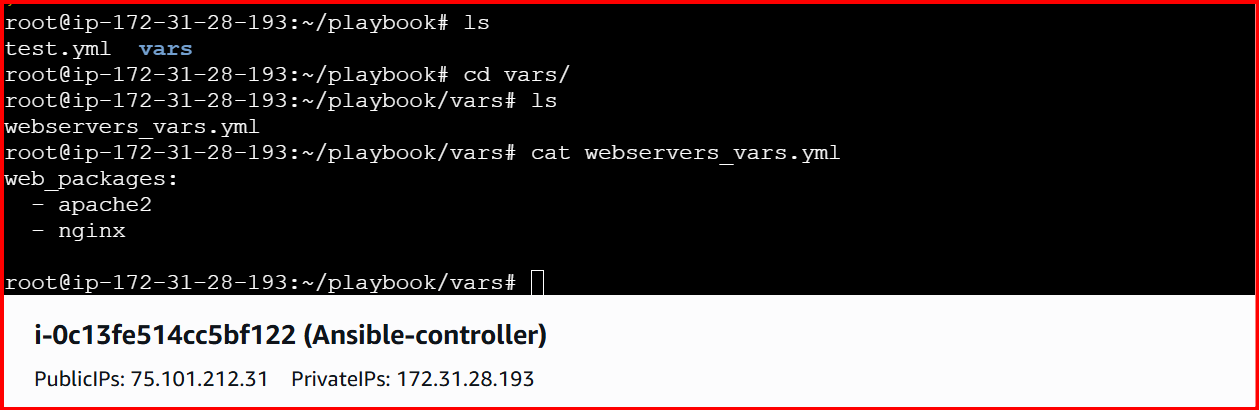
**Ansible Task-3**

1. **Write a single ansible playbook which will install apache and nginx. Note: Playbook should not be hardcoded and pass the variables from different file.**

1) Set the ansible controller



1. Create playbook
2. Create yml file 🡪 test.yml
3. mkdir -p vars
4. vi vars/webservers\_vars.yml



webservers\_vars.yml

web\_packages:

- httpd

- nginx

test.yml

---

- name: Install multiple web servers (apache, nginx) dynamically

hosts: all

become: yes

vars\_files:

- ./vars/webservers\_vars.yml

tasks:

- name: Ensure package cache is up-to-date (Debian/Ubuntu)

apt:

update\_cache: yes

when: ansible\_facts['pkg\_mgr'] == "apt"

- name: Ensure package cache is up-to-date (RHEL/CentOS)

yum:

update\_cache: yes

when: ansible\_facts['pkg\_mgr'] == "yum"

- name: Install required packages

package:

name: "{{ item }}"

state: present

loop: "{{ web\_packages }}"

register: install\_result

- name: Show install results

debug:

var: install\_result

- name: Create Apache index page

copy:

dest: /var/www/html/index.html

content: "<h1>This is Apache2 page</h1>"

when: "'apache2' in web\_packages or 'httpd' in web\_packages"

- name: Create Nginx index page

copy:

dest: /usr/share/nginx/html/index.html

content: "<h1>This is Nginx page</h1>"

when: "'nginx' in web\_packages"

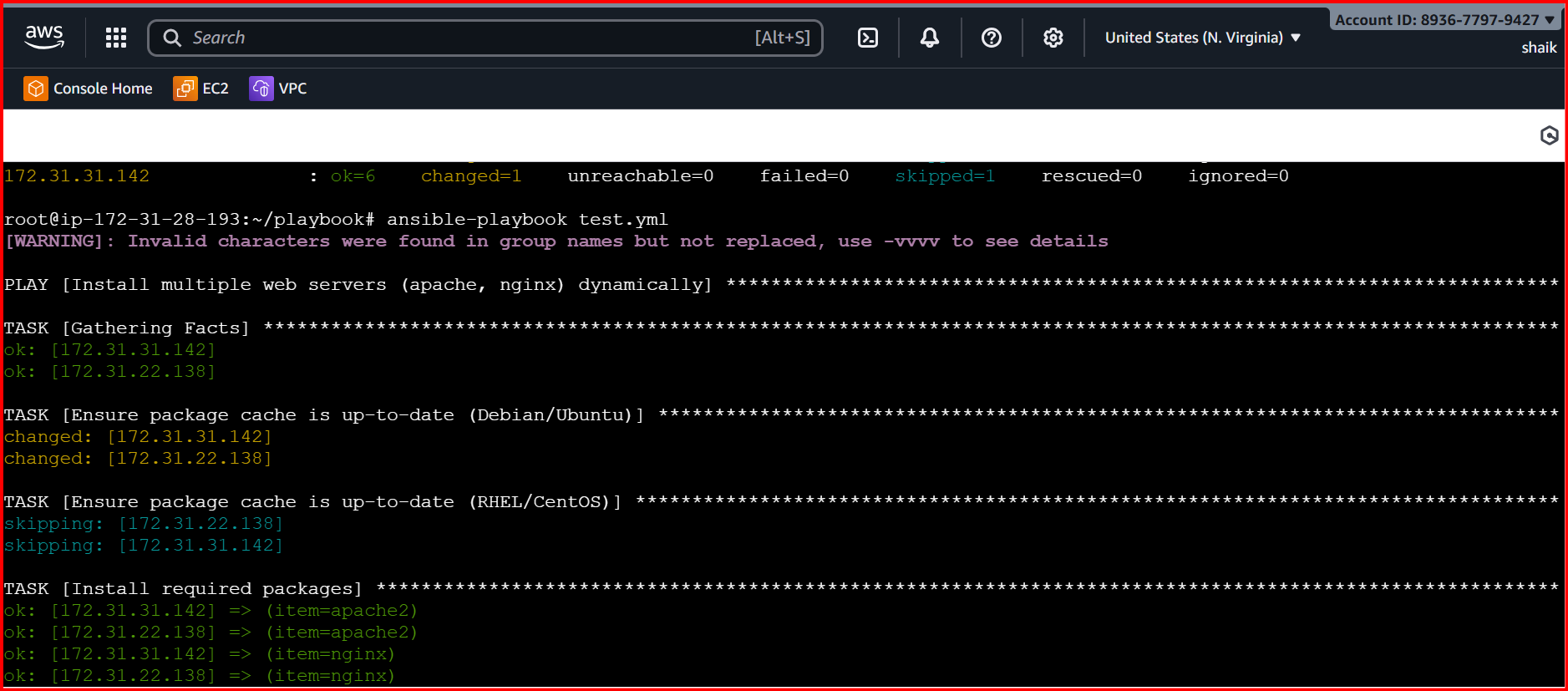
---------------------------------------------------------------------------------------------------

Run these commands

ansible-playbook test.yml --syntax-check

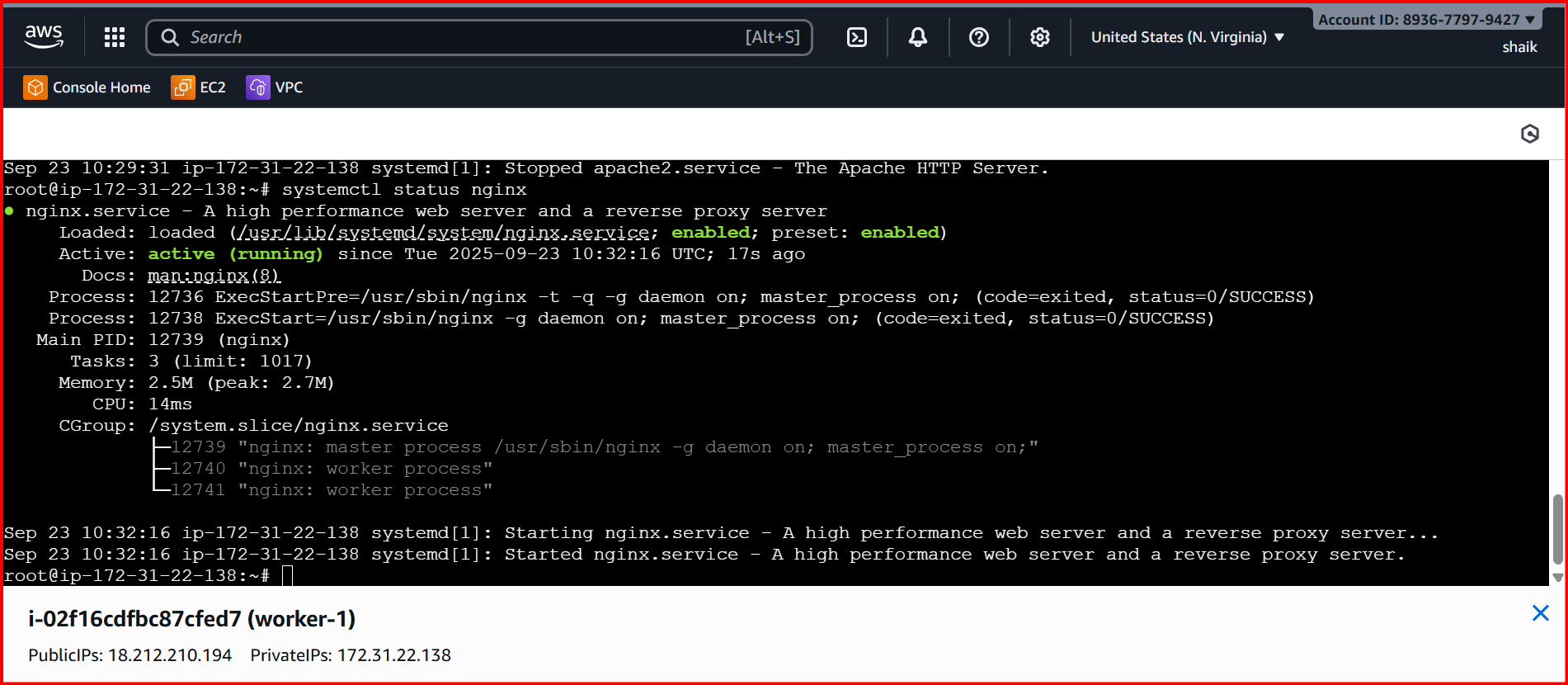
ansible-playbook test.yml --check

ansible-playbook test.yml



**ansible all -i /etc/ansible/hosts -b -m service -a "name=apache2 state=stopped"**

**ansible all -i /etc/ansible/hosts -b -m service -a "name=nginx state=started"**





**2. Ansible playbook to create 10 different directories with minimal code and directory names should be passed as variables.**

Create

create\_dirs.yml

---

- name: Create 10 directories with loop

hosts: all

become: yes

vars:

dir\_list:

- /opt/dir1

- /opt/dir2

- /opt/dir3

- /opt/dir4

- /opt/dir5

- /opt/dir6

- /opt/dir7

- /opt/dir8

- /opt/dir9

- /opt/dir10

tasks:

- name: Create directories

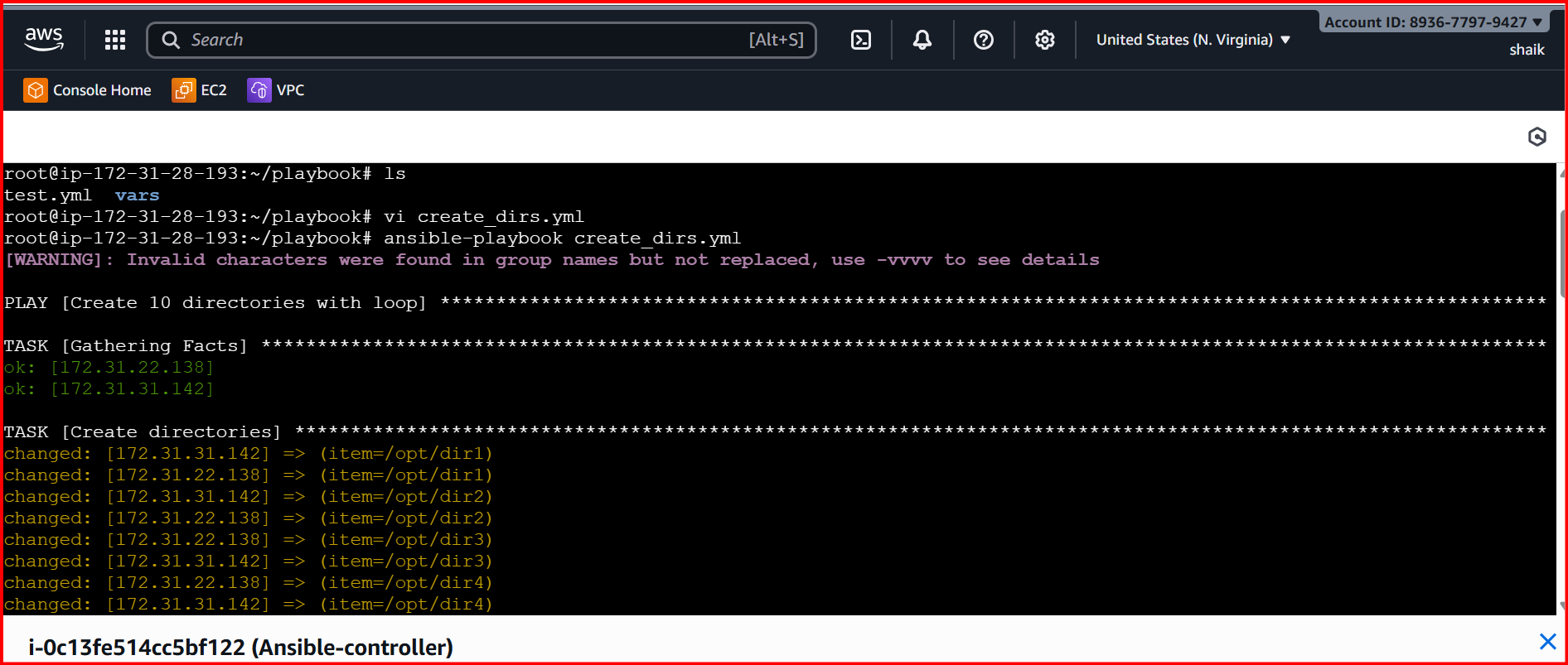
file:

path: "{{ item }}"

state: directory

mode: '0755'

loop: "{{ dir\_list }}"



Check it worker servers



**4.Ansible playbook to inject ansible vault variables**

**Run this command**

**ansible-vault create vault.yml**

**set new password**

**create one yml**

**vi use\_vault.yml**

**---**

**- name: Use Ansible Vault variables**

**hosts: all**

**become: yes**

**vars\_files:**

**- vault.yml # Inject vault file**

**tasks:**

**- name: Show DB username**

**debug:**

**msg: "Database user is {{ vault\_db\_user }}"**

**- name: Show DB password**

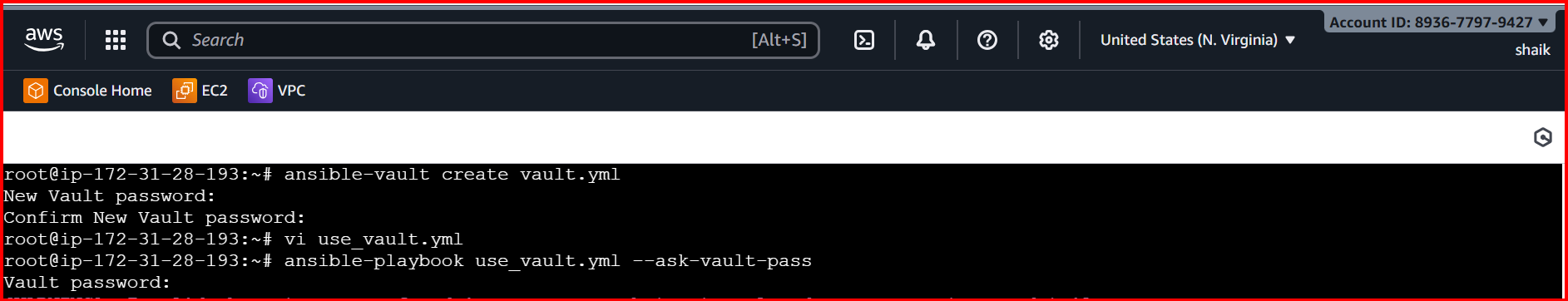
**debug:**

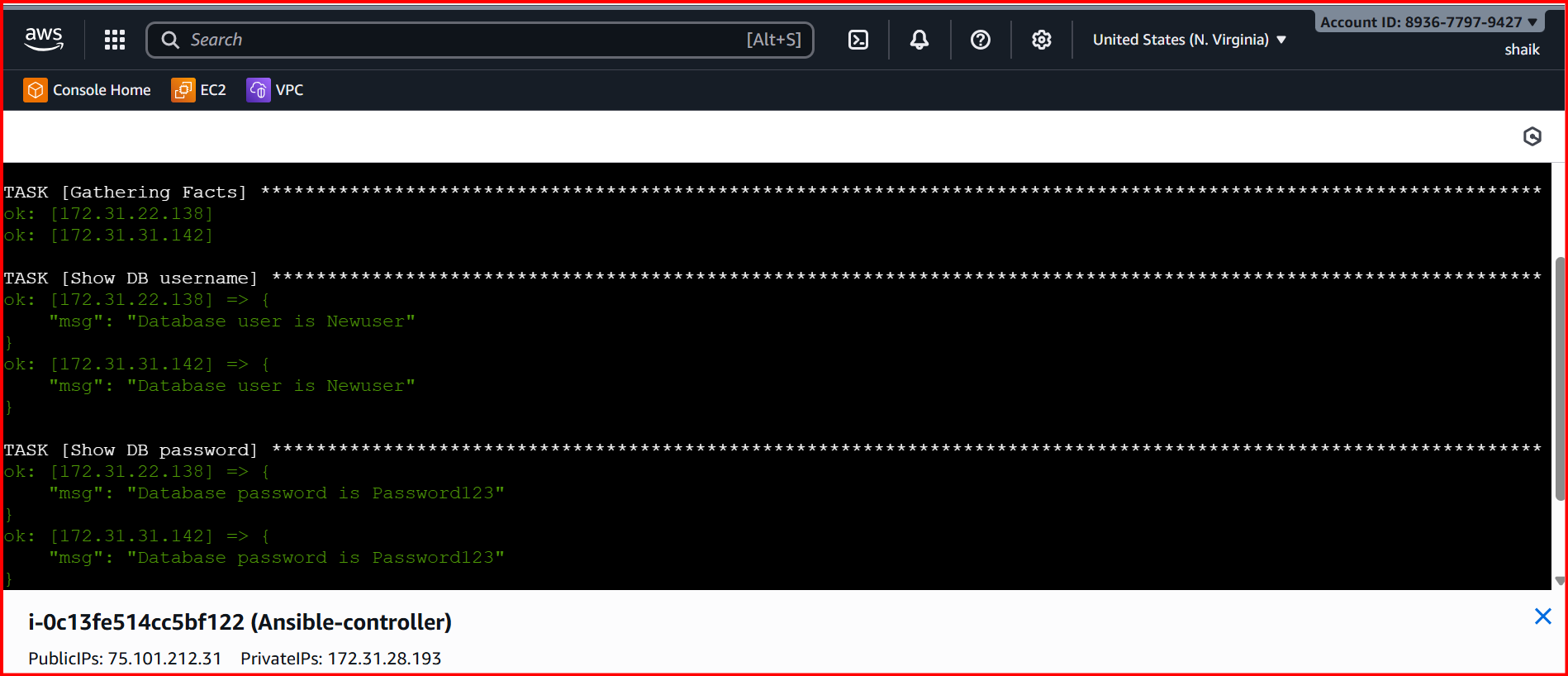
**msg: "Database password is {{ vault\_db\_pass }}"**

**run this command :**

**ansible-playbook use\_vault.yml --ask-vault-pass**

**this will ask you password : what you set first time**

****

****

**3.Ansible playbook to copy ssh-keygen from master to worker nodes. Note: a)Provision new 3 ec2 machines, one master and two worker nodes. b)Create common user called ansadm and provide sudo priviliges on 3 ec2 instances. c)Create ssh-keygen in master and your playbook should copy the keygen making it password less authentication.**

1. **Provision new 3 ec2 machines, one master and two worker nodes.**

**Step 1: Install python3-venv**

**sudo apt update**

**sudo apt install python3-venv -y**

**Step 2: Create the virtual environment**

**python3 -m venv ~/ansible-venv**

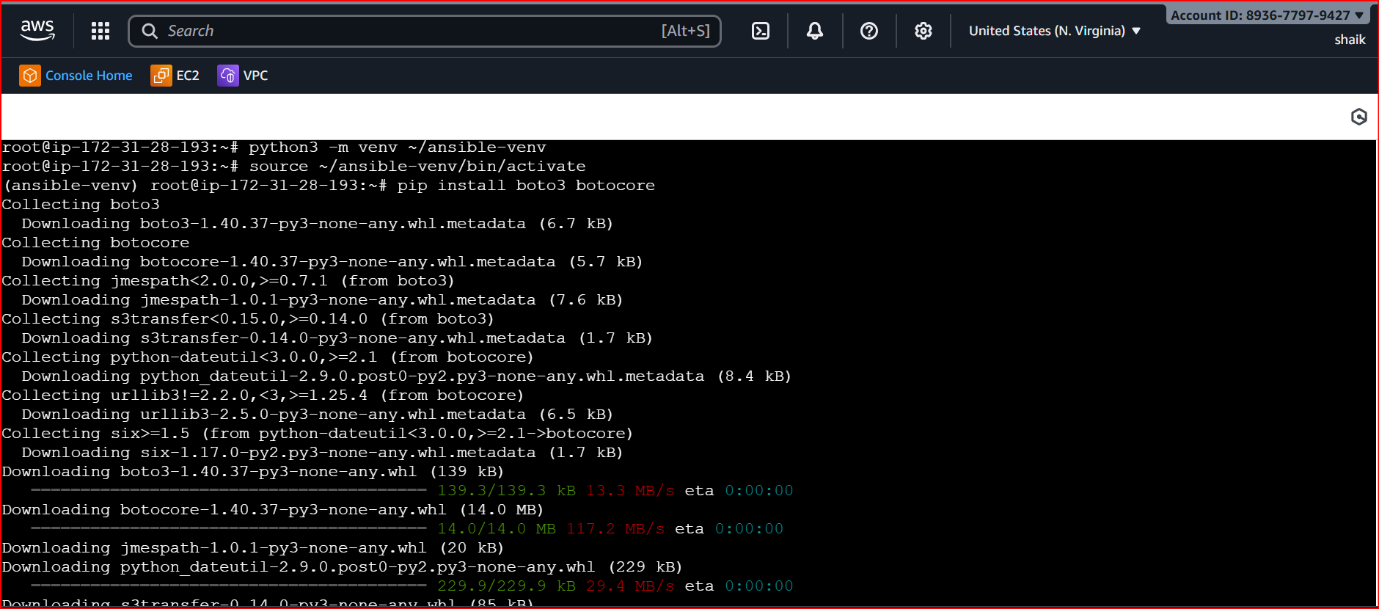
**Step 3: Activate the virtual environment**

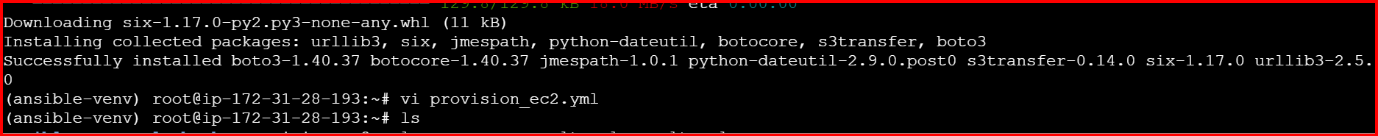
**source ~/ansible-venv/bin/activate**

**Your prompt should now show (ansible-venv) at the beginning.**

**Step 4: Install AWS packages inside the virtual environment**

**pip install boto3 botocore**

****

****

**Tip: Every time you start a new session, activate the venv first with source ~/ansible-venv/bin/activate.**

**AWS configuration:**

**curl "https://awscli.amazonaws.com/awscli-exe-linux-x86\_64.zip" -o "awscliv2.zip"**

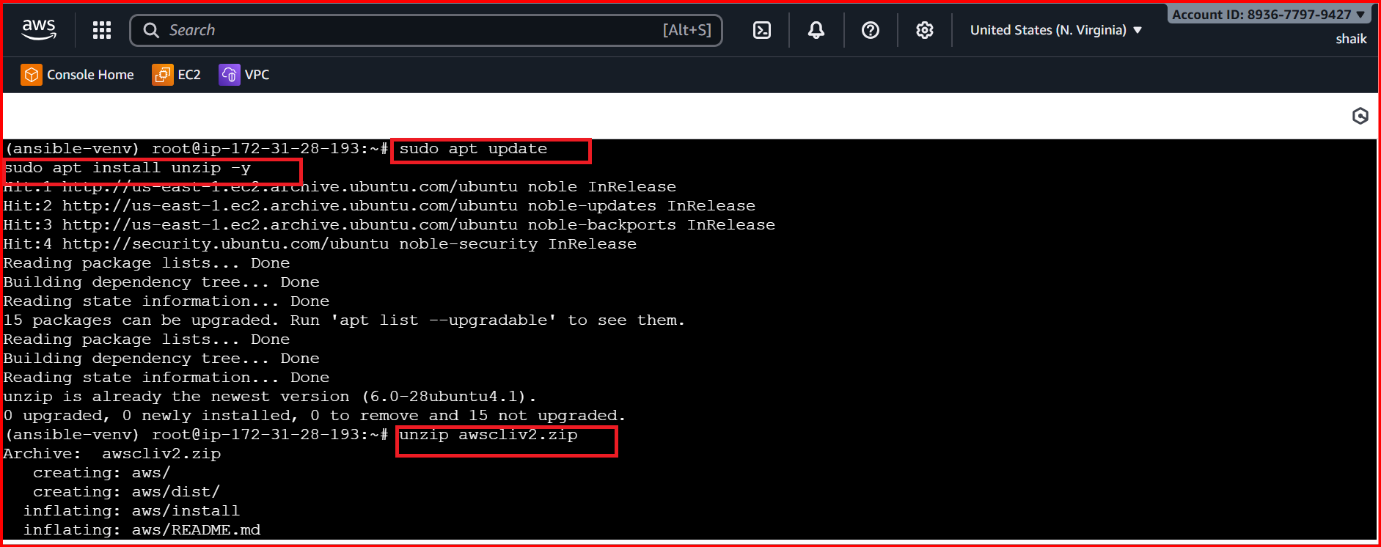
**sudo apt install unzip -y # if unzip is not installed**

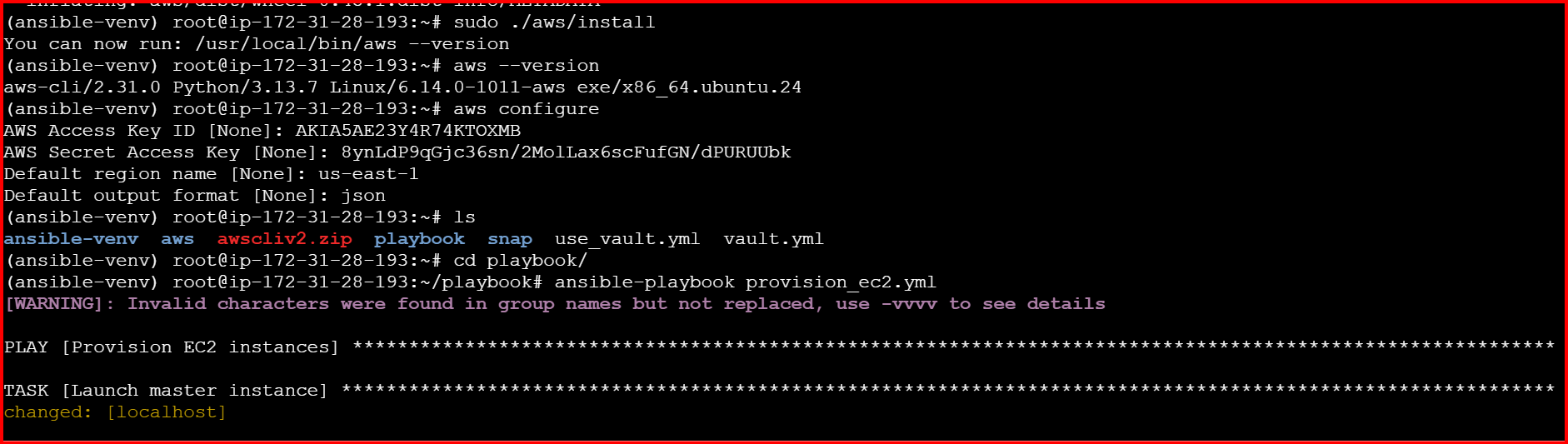
**unzip awscliv2.zip**

**sudo ./aws/install**

**aws --version**

**aws configure**

****

****

**create**

**provision\_ec2.yml**

**---**

**- name: Provision EC2 instances**

**hosts: localhost**

**connection: local**

**gather\_facts: false**

**vars:**

**region: us-east-1 # Change to your AWS region**

**key\_name: aws # Replace with your AWS key pair name**

**instance\_type: t3.micro**

**ami\_id: ami-0360c520857e3138f # Amazon Linux 2 AMI (change for your region)**

**vpc\_id: vpc-xxxxxxxx # Optional: your VPC ID**

**tasks:**

**- name: Launch master instance**

**amazon.aws.ec2\_instance:**

**name: master**

**key\_name: "{{ key\_name }}"**

**instance\_type: "{{ instance\_type }}"**

**image\_id: "{{ ami\_id }}"**

**region: "{{ region }}"**

**wait: yes**

**register: master\_instance**

**- name: Launch 2 worker instances**

**amazon.aws.ec2\_instance:**

**name: worker**

**key\_name: "{{ key\_name }}"**

**instance\_type: "{{ instance\_type }}"**

**image\_id: "{{ ami\_id }}"**

**region: "{{ region }}"**

**count: 2**

**wait: yes**

**register: worker\_instances**

**- name: Show EC2 instance details**

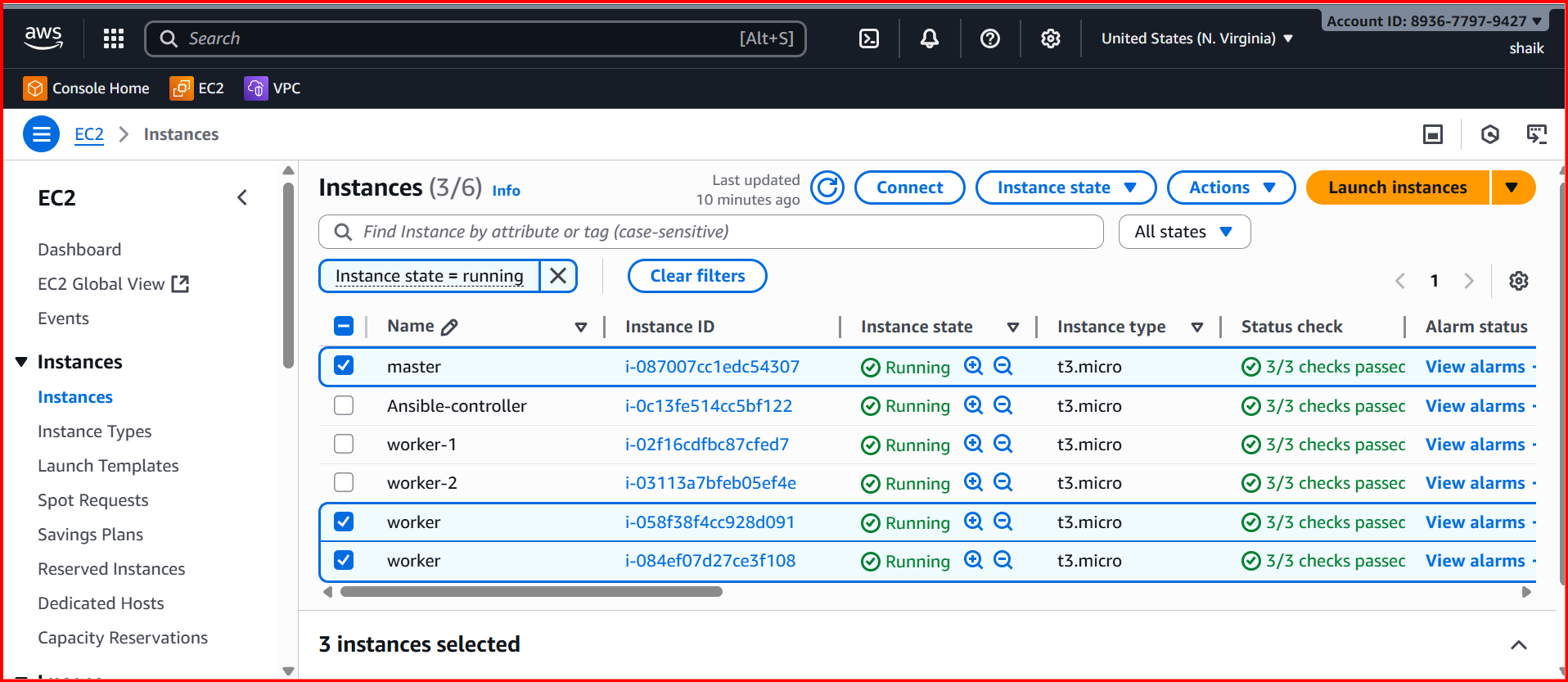
**debug:**

**msg:**

**- "Master Public IP: {{ master\_instance.instances[0].public\_ip\_address }}"**

**- "Workers Public IPs: {{ worker\_instances.instances | map(attribute='public\_ip\_address') | list }}"**

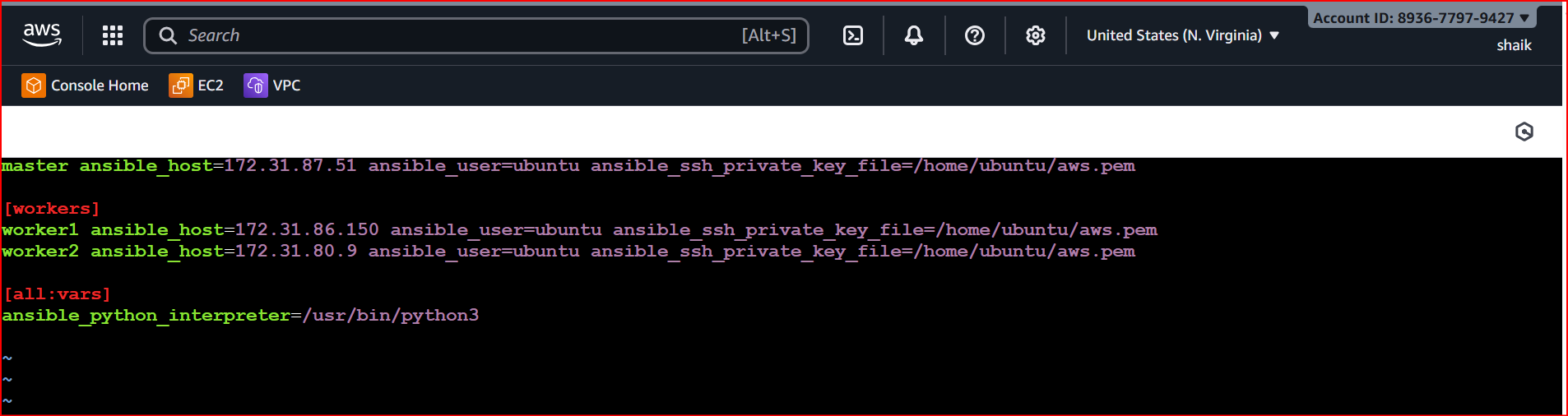
**Execute: ansible-playbook provision\_ec2.yml**

****

**b)Create common user called ansadm and provide sudo priviliges on 3 ec2 instances.**

**c)Create ssh-keygen in master and your playbook should copy the keygen making it password less authentication.**

Create vi hosts.ini



Create vi create\_user.yml

- name: Setup ansadm user and passwordless SSH

hosts: all

become: yes

tasks:

# 1. Create ansadm user

- name: Ensure ansadm user exists

user:

name: ansadm

shell: /bin/bash

state: present

create\_home: yes

# 2. Add ansadm to sudoers

- name: Add ansadm to sudoers

copy:

dest: /etc/sudoers.d/ansadm

content: "ansadm ALL=(ALL) NOPASSWD:ALL"

owner: root

group: root

mode: '0440'

- name: Generate SSH key on master

hosts: master

become: yes

tasks:

- name: Generate SSH key for ansadm

user:

name: ansadm

generate\_ssh\_key: yes

ssh\_key\_bits: 2048

ssh\_key\_file: .ssh/id\_rsa

- name: Copy SSH key from master to workers

hosts: workers

become: yes

tasks:

- name: Fetch master's public key

fetch:

src: /home/ansadm/.ssh/id\_rsa.pub

dest: /tmp/ansadm\_id\_rsa.pub

flat: yes

delegate\_to: master

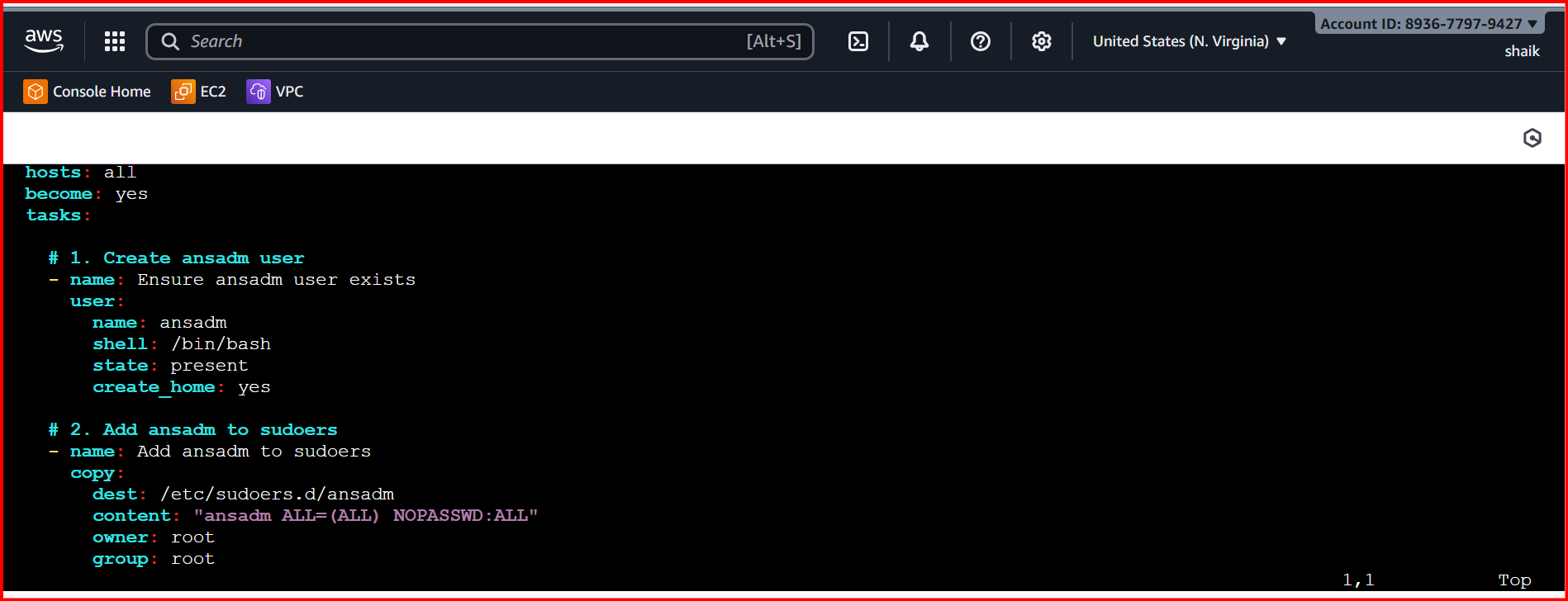
- name: Add master's public key to worker's authorized\_keys

authorized\_key:

user: ansadm

state: present

key: "{{ lookup('file', '/tmp/ansadm\_id\_rsa.pub') }}"



ansible all -i hosts.ini -m ping

