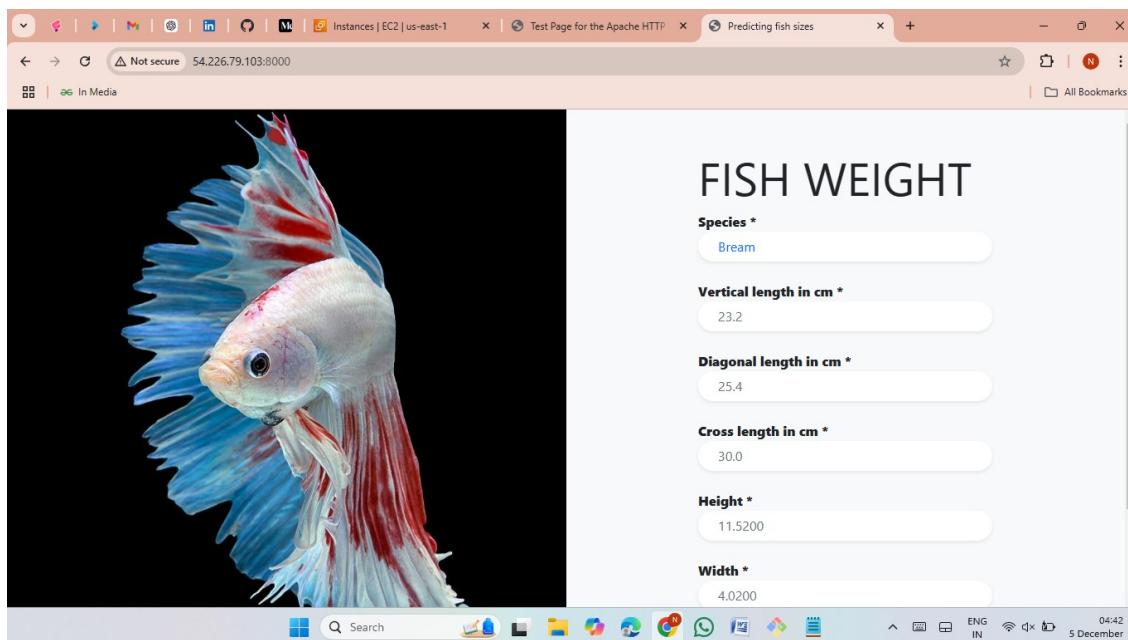
| Name     | VPC ID                | State     | Block Public... | IPv4 CIDR     |
|----------|-----------------------|-----------|-----------------|---------------|
| yash_vpc | vpc-06f48fe04ff57e1e8 | Available | Off             | 10.0.0/16     |
| -        | vpc-05f29c35f844e371  | Available | Off             | 172.31.0.0/16 |

Below the table, it says "vpc-06f48fe04ff57e1e8 / yash\_vpc". At the bottom, there are tabs for Details, Resource map, CIDs, Flow logs, Tags, and Integrations, along with a CloudShell and Feedback button, a search bar, and a toolbar with various icons.

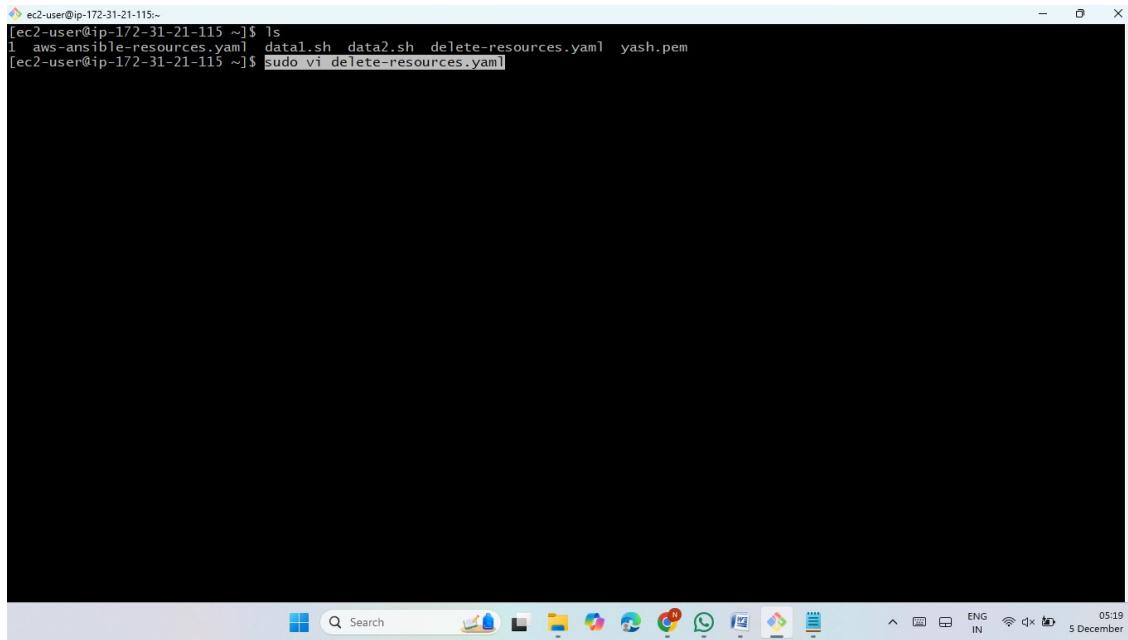
- Now copy the Instance 1 Public IP then Paste it on Google along with port 80.
- The static ecomm application was hosted successfully.



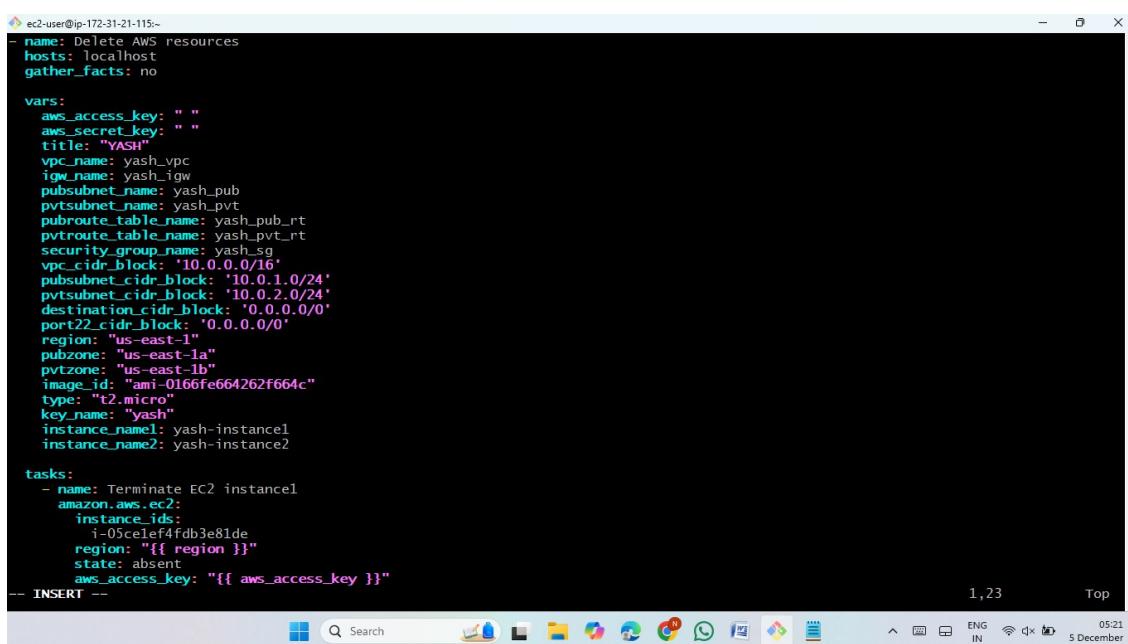
- Now copy the Instance2 Public IP then Paste it on Google along with port 8000.
- The dynamic python application was hosted successfully.



- 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:~$ ls
1 aws-ansible-resources.yaml data1.sh data2.sh delete-resources.yaml yash.pem
[ec2-user@ip-172-31-21-115 ~]$ sudo vi delete-resources.yaml
```

```
- ec2-user@ip-172-31-21-115:~
- name: Delete AWS resources
 hosts: localhost
 gather_facts: no

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:
 - name: Terminate EC2 instance1
 amazon.aws.ec2:
 instance_ids:
 - i-05ce1ef4fdb3e81de
 region: "{{ region }}"
 state: absent
 aws_access_key: "{{ aws_access_key }}"
-- INSERT --
```

```

ec2-user@ip-172-31-21-115:~$ tasks:
 - name: Terminate EC2 instance1
 amazon.aws.ec2:
 instance_ids:
 - i-03ce1ef4fdb3e81de
 region: "{{ region }}"
 state: absent
 aws_access_key: "{{ aws_access_key }}"
 aws_secret_key: "{{ aws_secret_key }}"

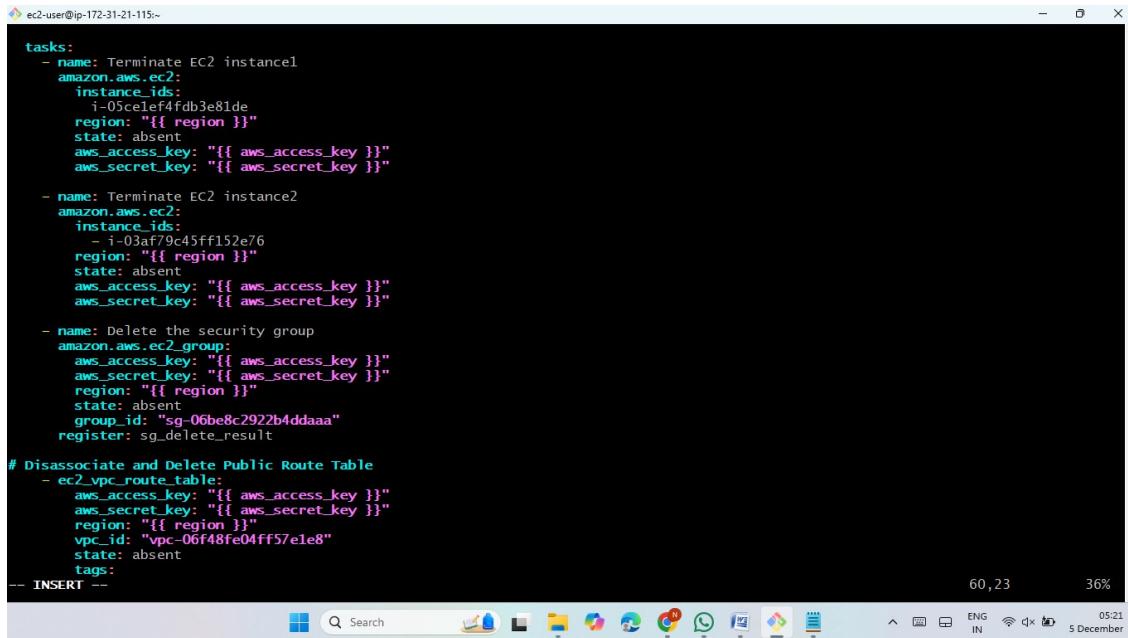
 - name: Terminate EC2 instance2
 amazon.aws.ec2:
 instance_ids:
 - i-03af79c45ff152e76
 region: "{{ region }}"
 state: absent
 aws_access_key: "{{ aws_access_key }}"
 aws_secret_key: "{{ aws_secret_key }}"

 - 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-06f48fe04ff57ele8"
 state: absent
 tags:
 - Name: "yash-pub-rt"

-- INSERT --

```

```

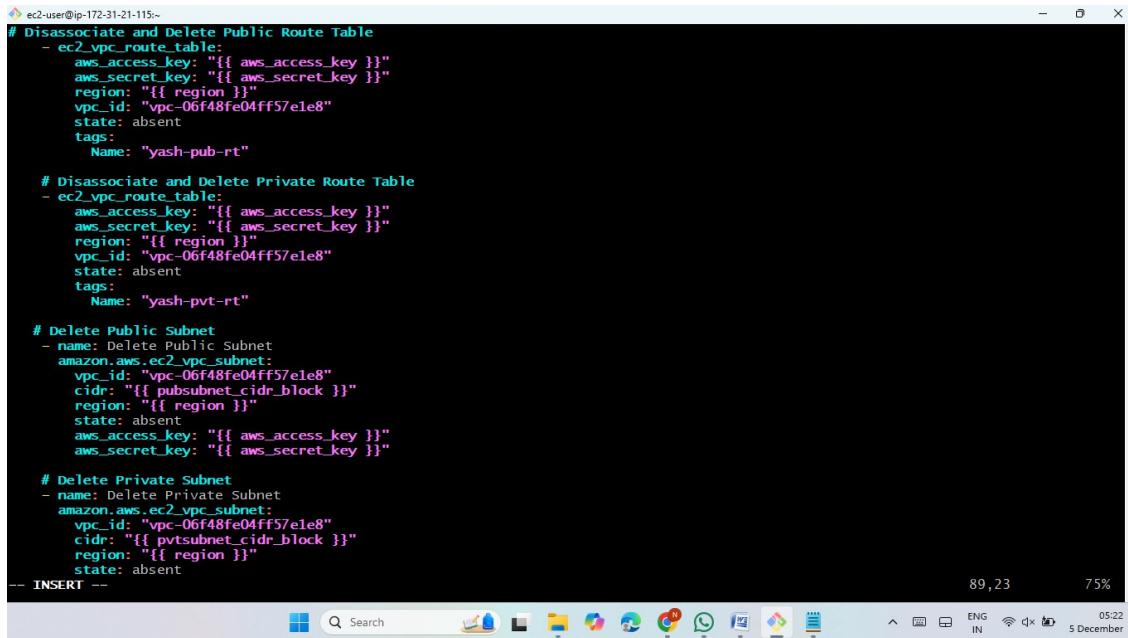
ec2-user@ip-172-31-21-115:~$ # Disassociate and Delete Public Route Table
 - ec2_vpc_route_table:
 aws_access_key: "{{ aws_access_key }}"
 aws_secret_key: "{{ aws_secret_key }}"
 region: "{{ region }}"
 vpc_id: "vpc-06f48fe04ff57ele8"
 state: absent
 tags:
 - Name: "yash-pub-rt"

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-06f48fe04ff57ele8"
 state: absent
 tags:
 - Name: "yash-pvt-rt"

Delete Public Subnet
 - name: Delete Public Subnet
 amazon.aws.ec2_vpc_subnet:
 vpc_id: "vpc-06f48fe04ff57ele8"
 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-06f48fe04ff57ele8"
 cidr: "{{ pvtsubnet_cidr_block }}"
 region: "{{ region }}"
 state: absent

```



```

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 --

```

```

ec2-user@ip-172-31-21-115:~$
PLAY RECAP ****
: ok=2 changed=0 unreachable=0 failed=1 skipped=0 rescued=0 ignored=0
[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 ****
: ok=9 changed=4 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0
[ec2-user@ip-172-31-21-115 ~]$

```

```

ec2-user@ip-172-31-21-115:~$ sudo vi delete-resources.yaml
TASK [ec2_vpc_route_table] *****
ok: [ansible]

TASK [ec2_vpc_route_table] *****
ok: [ansible]

TASK [ec2_vpc_net] *****
ok: [ansible]

PLAY RECAP *****
ansible : ok=3 changed=0 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0

[ec2-user@ip-172-31-21-115 ~]$ sudo 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 [ec2_vpc_route_table] *****
changed: [ansible]

TASK [ec2_vpc_route_table] *****
changed: [ansible]

TASK [ec2_vpc_net] *****
changed: [ansible]

PLAY RECAP *****
ansible : ok=3 changed=3 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0

[ec2-user@ip-172-31-21-115 ~]$ |

```

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

The screenshot shows the AWS EC2 Instances page. The left sidebar has sections for Dashboard, EC2 Global View, Events, Instances (with sub-options like Instances, Instance Types, Launch Templates, etc.), Images, and Elastic Block Store. The main content area displays a table titled "Instances (2/5)" with the following data:

| Name              | Instance ID         | Instance state | Instance type | Status check      | Alarm status                  | Available |
|-------------------|---------------------|----------------|---------------|-------------------|-------------------------------|-----------|
| yash-instance1    | i-0de486a3664fc7239 | Terminated     | t2.micro      | -                 | <a href="#">View alarms +</a> | us-ea     |
| yash-instance2    | i-048c3dd76dd835bbc | Terminated     | t2.micro      | -                 | <a href="#">View alarms +</a> | us-ea     |
| yash-instance2    | i-03af79c45ff152e76 | Terminated     | t2.micro      | -                 | <a href="#">View alarms +</a> | us-ea     |
| yash-inline-an... | i-0d27a1c9305b57844 | Running        | t2.micro      | 2/2 checks passed | <a href="#">View alarms +</a> | us-ea     |

Below the table, it says "2 instances selected". The "Monitoring" section is visible at the bottom, showing CPU utilization, Network in, Network out, and Network packets metrics. The footer includes copyright information and links for Privacy, Terms, and Cookie preferences.

➤ Here the VPC was deleted.

The screenshot shows the AWS VPC dashboard. On the left, there's a sidebar with options like EC2 Global View, Filter by VPC, Virtual private cloud, Your VPCs, Subnets, Route tables, Internet gateways, Egress-only internet gateways, Carrier gateways, DHCP option sets, Elastic IPs, Managed prefix lists, and NAT gateways. The main area is titled "Your VPCs (1) Info" and shows one VPC entry:

| Name | VPC ID               | State     | Block Public... | IPv4 CIDR     |
|------|----------------------|-----------|-----------------|---------------|
| -    | vpc-05f29c35f844e571 | Available | Off             | 172.31.0.0/16 |

Below the table, it says "Select a VPC above". At the bottom, there's a toolbar with CloudShell, Feedback, and other AWS services.

### METHOD-3 (OUTLINE)

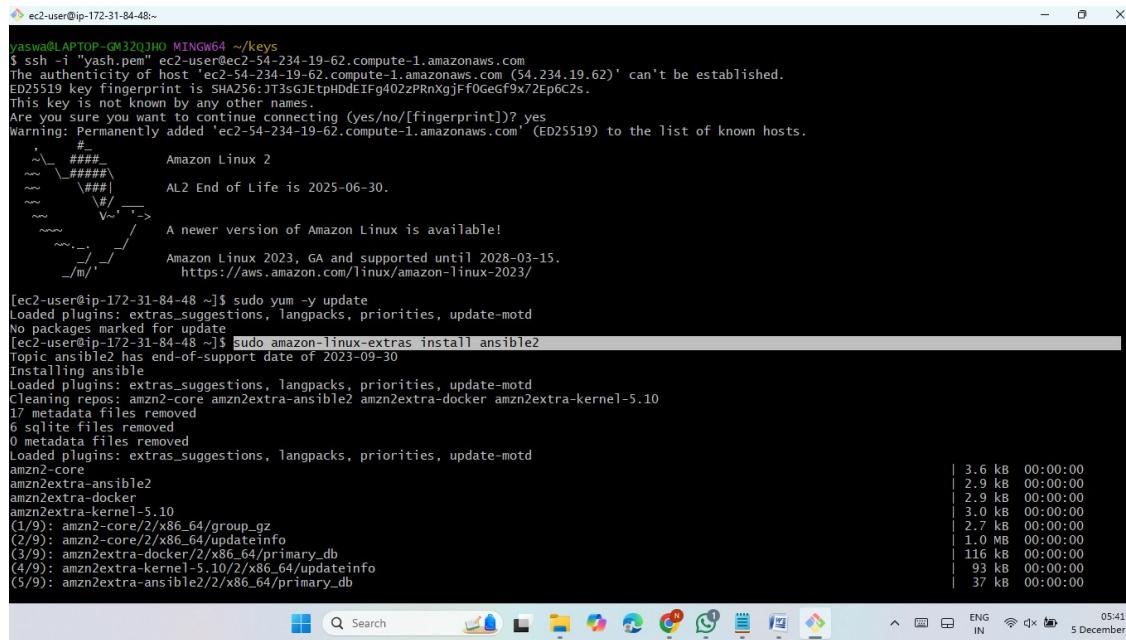
➤ Launch an Instances along with required ports.

The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with options like Dashboard, EC2 Global View, Events, Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images, AMIs, and AMI Catalog. The main area is titled "Instances (1/6) Info" and shows a list of instances:

| Name              | Instance ID         | Instance state | Instance type | Status check | Alarm status  | Avail. |
|-------------------|---------------------|----------------|---------------|--------------|---------------|--------|
| yash-outline-a... | i-05932d7317fd70494 | Running        | t2.micro      | Initializing | View alarms + | us-ea  |
| yash-instance1    | i-0de486a3664fc7239 | Terminated     | t2.micro      | -            | View alarms + | us-ea  |
| yash-instance2    | i-048c3dd76dd835bbc | Terminated     | t2.micro      | -            | View alarms + | us-ea  |
| yash-instance2    | i-03af79c45ff152e76 | Terminated     | t2.micro      | -            | View alarms + | us-ea  |

Below the table, it says "i-05932d7317fd70494 (yash-outline-ansible)". There are tabs for Details, Status and alarms, Monitoring, Security, Networking, Storage, and Tags. At the bottom, it shows Public IPv4 address (54.234.19.62) and Private IPv4 addresses (172.31.84.48). The bottom of the screen has a toolbar with CloudShell, Feedback, and other AWS services.

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

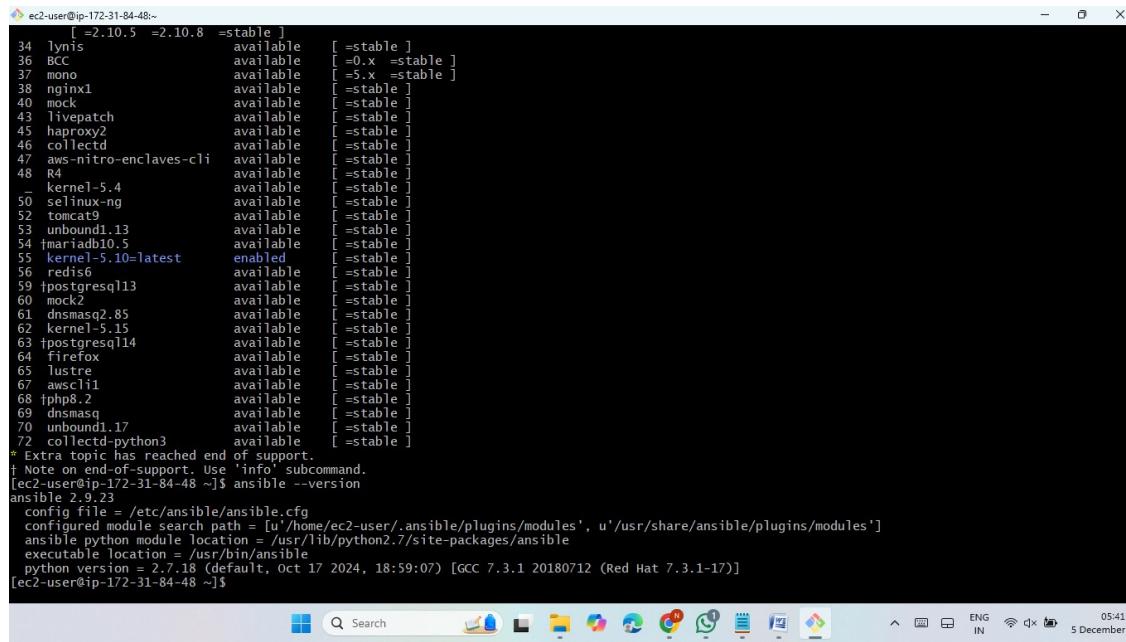


```
yasawa@LAPTOP-QM32QJHO MINGW64 ~/keys
$ ssh -1 "yash.pem" ec2-user@ec2-54-234-19-62.compute-1.amazonaws.com
The authenticity of host 'ec2-54-234-19-62.compute-1.amazonaws.com (54.234.19.62)' can't be established.
ED25519 key fingerprint is SHA256:JT3sGJEtpHDeIfg402zPRnxqjFFoGeGf9x/2Ep6C2s.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-54-234-19-62.compute-1.amazonaws.com' (ED25519) to the list of known hosts.

.
 ##
 ~\### Amazon Linux 2
 ~\### AL2 End of Life is 2025-06-30.
 ~\### A newer version of Amazon Linux is available!
 ~\### Amazon Linux 2023, GA and supported until 2028-03-15.
 https://aws.amazon.com/linux/amazon-linux-2023/

[ec2-user@ip-172-31-84-48 ~]$ sudo yum -y update
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
No packages marked for update
[ec2-user@ip-172-31-84-48 ~]$ sudo amazon-linux-extras install ansible2
Topic ansible2 has end-of-support date of 2023-09-30
Installing ansible
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Cleaning repos: amzn2-core amzn2extra-ansible2 amzn2extra-docker amzn2extra-kernel-5.10
1/7 metadata files removed
6 sqlite files removed
0 metadata files removed
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core
amzn2extra-ansible2
amzn2extra-docker
amzn2extra-kernel-5.10
(1/9): amzn2-core/2/x86_64/group_gz 3.6 kB 00:00:00
(2/9): amzn2-core/2/x86_64/updateinfo 2.9 kB 00:00:00
(3/9): amzn2extra-docker/2/x86_64/primary_db 2.9 kB 00:00:00
(4/9): amzn2extra-kernel-5.10/2/x86_64/updateinfo 3.0 kB 00:00:00
(5/9): amzn2extra-ansible2/2/x86_64/primary_db 2.7 kB 00:00:00
1.0 MB 00:00:00
116 kB 00:00:00
93 kB 00:00:00
37 kB 00:00:00

[ec2-user@ip-172-31-84-48 ~]$
```



```
[ec2-user@ip-172-31-84-48 ~]$ [=2.10.5 =2.10.8 =stable]
34 lynis available [=stable]
36 BCC available [=0.x =stable]
37 mono available [=5.x =stable]
38 nginx1 available [=stable]
40 mock available [=stable]
43 livepatch available [=stable]
45 haproxy2 available [=stable]
46 collected available [=stable]
47 aws-nitro-enclaves-cli available [=stable]
48 R4 available [=stable]
kernel-5.4 available [=stable]
50 selinux-ng available [=stable]
52 tomcat9 available [=stable]
53 unbound1.13 available [=stable]
54 mariadb10.5 available [=stable]
55 kernel-5.10-latest enabled [=stable]
56 redis6 available [=stable]
59 postgresql13 available [=stable]
60 mock2 available [=stable]
61 dnsmasq2.85 available [=stable]
62 kernel-5.15 available [=stable]
63 postgresql14 available [=stable]
64 firefox available [=stable]
65 lustre available [=stable]
67 awsccli1 available [=stable]
68 php8.2 available [=stable]
69 dnsmasq available [=stable]
70 unbound1.17 available [=stable]
72 collected-python3 available [=stable]
* Extra topic has reached end of support.
† Note on end-of-support: Use 'info' subcommand.
[ec2-user@ip-172-31-84-48 ~]$ ansible --version
ansible 2.9.23
 config file = /etc/ansible/ansible.cfg
 configured module search path = ['~/home/ec2-user/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
 ansible python module location = /usr/lib/python2.7/site-packages/ansible
 executable location = /usr/bin/ansible
 python version = 2.7.18 (default, Oct 17 2024, 18:59:07) [GCC 7.3.1 20180712 (Red Hat 7.3.1-17)]
[ec2-user@ip-172-31-84-48 ~]$
```

- 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
```

```
ec2-user@ip-172-31-84-48:/etc/ansible
43 livepatch available [=stable]
45 haproxy2 available [=stable]
46 collectd available [=stable]
47 aws-nitro-enclaves-cli available [=stable]
48 R4 available [=stable]
kernel-5.4 available [=stable]
50 selinux-ng available [=stable]
52 tomcat9 available [=stable]
53 unbound1.13 available [=stable]
54 mariadb10.5 available [=stable]
55 kernel-5.10-latest enabled [=stable]
56 redis6 available [=stable]
59 psycopgql13 available [=stable]
60 mock2 available [=stable]
61 dnsmasq2.85 available [=stable]
62 kernel-5.15 available [=stable]
63 psycopgql14 available [=stable]
64 firefox available [=stable]
65 lustre available [=stable]
67 awsccli1 available [=stable]
68 tphp8.2 available [=stable]
69 dnsmasq available [=stable]
70 unbound1.17 available [=stable]
72 collectd-python3 available [=stable]
* Extra topic has reached end of support.
† Note on end-of-support. Use 'info' subcommand.
[ec2-user@ip-172-31-84-48 ~]$ ansible --version
ansible 2.9.23
 config file = /etc/ansible/ansible.cfg
 configured module search path = ['/home/ec2-user/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
 ansible python module location = /usr/lib/python2.7/site-packages/ansible
 executable location = /usr/bin/ansible
 python version = 2.7.18 (default, Oct 17 2024, 18:59:07) [GCC 7.3.1 20180712 (Red Hat 7.3.1-17)]
[ec2-user@ip-172-31-84-48 ~]$ cd /etc/ansible/
[ec2-user@ip-172-31-84-48 ansible]$ ls
ansible.cfg hosts roles
[ec2-user@ip-172-31-84-48 ansible]$ sudo vi hosts
[ec2-user@ip-172-31-84-48 ansible]$ pwd
/etc/ansible
[ec2-user@ip-172-31-84-48 ansible]$ |
```

```
[localhost]
ansible ansible_host=172.31.84.48 ansible_python_interpreter=/usr/bin/python3 ansible_user=ec2-user ansible_ssh_private_key_file=/home/ec2-user/yash.pem

This is the default ansible 'hosts' file.
#
It should live in /etc/ansible/hosts
#
- Comments begin with the '#' character
- Blank lines are ignored
- Groups of hosts are delimited by [header] elements
- You can enter hostnames or ip addresses
- A hostname/ip can be a member of multiple groups

Ex 1: Ungrouped hosts, specify before any group headers.

green.example.com
blue.example.com
192.168.100.1
192.168.100.10

Ex 2: A collection of hosts belonging to the 'webservers' group

[webservers]
alpha.example.org
beta.example.org
192.168.1.100
192.168.1.110

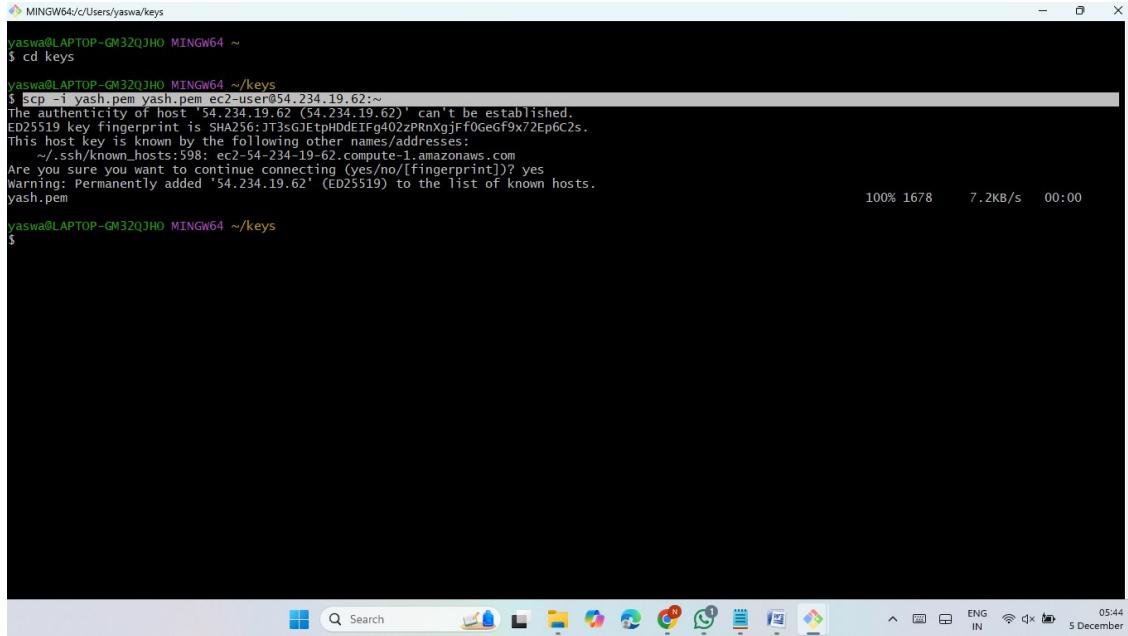
If you have multiple hosts following a pattern you can specify
them like this:

www[001:006].example.com

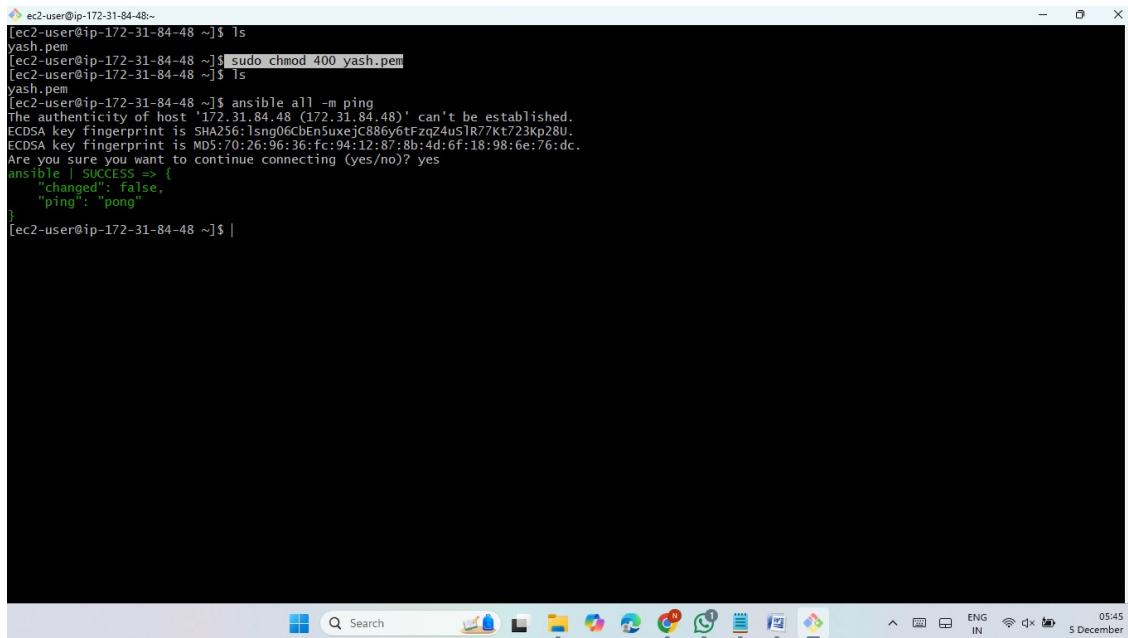
Ex 3: A collection of database servers in the 'dbservers' group

[dbservers]
db01.intranet.mydomain.net
db02.intranet.mydomain.net
-- INSERT --
```

➤ Now open another terminal then copy the private key from local to instances.



```
MINGW64/c/Users/yaswa/keys
yaswa@LAPTOP-GM32QJHO MINGW64 ~
$ cd keys
yaswa@LAPTOP-GM32QJHO MINGW64 ~/keys
$ scp -i yash.pem yash.pem ec2-user@54.234.19.62:~
The authenticity of host '54.234.19.62 (54.234.19.62)' can't be established.
ED25519 key fingerprint is SHA256:JT3sGJEtpHDeIfFg402zPrnxjfF0GeGf9x72Ep6C2s.
This host key is known by the following other names/addresses:
 ~/.ssh/known_hosts:598: ec2-54-234-19-62.compute-1.amazonaws.com
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '54.234.19.62' (ED25519) to the list of known hosts.
yash.pem
100% 1678 7.2KB/s 00:00
yaswa@LAPTOP-GM32QJHO MINGW64 ~/keys
$
```



```
ec2-user@ip-172-31-84-48:~
[ec2-user@ip-172-31-84-48 ~]$ ls
yash.pem
[ec2-user@ip-172-31-84-48 ~]$ sudo chmod 400 yash.pem
[ec2-user@ip-172-31-84-48 ~]$ ls
yash.pem
[ec2-user@ip-172-31-84-48 ~]$ ansible all -m ping
The authenticity of host '172.31.84.48 (172.31.84.48)' can't be established.
ED25519 key fingerprint is SHA256:lsng06CbEn5uxejc886y6tFzqZ4u5lr77Kt723kp28U.
Are you sure you want to continue connecting (yes/no)? yes
ansible | SUCCESS => {
 "changed": false,
 "ping": "pong"
}
[ec2-user@ip-172-31-84-48 ~]$ |
```

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

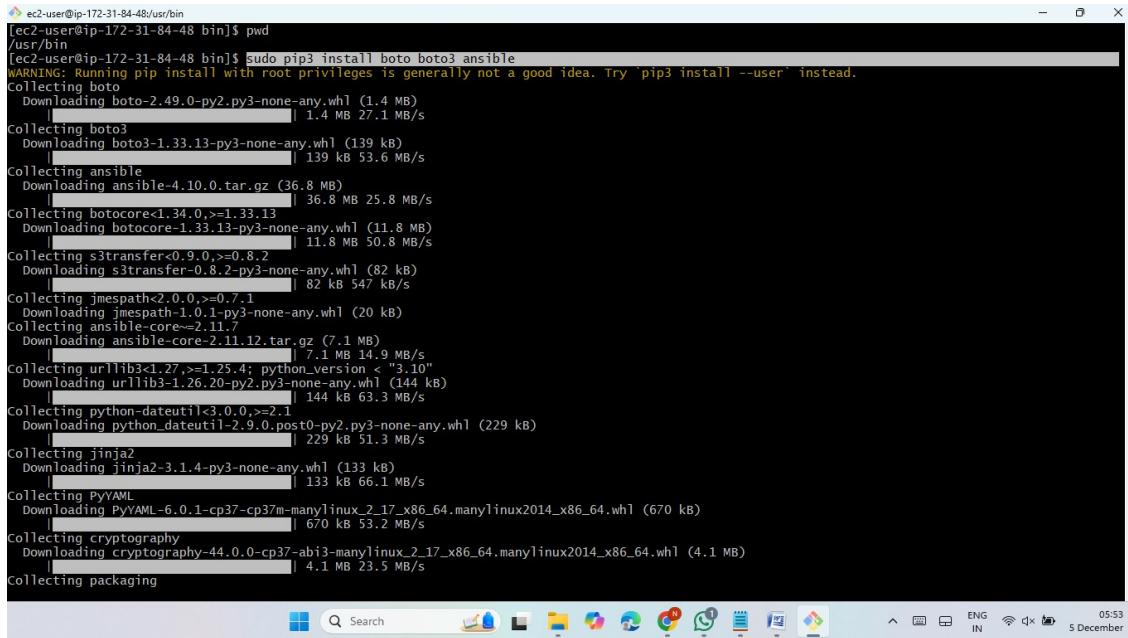
```
ec2-user@ip-172-31-84-48:/etc/ansible
[ec2-user@ip-172-31-84-48 ~]$ cd /etc/ansible/
[ec2-user@ip-172-31-84-48 ansible]$ ls
ansible.cfg hosts roles
[ec2-user@ip-172-31-84-48 ansible]$ sudo vi ansible.cfg
[ec2-user@ip-172-31-84-48 ansible]$ pwd
/etc/ansible
[ec2-user@ip-172-31-84-48 ansible]$ |
```

The screenshot shows a terminal window with a black background and white text. It displays a series of commands being run at a shell prompt. The user is navigating to the `/etc/ansible` directory, listing its contents, and then opening the `ansible.cfg` file for editing using the `vi` text editor. The terminal window has a standard Windows-style title bar and a taskbar at the bottom with various icons.

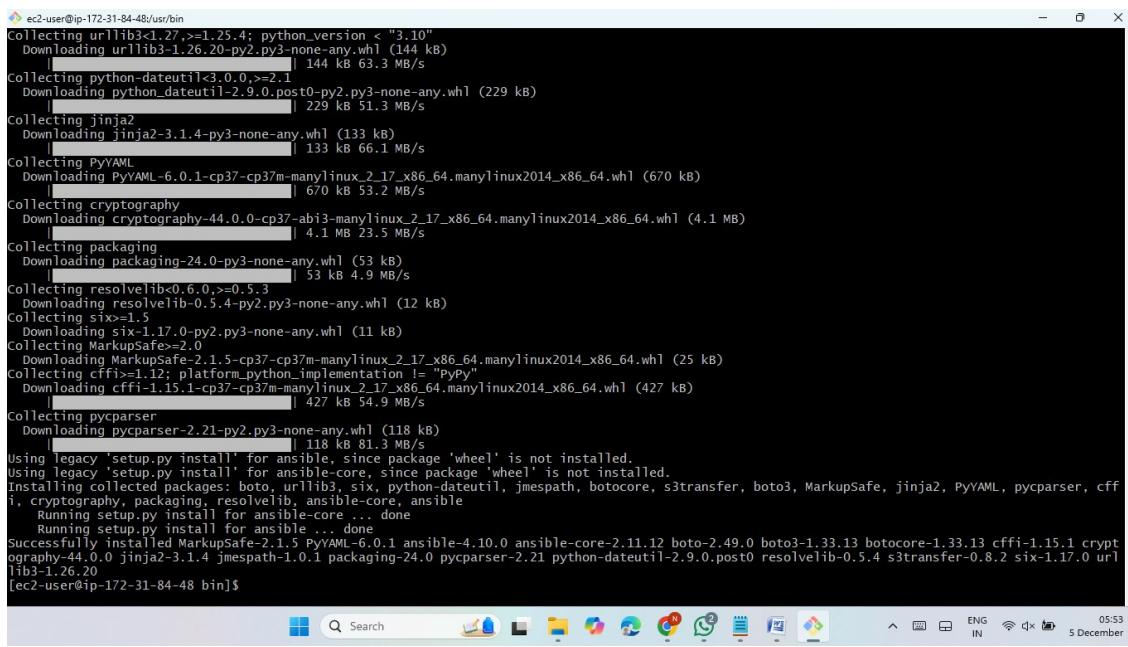
```
39 # by default gathering retrieves all facts subsets
40 # all - gather all subsets
41 # network - gather min and network facts
42 # hardware - gather hardware facts (longest facts to retrieve)
43 # virtual - gather min and virtual facts
44 # facter - import facts from facter
45 # ohai - import facts from ohai
46 # You can combine them using comma (ex: network,virtual)
47 # You can negate them using ! (ex: !hardware,!facter,!ohai)
48 # A minimal set of facts is always gathered.
49 #gather_subset = all
50
51 # some hardware related facts are collected
52 # with a maximum timeout of 10 seconds. This
53 # option lets you increase or decrease that
54 # timeout to something more suitable for the
55 # environment.
56 # gather_timeout = 10
57
58 # Ansible facts are available inside the ansible_facts.* dictionary
59 # namespace. This setting maintains the behaviour which was the default prior
60 # to 2.5, duplicating these variables into the main namespace, each with a
61 # prefix of 'ansible'.
62 # This variable is set to True by default for backwards compatibility. It
63 # will be changed to a default of 'False' in a future release.
64 # ansible_facts.
65 # inject_facts_as_vars = True
66
67 # additional paths to search for roles in, colon separated
68 #roles_path = /etc/ansible/roles
69
70 # uncomment this to disable SSH key host checking
71 host_key_checking = False
72
73 # change the default callback, you can only have one 'stdout' type enabled at a time.
74 #stdout_callback = skippy
75
76
77 ## Ansible ships with some plugins that require whitelisting,
-- INSERT --
```

The screenshot shows a terminal window with a black background and white text. It displays the content of the `ansible.cfg` file. The 71st line, `host\_key\_checking = False`, is highlighted in red, indicating it has been uncommented. The terminal window has a standard Windows-style title bar and a taskbar at the bottom with various icons.

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



```
ec2-user@ip-172-31-84-48:/usr/bin$ pwd
/usr/bin
[ec2-user@ip-172-31-84-48 bin]$ sudo pip3 install boto boto3 ansible
WARNING: Running pip install with root privileges is generally not a good idea. Try 'pip3 install --user' instead.
Collecting boto
 Downloading boto-2.49.0-py2.py3-none-any.whl (1.4 MB)
 1.4 MB 27.1 MB/s
Collecting boto3
 Downloading boto3-1.33.13-py3-none-any.whl (139 kB)
 139 kB 53.6 MB/s
Collecting ansible
 Downloading ansible-4.10.0.tar.gz (36.8 MB)
 36.8 MB 25.8 MB/s
Collecting botocore<1.34.0,>=1.33.13
 Downloading botocore-1.33.13-py3-none-any.whl (11.8 MB)
 11.8 MB 50.8 MB/s
Collecting s3transfer<0.9.0,>=0.8.2
 Downloading s3transfer-0.8.2-py3-none-any.whl (82 kB)
 82 kB 547 kB/s
Collecting jmespath<2.0.0,>=0.7.1
 Downloading jmespath-1.0.1-py3-none-any.whl (20 kB)
Collecting ansible-core<2.11.7
 Downloading ansible-core-2.11.12.tar.gz (7.1 MB)
 7.1 MB 14.9 MB/s
Collecting urllib3<1.27,>=1.25.4; python_version < "3.10"
 Downloading urllib3-1.26.20-py2.py3-none-any.whl (144 kB)
 144 kB 63.3 MB/s
Collecting python-dateutil<3.0.0,>=2.1
 Downloading python_dateutil-2.9.0.post0-py2.py3-none-any.whl (229 kB)
 229 kB 51.3 MB/s
Collecting jinja2
 Downloading jinja2-3.1.4-py3-none-any.whl (133 kB)
 133 kB 66.1 MB/s
Collecting PyYAML
 Downloading PyYAML-6.0.1-cp37-cp37m-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (670 kB)
 670 kB 53.2 MB/s
Collecting cryptography
 Downloading cryptography-44.0.0-cp37abi3-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (4.1 MB)
 4.1 MB 23.5 MB/s
collecting packaging
```



```
ec2-user@ip-172-31-84-48:/usr/bin
Collecting urllib3<1.27,>=1.25.4; python_version < "3.10"
 Downloading urllib3-1.26.20-py2.py3-none-any.whl (144 kB)
 144 kB 63.3 MB/s
Collecting python-dateutil<3.0.0,>=2.1
 Downloading python_dateutil-2.9.0.post0-py2.py3-none-any.whl (229 kB)
 229 kB 51.3 MB/s
Collecting jinja2
 Downloading jinja2-3.1.4-py3-none-any.whl (133 kB)
 133 kB 66.1 MB/s
Collecting PyYAML
 Downloading PyYAML-6.0.1-cp37-cp37m-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (670 kB)
 670 kB 53.2 MB/s
Collecting cryptography
 Downloading cryptography-44.0.0-cp37abi3-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (4.1 MB)
 4.1 MB 23.5 MB/s
Collecting packaging
 Downloading packaging-24.0-py3-none-any.whl (53 kB)
 53 kB 4.9 MB/s
Collecting resolvelib<0.6.0,>=0.5.3
 Downloading resolvelib-0.5.4-py2.py3-none-any.whl (12 kB)
Collecting six<1.5
 Downloading six-1.17.0-py2.py3-none-any.whl (11 kB)
Collecting MarkupSafe>=2.0
 Downloading MarkupSafe-2.1.5-cp37-cp37m-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (25 kB)
Collecting cffi>1.12; platform_python_implementation != "PyPy"
 Downloading cffi-1.15.1-cp37-cp37m-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (427 kB)
 427 kB 54.9 MB/s
Collecting pycparser
 Downloading pycparser-2.21-py2.py3-none-any.whl (118 kB)
 118 kB 81.3 MB/s
Using Legacy 'setup.py install' for ansible, since package 'wheel' is not installed.
Using legacy 'setup.py install' for ansible-core, since package 'wheel' is not installed.
Installing collected packages: boto, urllib3, six, python-dateutil, jmespath, botocore, s3transfer, boto3, MarkupSafe, jinja2, PyYAML, pycparser, cffi, cryptography, packaging, resolvelib, ansible-core, ansible
 Running setup.py install for ansible-core ... done
 Running setup.py install for ansible ... done
Successfully installed MarkupSafe-2.1.5 PyYAML-6.0.1 ansible-4.10.0 ansible-core-2.11.12 boto-2.49.0 boto3-1.33.13 botocore-1.33.13 cffi-1.15.1 cryptography-44.0.0 jinja2-3.1.4 jmespath-1.0.1 packaging-24.0 pycparser-2.21 python-dateutil-2.9.0.post0 resolvelib-0.5.4 s3transfer-0.8.2 six-1.17.0 urllib3-1.26.20
[ec2-user@ip-172-31-84-48 bin]$
```

- 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

```
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% 02:30 8 December

```
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%
02:31 8 December
Windows Taskbar icons: File Explorer, OneDrive, Mail, Photos, Snipping Tool, Task View, Task Manager, File History, Control Panel, File Explorer, OneDrive, Mail, Photos, Snipping Tool, Task View, Task Manager, File History, Control Panel
```

```
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', 'data.sh') }}"
 instance_tags:
 Name: "{{ instance_name1 }}"

Launch EC2 Instance 2
- ec2:
-- INSERT --
105,10 82%
02:31 8 December
Windows Taskbar icons: File Explorer, OneDrive, Mail, Photos, Snipping Tool, Task View, Task Manager, File History, Control Panel, File Explorer, OneDrive, Mail, Photos, Snipping Tool, Task View, Task Manager, File History, Control Panel
```

```
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 --
```

```
es2-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/GUSERABBANI44/ecomm.git
sudo mv ecomm/* /var/www/html/

~
-- INSERT --
```

```
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
```

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

```

[ec2-user@ip-172-31-84-48 ~]$ sudo vi secret.yaml
[ec2-user@ip-172-31-84-48 ~]$ ansible-playbook aws-resources-ansible.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 depreciation_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 [localhost] ****
TASK [ec2_vpc_net] ****
changed: [ansible]

TASK [ec2_vpc_igw] ****
changed: [ansible]

TASK [ec2_vpc_subnet] ****
changed: [ansible]

TASK [ec2_vpc_subnet] ****
changed: [ansible]

TASK [ec2_vpc_route_table] ****
changed: [ansible]

TASK [ec2_group] ****
changed: [ansible]

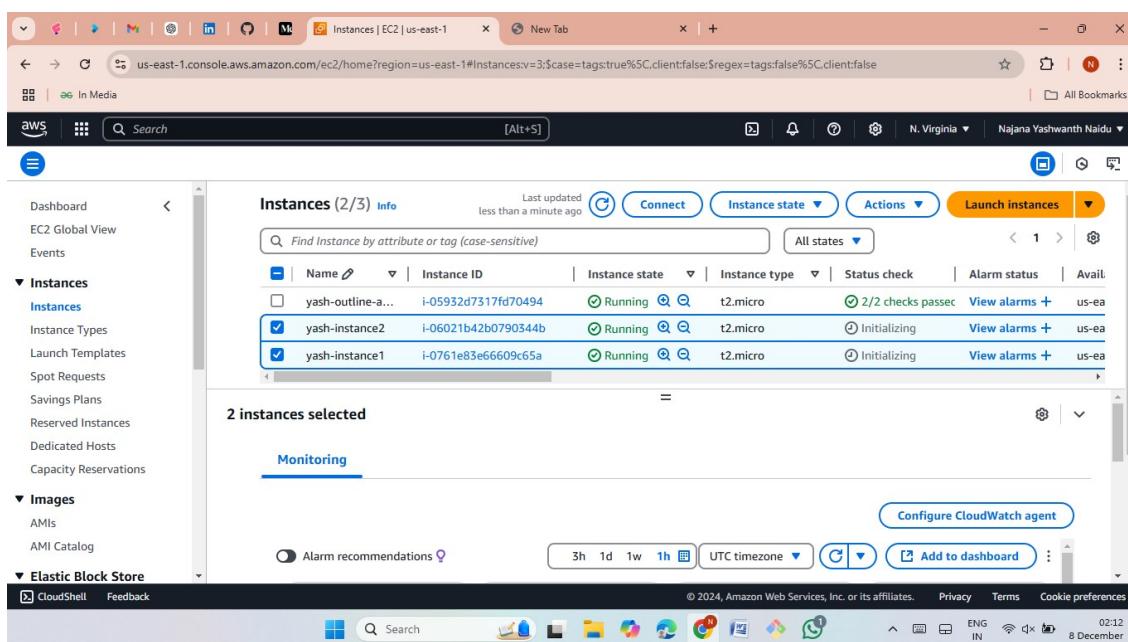
TASK [ec2] ****
changed: [ansible]

TASK [ec2] ****
changed: [ansible]

PLAY RECAP ****
ansible : ok=8 changed=8 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0
[ec2-user@ip-172-31-84-48 ~]$ |

```

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



- Here the VPC was created.

The screenshot shows the AWS VPC console interface. On the left, there's a sidebar with options like EC2 Global View, Filter by VPC, Virtual private cloud, Your VPCs, Subnets, Route tables, Internet gateways, Egress-only internet gateways, Carrier gateways, DHCP option sets, Elastic IPs, Managed prefix lists, and NAT gateways. The main area is titled "Your VPCs (1/2) Info" and shows a table with two rows:

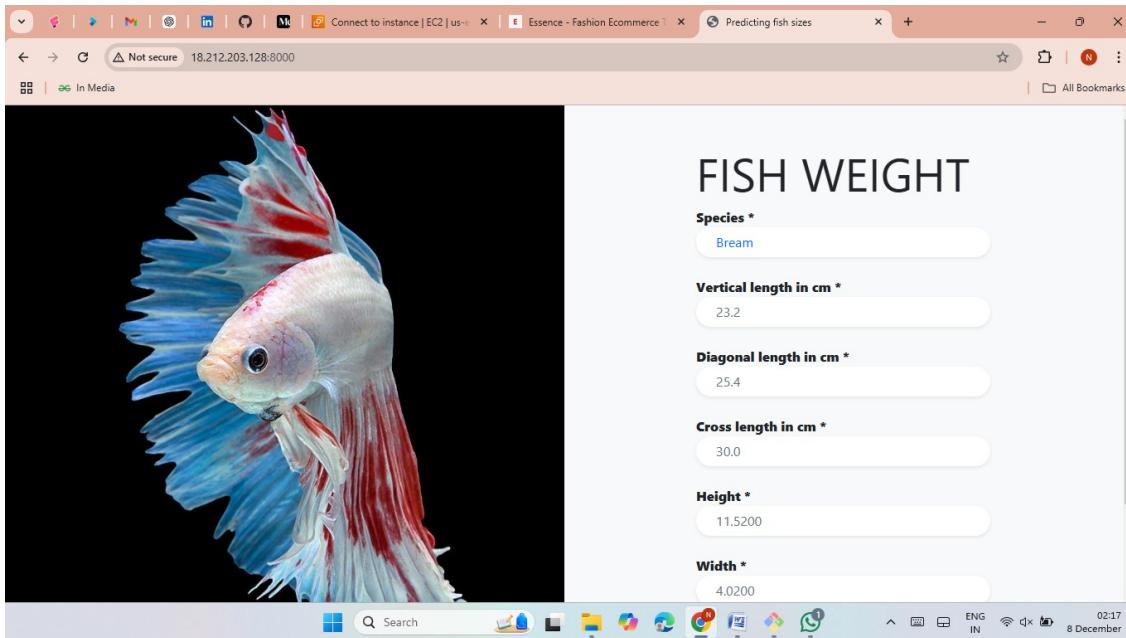
| Name     | VPC ID                | State     | Block Public... | IPv4 CIDR     |
|----------|-----------------------|-----------|-----------------|---------------|
| yash_vpc | vpc-01633278668de7905 | Available | Off             | 10.0.0.0/16   |
| -        | vpc-05f2f9c35f844e371 | Available | Off             | 172.31.0.0/16 |

Below the table, there's a section for "vpc-01633278668de7905 / yash\_vpc" with tabs for Details, Resource map, CIDRs, Flow logs, Tags, and Integrations. The status bar at the bottom indicates it's from December 8, 2024.

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

The screenshot shows a web browser window displaying the "Essence - Fashion Ecommerce" website. The URL in the address bar is "54.226.51.113". The page features a large image of a woman wearing sunglasses and a t-shirt, with the text "Winter Collection" overlaid. Below the image is a blue button labeled "VIEW COLLECTION". The top navigation bar includes links for "ESSENCE", "Shop", "Pages", "Blog", and "Contact". A search bar and user icons are also present. The status bar at the bottom shows the date as December 8, 2024.

- Now copy the Instance2 Public IP then Paste 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:~$ cat delete_ec2_instances_and_sg.yml
- hosts: localhost
 become: yes
 gather_facts: false
 vars_files:
 - secret.yaml

 tasks:
 - name: Terminate EC2 instance1
 amazon.aws.ec2:
 instance_ids:
 - i-0761e83e66609c65a
 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-06021b42b0790344b
 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-020bcae1e8ff031ad3"
 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 }}"
-- INSERT --
```

```
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 --
```

```
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 --
```

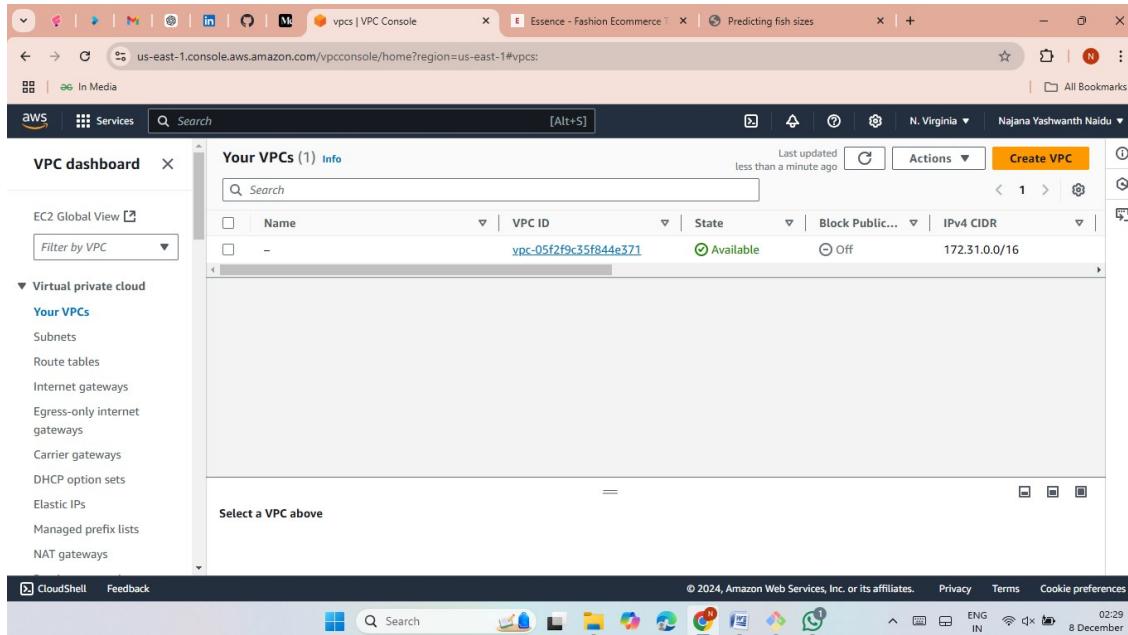
- Now run the file with the command **ansible-playbook <name of the file>**.

```

[ec2-user@ip-172-31-84-48 ~]$ 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: 2.7.16 (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 [localhost] ****
TASK [Terminate EC2 instance1] ****
ok: [ansible]
TASK [Terminate EC2 instance2] ****
ok: [ansible]
TASK [Delete the security group] ****
changed: [ansible]
TASK [ec2_vpc_route_table] ****
changed: [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] ****
changed: [ansible]
PLAY RECAP ****
ansible : ok=9 changed=6 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0
[ec2-user@ip-172-31-84-48 ~]$

```

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



The screenshot shows the AWS EC2 Instances page. On the left, a sidebar menu includes options like Dashboard, EC2 Global View, Events, Instances (selected), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images, AMIs, AMI Catalog, and Elastic Block Store. The main content area displays a table titled "Instances (2/3) Info" with the following data:

| Name                                               | Instance ID         | Instance state | Instance type | Status check      | Alarm status  | Available |
|----------------------------------------------------|---------------------|----------------|---------------|-------------------|---------------|-----------|
| yash-outline-a...                                  | i-05932d7317fd70494 | Running        | t2.micro      | 2/2 checks passed | View alarms + | us-ea     |
| <input checked="" type="checkbox"/> yash-instance2 | i-06021b42b0790344b | Terminated     | t2.micro      | -                 | View alarms + | us-ea     |
| <input checked="" type="checkbox"/> yash-instance1 | i-0761e83e66609c65a | Terminated     | t2.micro      | -                 | View alarms + | us-ea     |

Below the table, a message says "2 instances selected". A "Monitoring" tab is visible above a chart area. The bottom of the screen shows the Windows taskbar with various pinned icons.