**Ansible Task-2**

1. **Watch ansible-02 video and write down notes.**
2. **Install httpd using ansible playbook, use handlers, notifiers.**
3. **Write a ansible playbook to install Apache tomcat.**
4. **Write a ansible playbook to provision one ec2 on aws.**
5. **Write a ansible playbook to copy one file from node-1 to node-2.**
6. **Write a ansible playbook to create different files with different names using single playbook.**

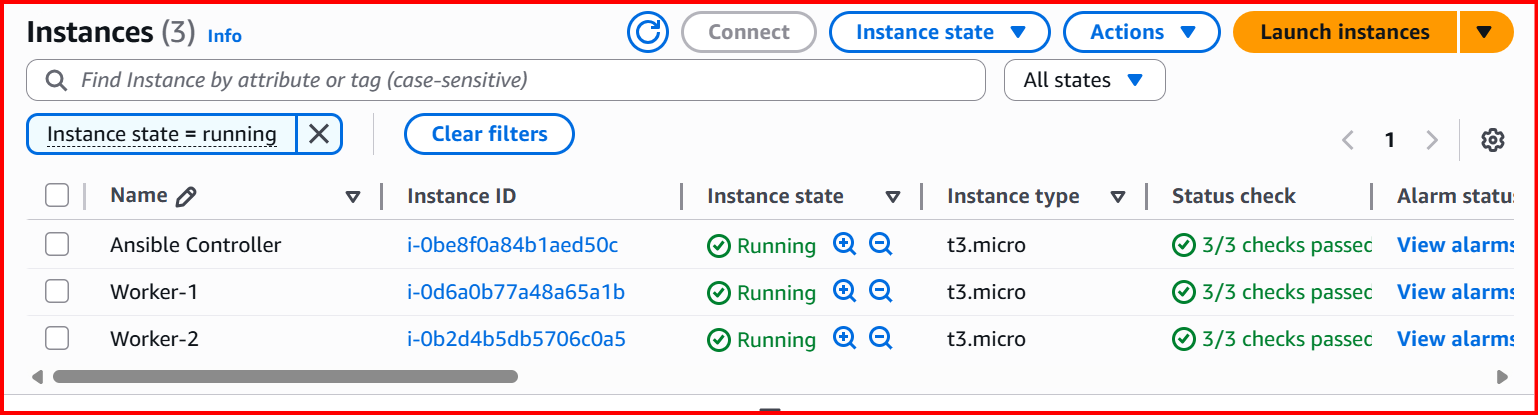
**---------------------------------------------------------------------------------------------------------**

**1 completed watching and taken notes**

**2. Install httpd using ansible playbook, use handlers, notifiers.**

**Step1:**

**Launch 3 instance, first one is Ansible controller remaining two are worker machines.**

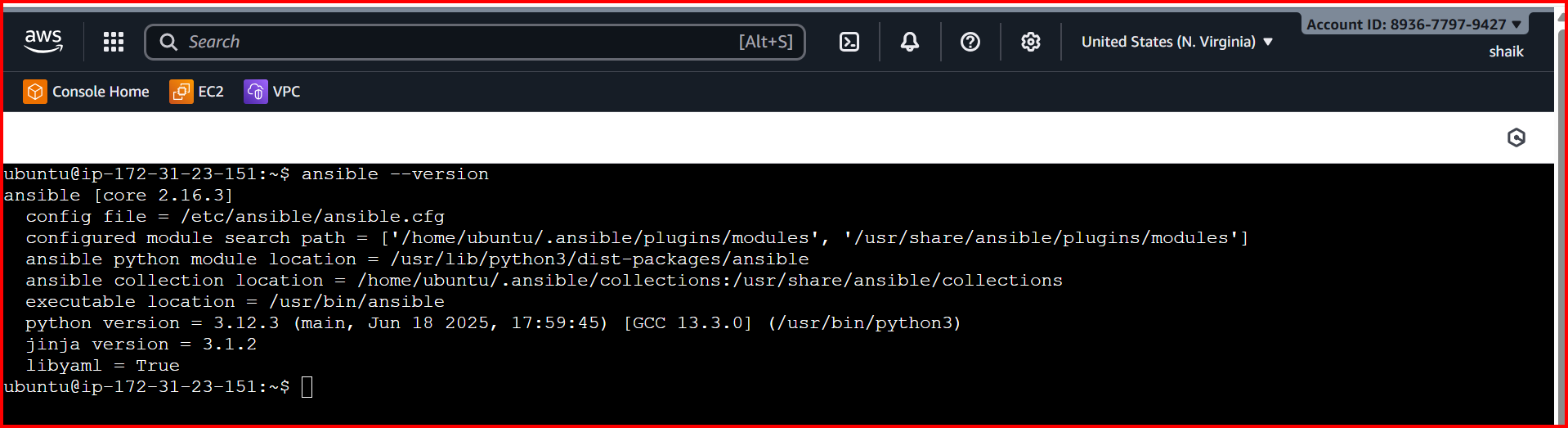
****

**Install Ansible in controller**

**sudo apt update**

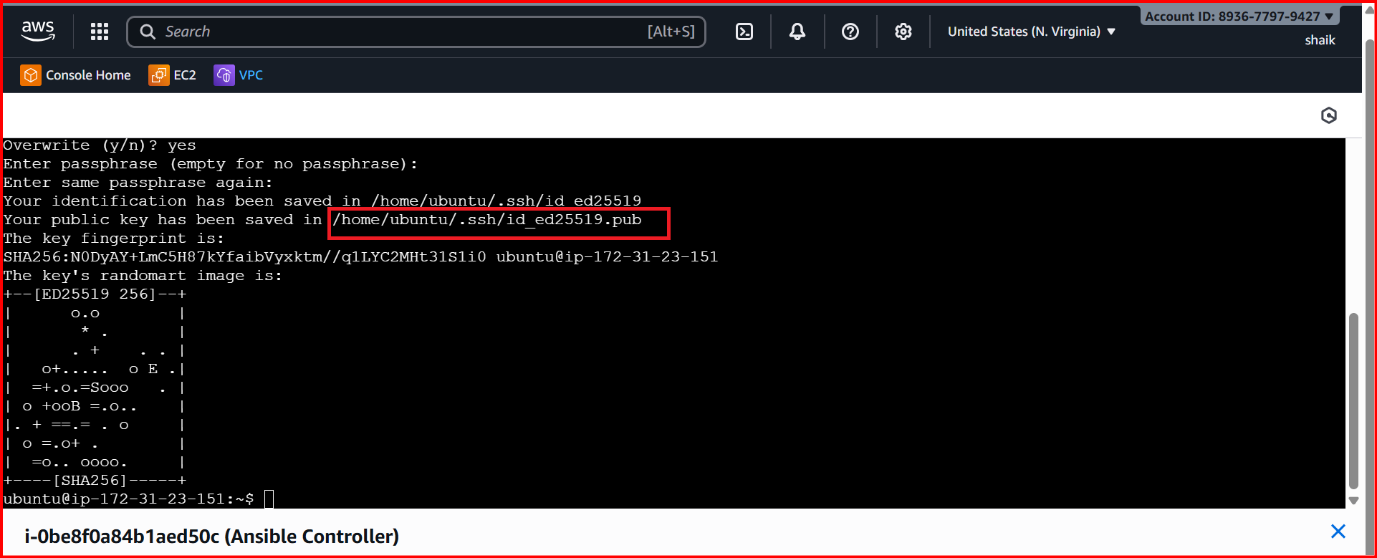
**sudo apt install ansible -y**

**ansible –version**

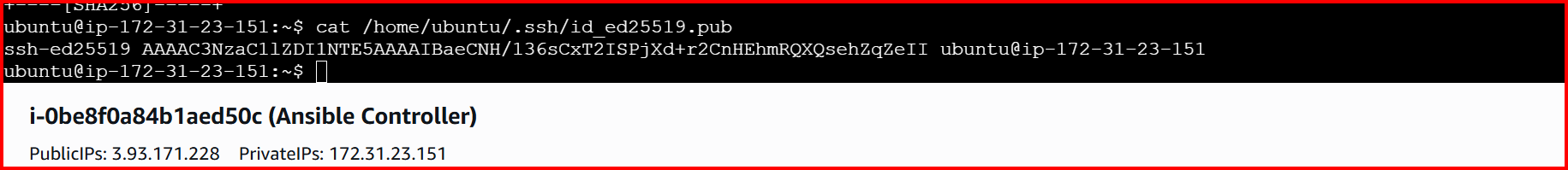
****

**Password less authentication for both worker machines**

**In controller ssh-keygen**

****

**Cat the /home/ubuntu/.ssh/id\_ed25519.pub**

****

**Copy the key and past in worker machines**

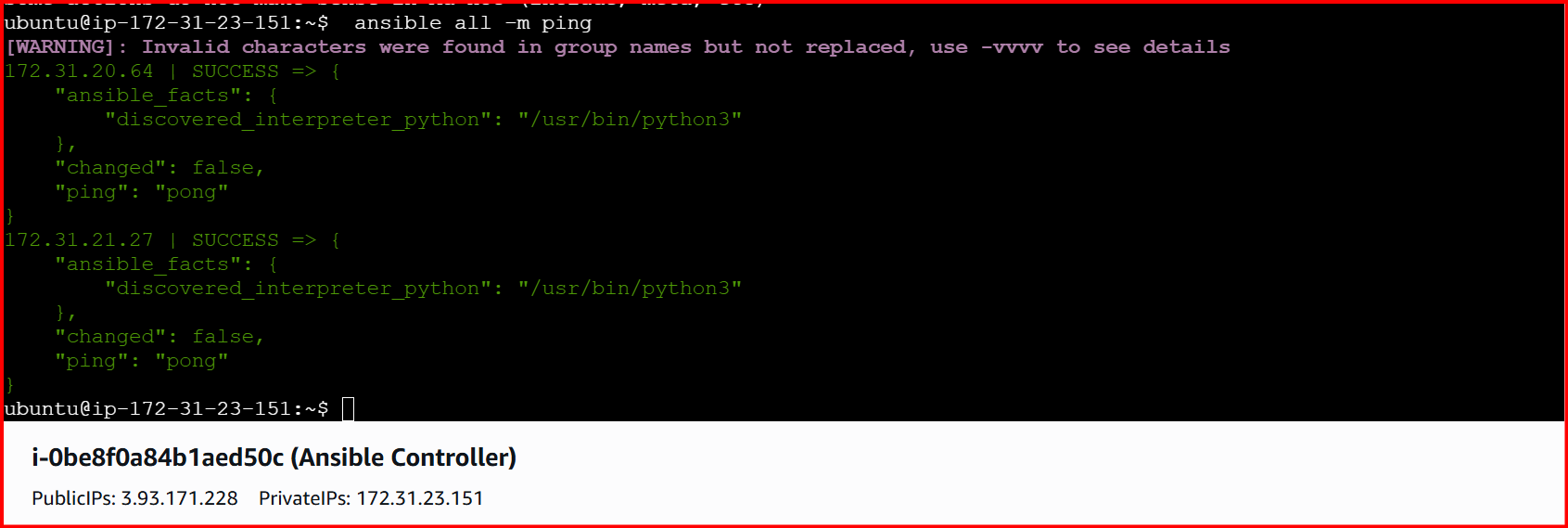
**Location**

**Cd .ssh/**

**Vi authorized\_keys**

****

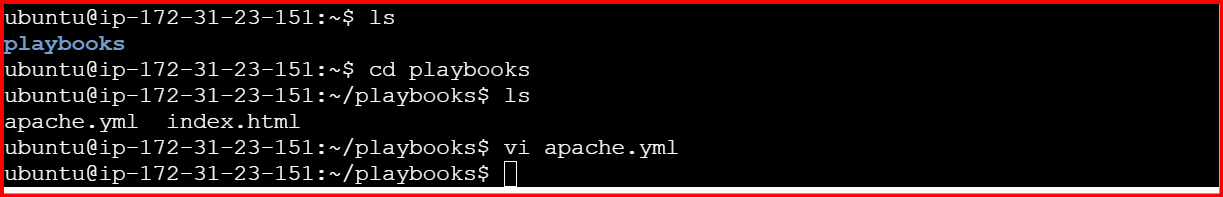
**Once done check with below command we can able to connect or not**

****

**Create directory playbooks**

**Vi apache.yml**

**Vi index.html**

****

**---**

**- hosts: all**

**become: yes**

**tasks:**

**- name: Install Apache on Ubuntu/Debian**

**apt:**

**name: apache2**

**state: latest**

**update\_cache: yes**

**- name: Copy index.html**

**copy:**

**src: index.html**

**dest: /var/www/html/**

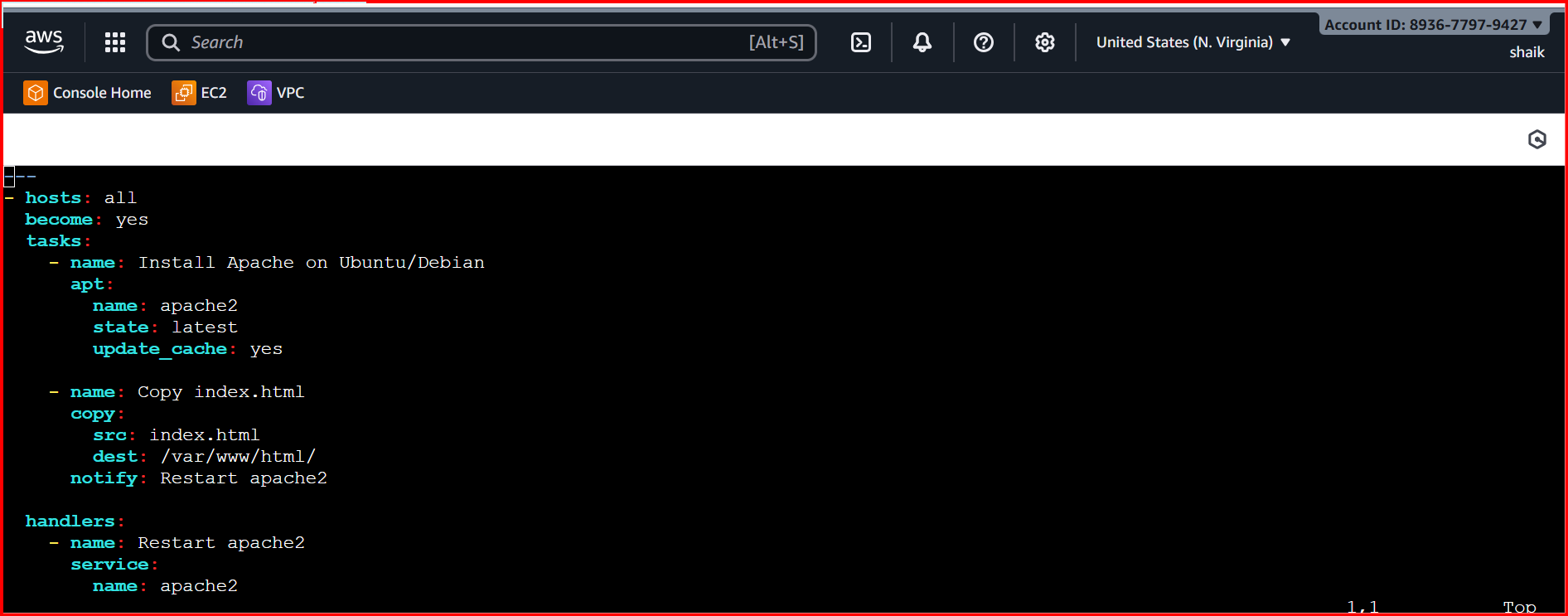
**notify: Restart apache2**

**handlers:**

**- name: Restart apache2**

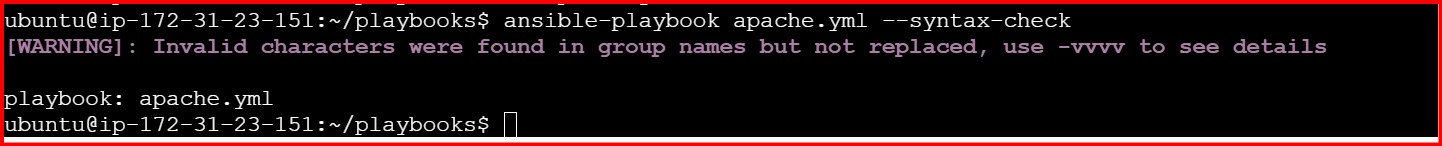
**service:**

**name: apache2**

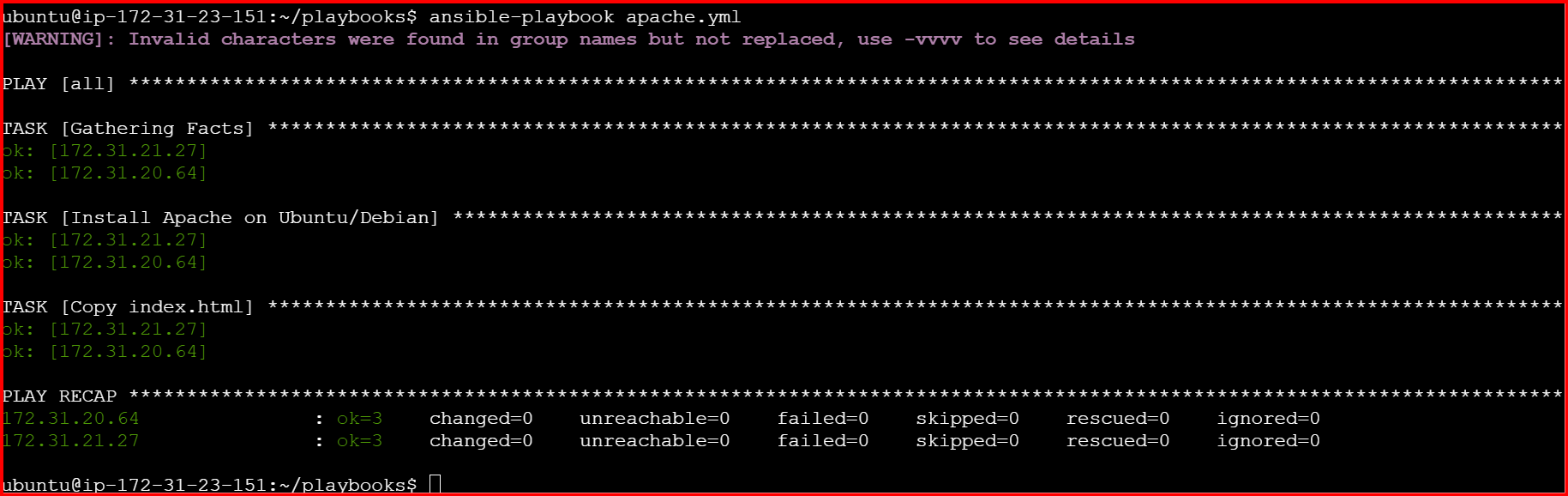
****

**Check the syntax**

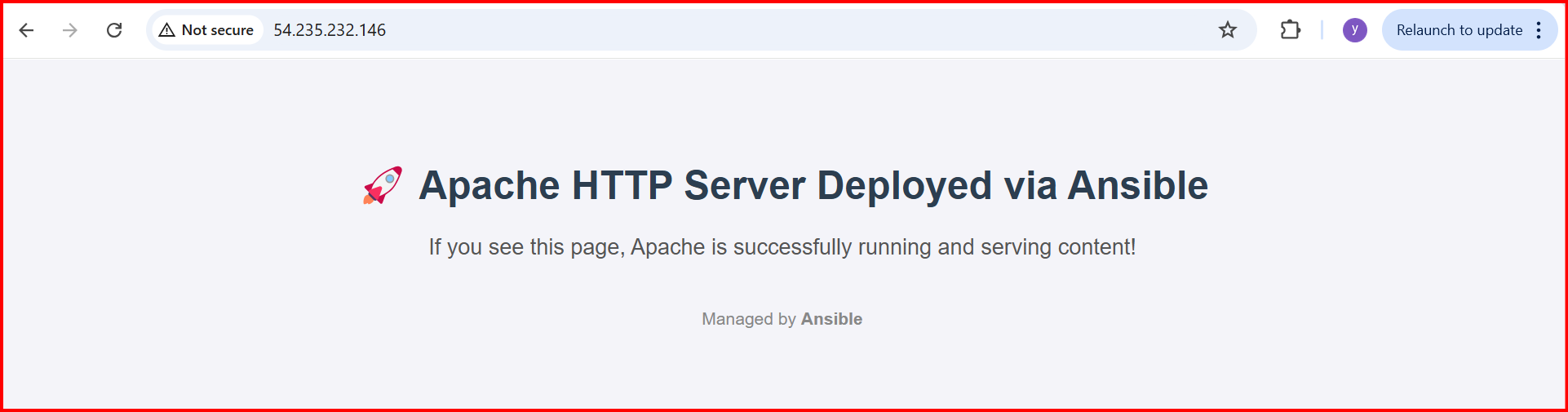
**ansible-playbook apache.yml --syntax-check**

****

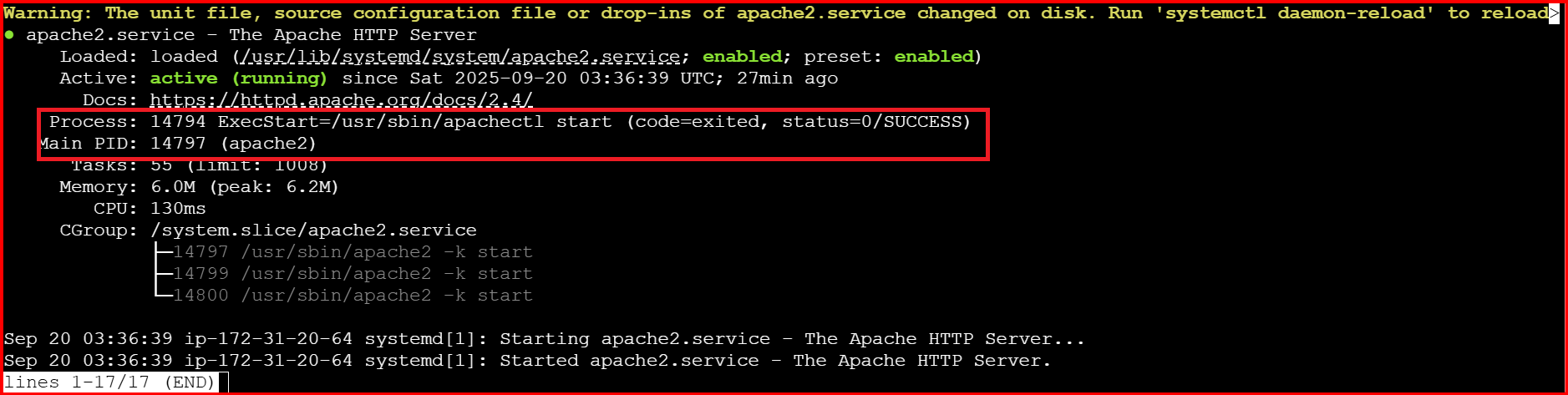
**Run the apache.yml**

****

**Check worker machine public address in browser**

****

**Apache status:**

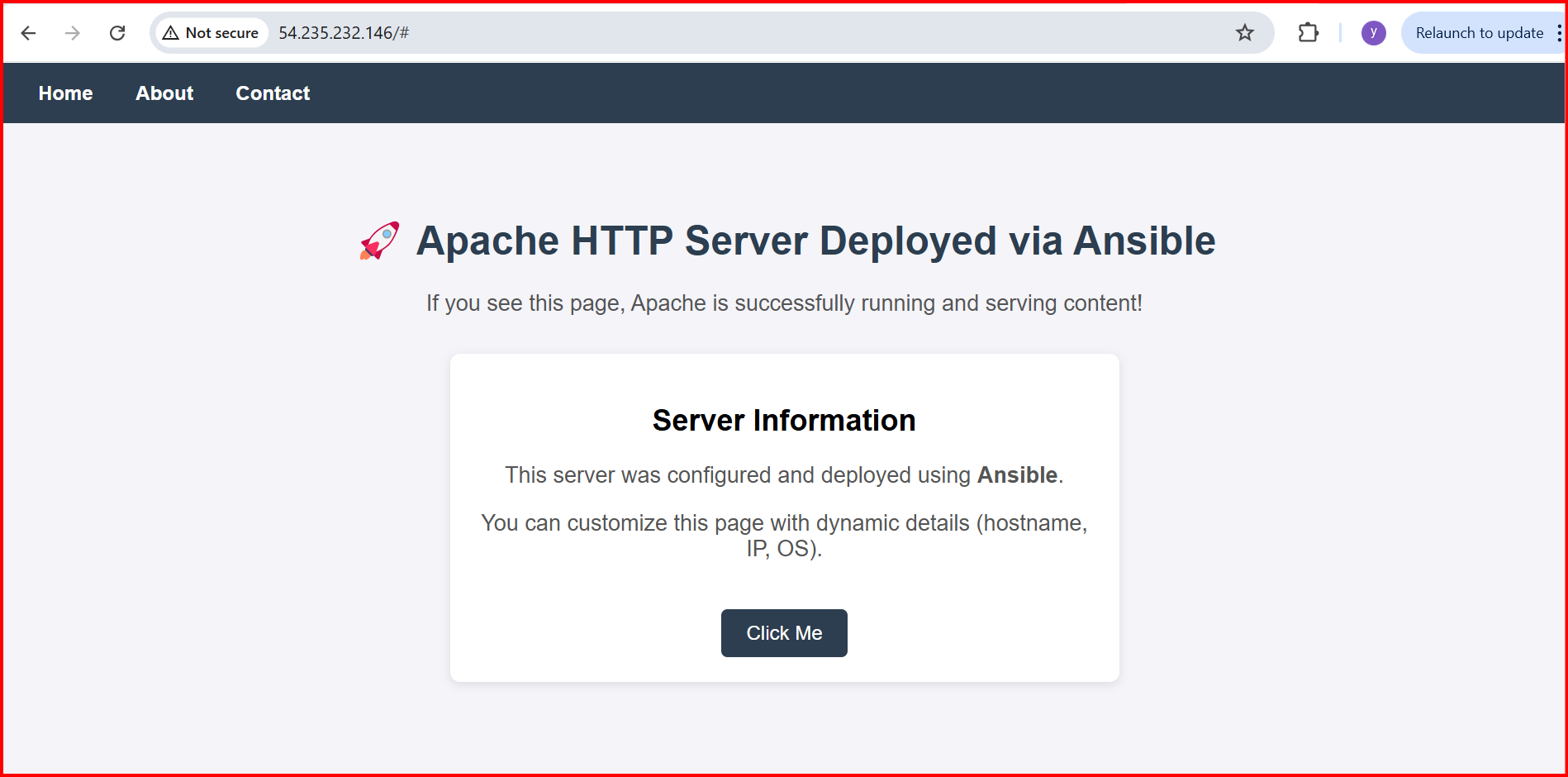
****

**If you restart the service the status will not be changes**

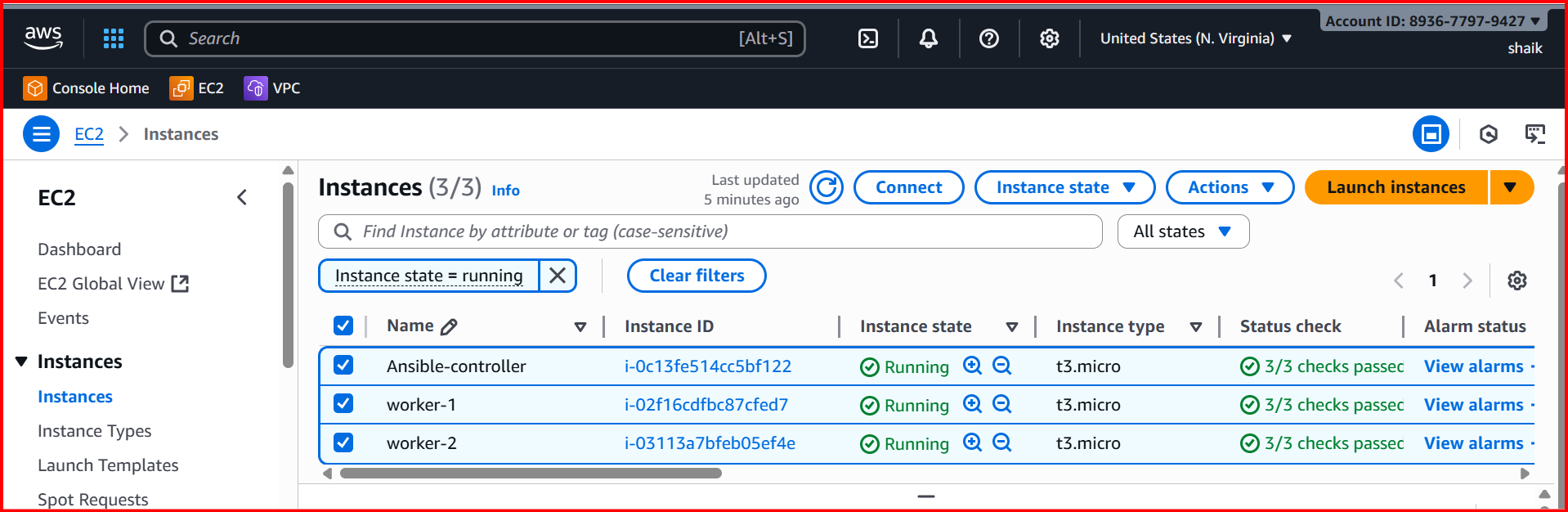
**Why because I have used notify and handlers**

**With this whenever changes occur in index.html then only it will restart then changes will be deployed.**

**Same index.html I have updated the content**

****

**3.Write a ansible playbook to install Apache tomcat.**

****

**In Ansible controller**

**Playbook🡪tomcat.yml**

**- hosts: all**

**remote\_user: root**

**tasks:**

**- name: Install Java (required for Tomcat)**

**apt:**

**name: default-jdk**

**state: present**

**update\_cache: yes**

**- name: Ensure /opt directory exists**

**file:**

**path: /opt**

**state: directory**

**mode: '0755'**

**- name: Ensure /opt/tomcat directory exists**

**file:**

**path: /opt/tomcat**

**state: directory**

**mode: '0755'**

**- name: Download and extract Tomcat 9.0.109 directly into /opt/tomcat**

**unarchive:**

**src: https://archive.apache.org/dist/tomcat/tomcat-9/v9.0.109/bin/apache-tomcat-9.0.109.tar.gz**

**dest: /opt/tomcat**

**remote\_src: yes**

**creates: /opt/tomcat/apache-tomcat-9.0.109**

**- name: Create systemd service file for Tomcat**

**copy:**

**dest: /etc/systemd/system/tomcat.service**

**content: |**

**[Unit]**

**Description=Apache Tomcat 9**

**After=network.target**

**[Service]**

**Type=forking**

**Environment=CATALINA\_HOME=/opt/tomcat/apache-tomcat-9.0.109**

**Environment=CATALINA\_BASE=/opt/tomcat/apache-tomcat-9.0.109**

**ExecStart=/opt/tomcat/apache-tomcat-9.0.109/bin/startup.sh**

**ExecStop=/opt/tomcat/apache-tomcat-9.0.109/bin/shutdown.sh**

**User=root**

**Group=root**

**Restart=always**

**[Install]**

**WantedBy=multi-user.target**

**- name: Reload systemd to recognize new service**

**systemd:**

**daemon\_reload: yes**

**- name: Enable Tomcat service**

**systemd:**

**name: tomcat**

**enabled: yes**

**- name: Start Tomcat service**

**systemd:**

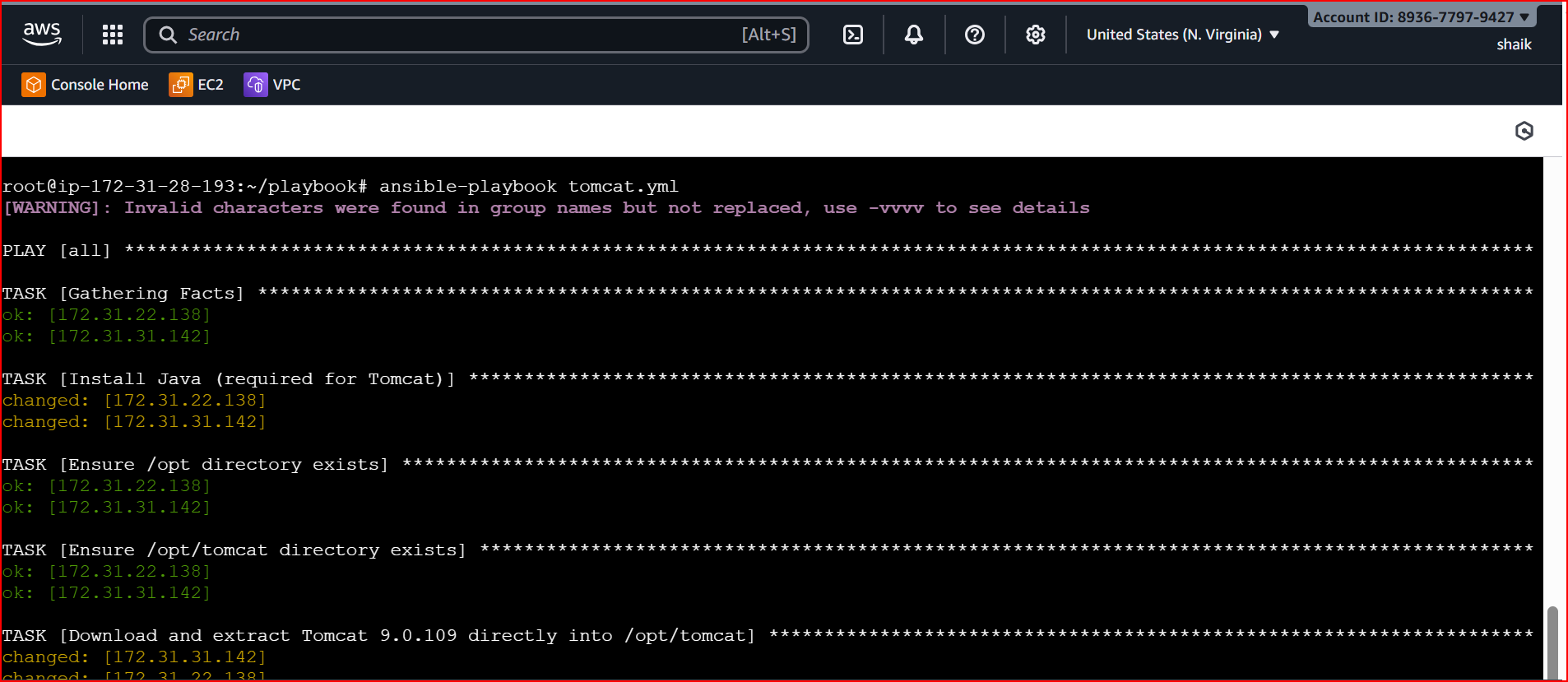
**name: tomcat**

**state: started**

**ansible-playbook tomcat.yml --syntax-check**

**ansible-playbook tomcat.yml --check**

**ansible-playbook tomcat.yml**

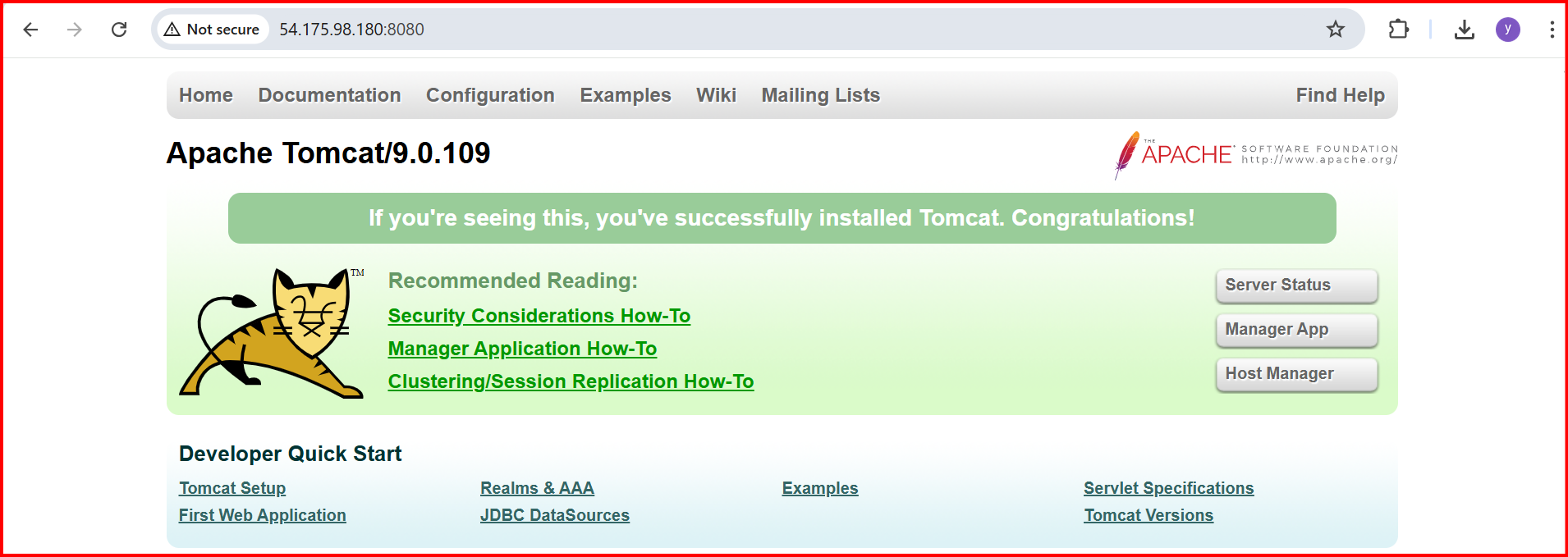
****

**Check with public ip of both the servers**

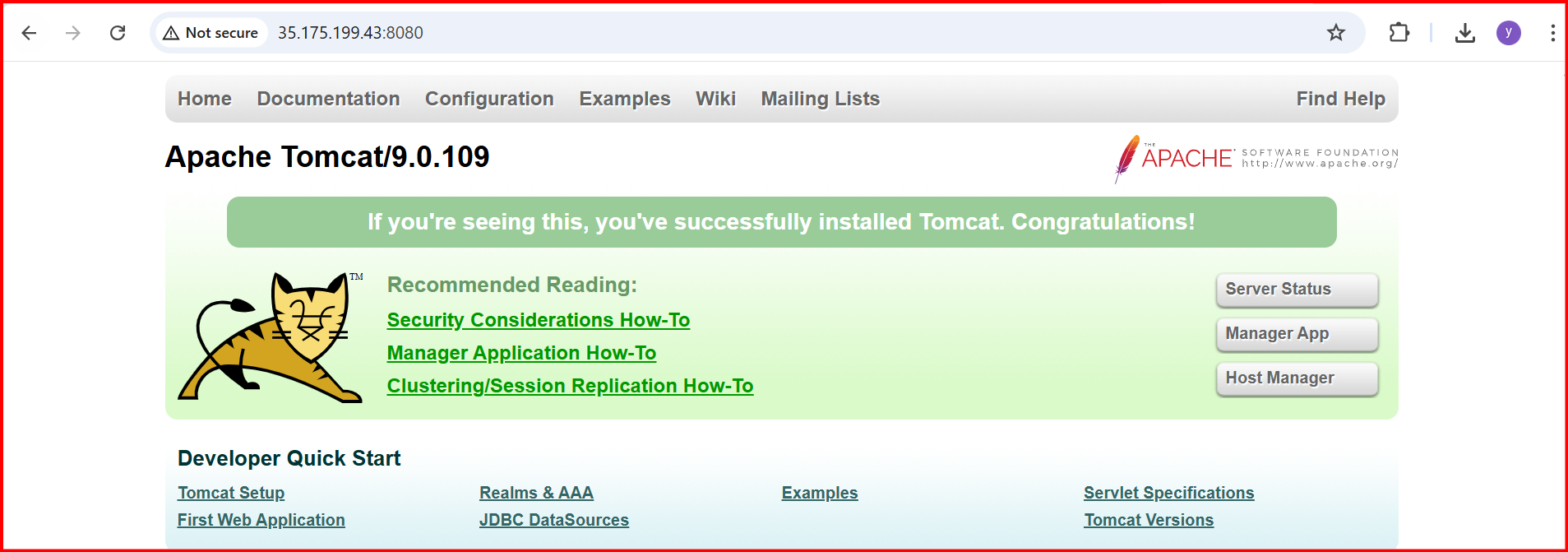
[**http://35.175.199.43:8080/**](http://35.175.199.43:8080/)

[**http://54.175.98.180:8080/**](http://54.175.98.180:8080/)

****

****

****

****

**4.Write a ansible playbook to provision one ec2 on aws**

**Create file ec2.yml**

**---**

**- name: Provision EC2 instance in AWS**

**hosts: localhost**

**connection: local**

**gather\_facts: false**

**tasks:**

**- name: Launch EC2 instance**

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

**name: "ansible-demo-instance"**

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

**instance\_type: t2.micro**

**image\_id:** [**i-083804d1e492519e2**](https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#InstanceDetails:instanceId=i-083804d1e492519e2) **# Ensure this AMI ID is valid in your region**

**region: us-east-1**

**wait: true**

**count: 1**

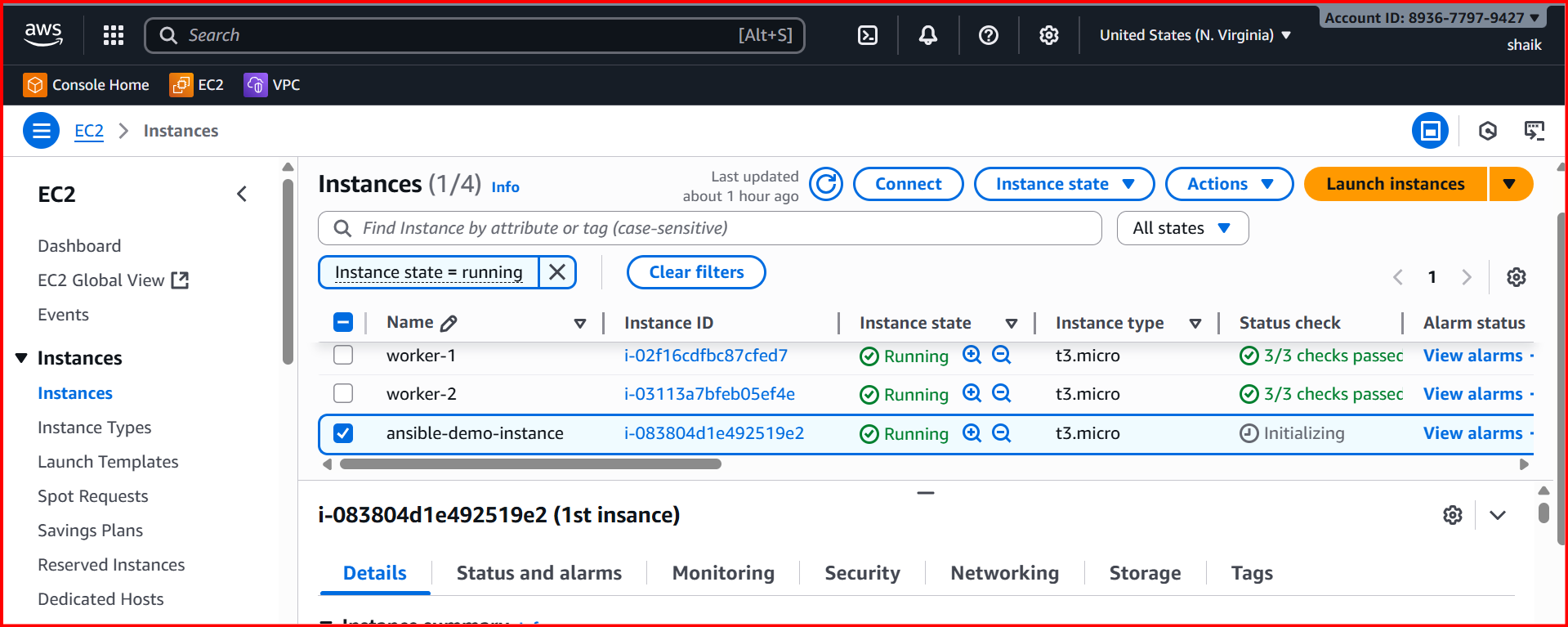
**security\_group: default**

**register: ec2**

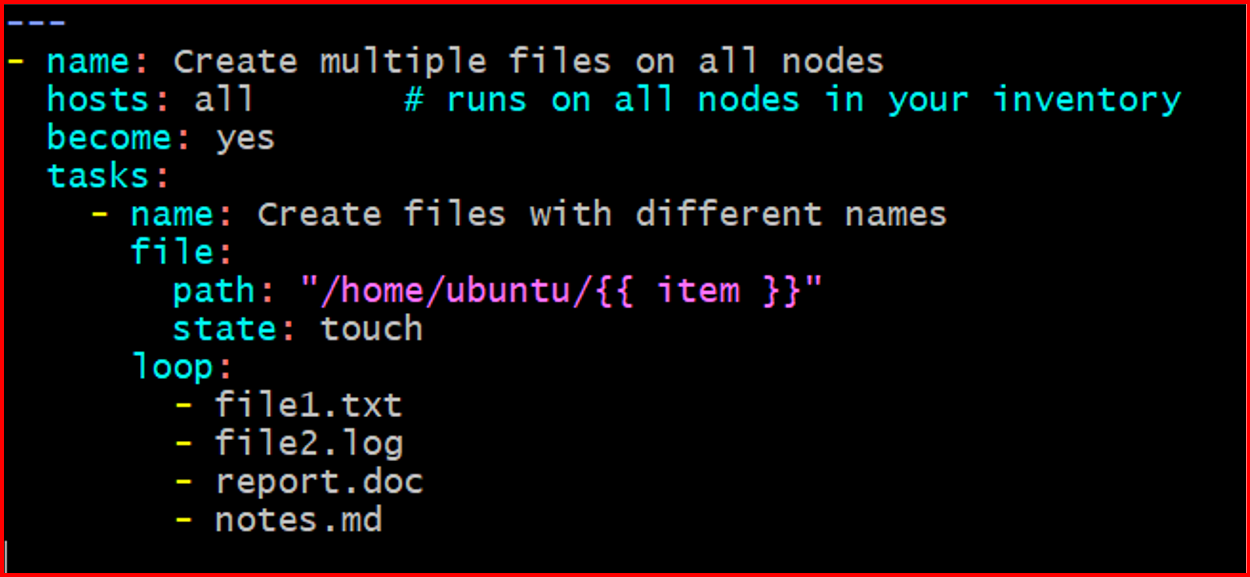
**- name: Show instance info**

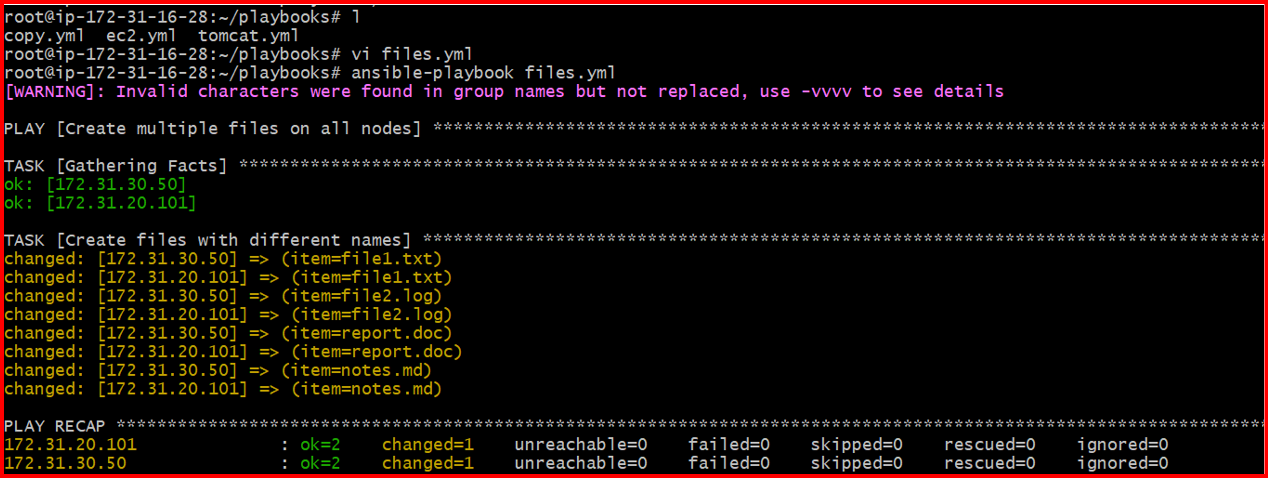
**debug:**

**var: ec2.instances**

****

**6.Write a ansible playbook to create different files with different names using single playbook.**

****

****