Infrastructure Provisioning using Ansible

Problem:

provisioning cloud resources using Ansible to set up a basic web application infrastructure.

infrastructure consist of the following components:

1. An AWS EC2 instance running a basic web server.
2. An AWS RDS MySQL database for storing application data.

Requirements:
Create an EC2 instance with the following specifications:

1. Instance Type: t2.micro
2. AMI: Ubuntu Server 20.04 LTS
3. Security Group: Allow incoming HTTP (port 80) and SSH (port 22) traffic.

Create an RDS MySQL instance with the following specifications:

1. Instance Class: db.t2.micro
2. Engine: MySQL
3. Allow incoming traffic on port 3306 from the EC2 instance's security group.

ANISBLE PLAYBOOK FOR CREATING EC2 AND RDS INFRASTRUCTURE

```
---
- name: Provision AWS Infrastructure
  hosts: localhost
  gather_facts: no
  vars:
    aws_access_key: "AKIA6ODU7LJYX2CVOHHA"
    aws_secret_key: "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"
    aws_region: "us-west-2"
    ec2_instance_type: "t2.micro"
    ec2_ami: "ami-0c66c6068e0b884e1"
```

```
  ec2_security_group: "web-server-sg"
  rds_instance_class: "db.t3.micro"
  rds_engine: "mysql"
  rds_engine_version: "5.7.44"
  rds_db_name: "webapp_db"
  rds_master_username: "admin"
  rds_master_password: "password"
  rds_security_group: "rds-sg"
  key_pair_name: "KEY1"
tasks:
  - name: Create security group for EC2
    amazon.aws.ec2_security_group:
      name: web_sg
      description: Security group for web server
      region: "{{ aws_region }}"
      rules:
        - proto: tcp
          ports:
           - 22
            - 80
          cidr_ip: 0.0.0.0/0
      state: present
    register: ec2_sg

  - name: Create security group for RDS
    amazon.aws.ec2_security_group:
      name: rds_sg
      description: Security group for RDS
      region: "{{ aws_region }}"
      rules:
```

```
      - proto: tcp
        ports:
          - 3306
        group_id: "{{ ec2_sg.group_id }}"
    state: present
  register: rds_sg

- name: Launch EC2 instance
  amazon.aws.ec2_instance:
    key_name: "{{ key_pair_name }}"
    instance_type: "{{ ec2_instance_type }}"
    image_id: "{{ ec2_ami }}"
    wait: yes
    security_groups: "{{ ec2_sg.group_id }}"
    region: "{{ aws_region }}"
    count: 1
    tags:
      Name: web_server
      wait: yes
    user_data: |
      #!/bin/bash
      sudo apt update
      sudo apt install -y apache2
      sudo systemctl start apache2
      sudo systemctl enable apache2
  register: ec2_instance

- name: Create RDS instance
  amazon.aws.rds_instance:
    db_instance_identifier: mydbinstance
```

engine: "{{ rds_engine }}"

db_instance_class: "{{ rds_instance_class }}"

master_username: "{{ rds_master_username }}"

master_user_password: "{{ rds_master_password }}"

allocated_storage: 20

vpc_security_group_ids: "{{ rds_sg.group_id }}"

db_name: "{{ rds_db_name }}"

region: "{{ aws_region }}"

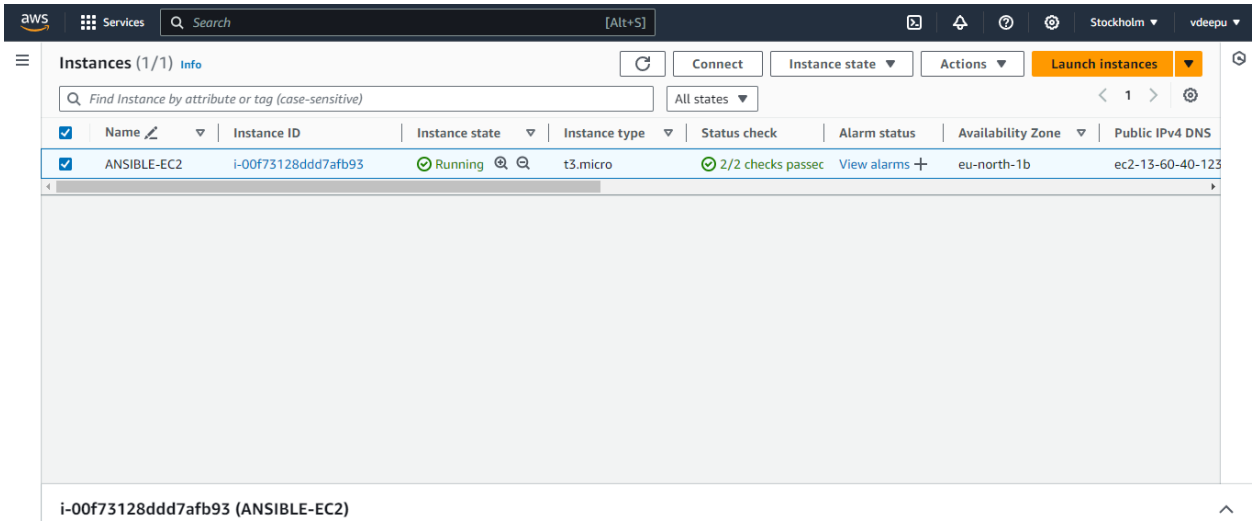
publicly_accessible: yes

wait: yes

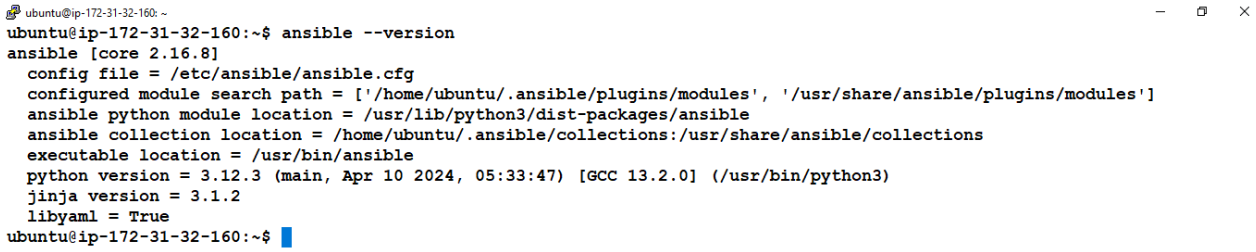
aws_access_key: "{{ aws_access_key }}"

aws_secret_key: "{{ aws_secret_key }}"

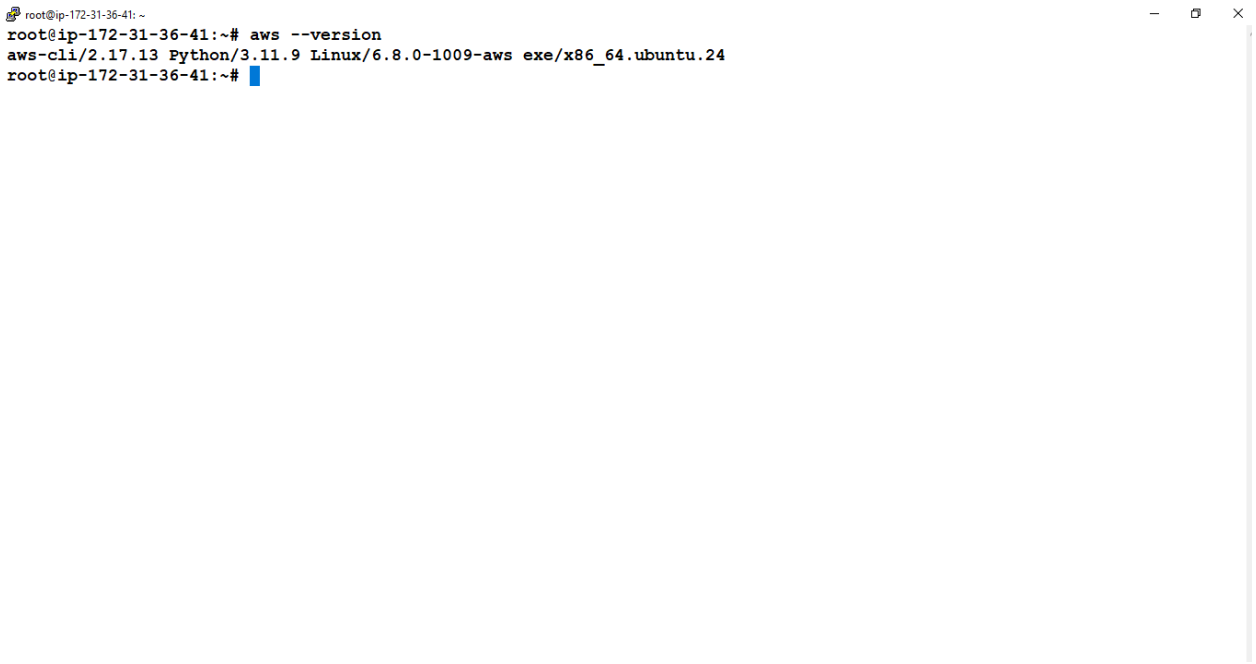
1. Creating a control node EC2 instance for installing Ansible and provisioning the infrastructure.



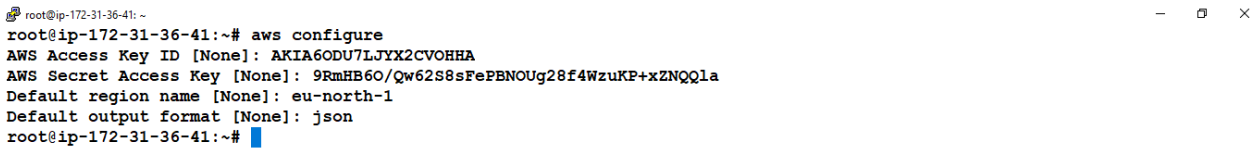
2. In that control node installing ansible and verifying its version.



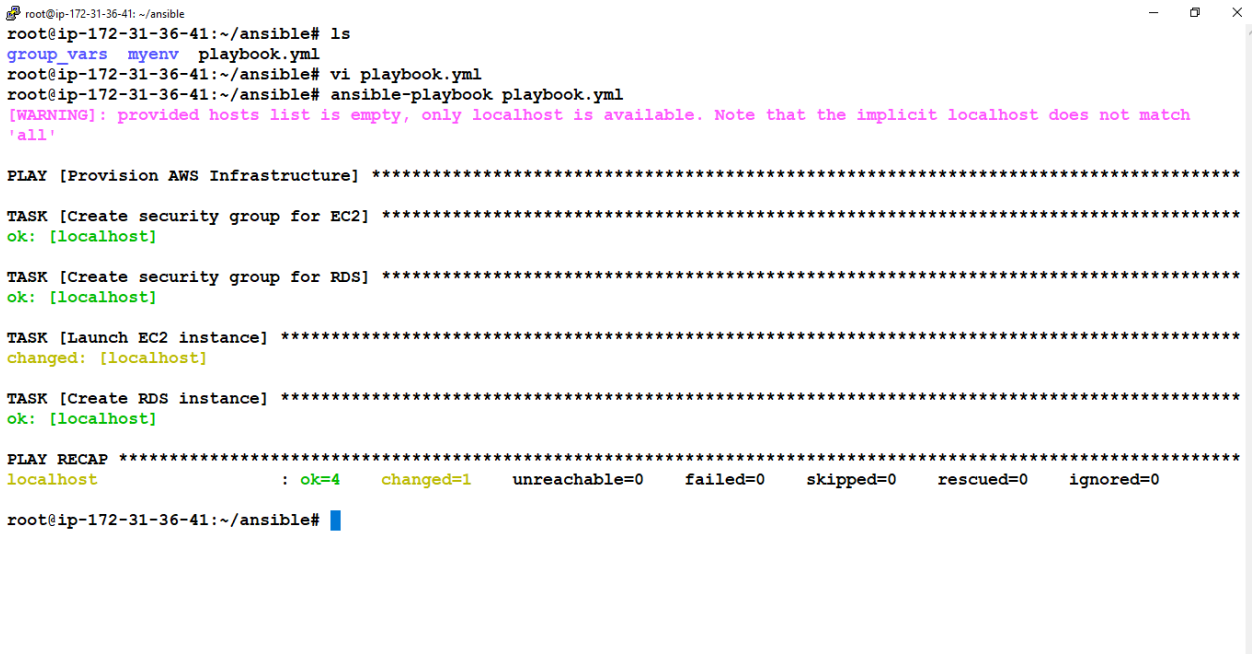
3. Installing aws cli in control node that allows us to manage AWS resources directly.



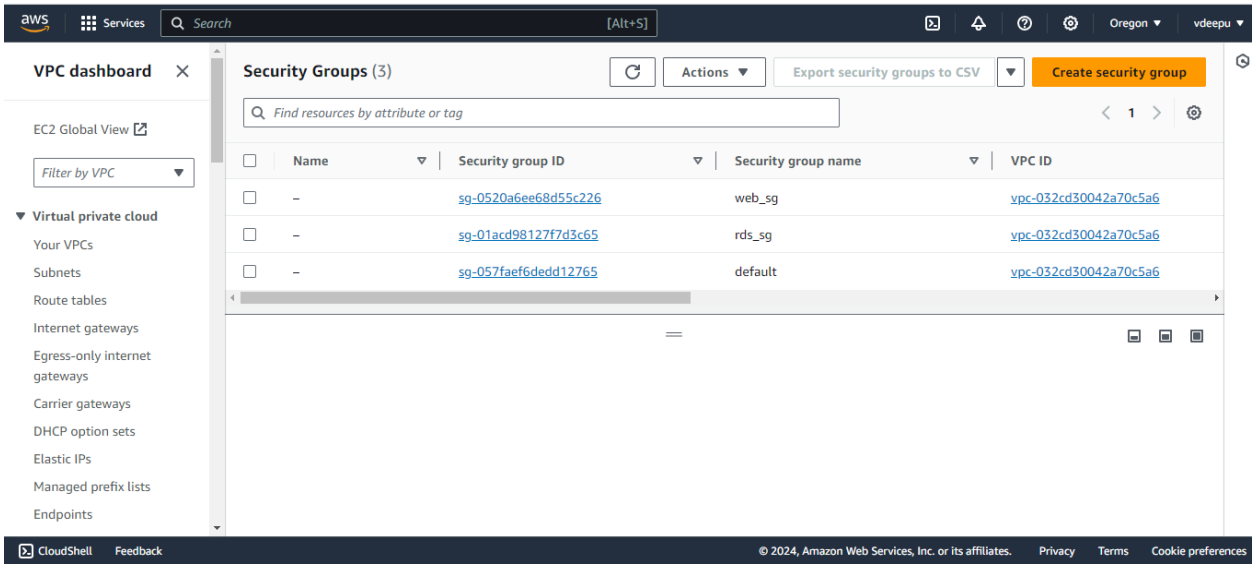
4. Configuring aws configure in control node.



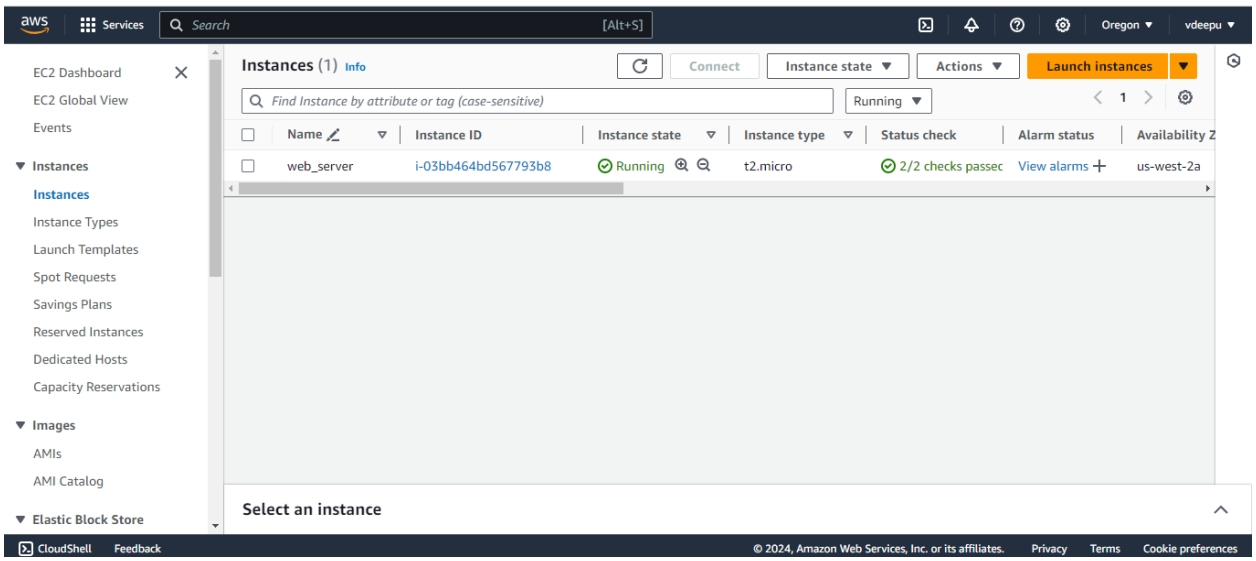
5. Running a Ansible playbook for creating a ec2 and rds infrastructure.

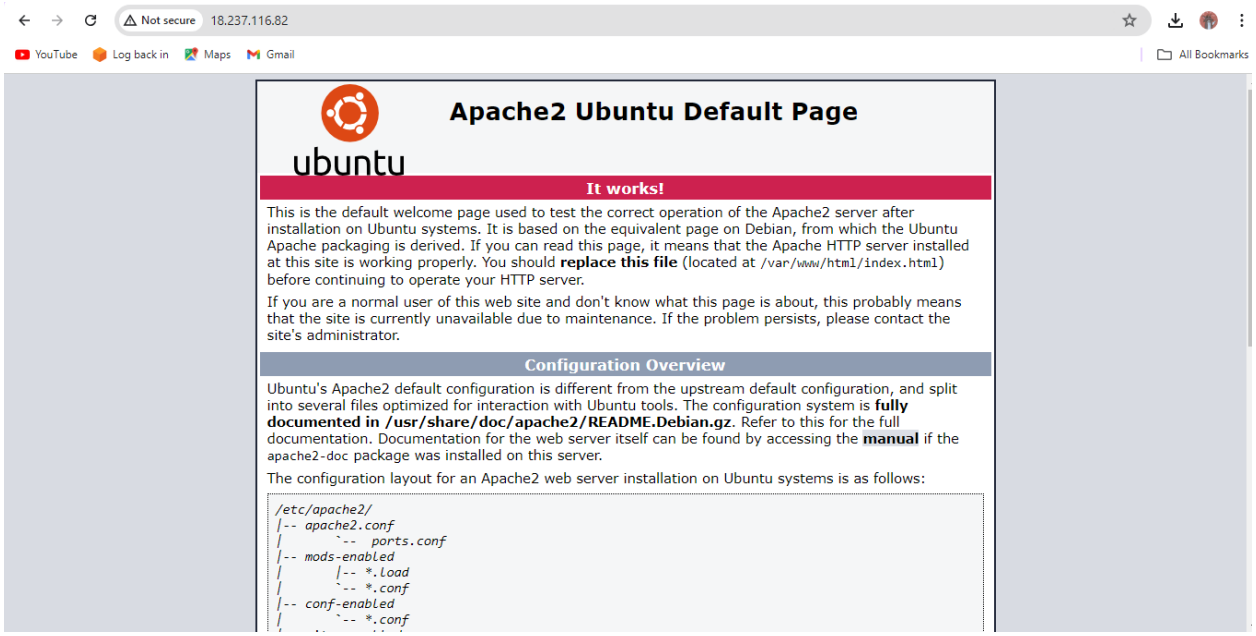


6. Security group is created for EC2 and RDS with specific ports opened for allowing incoming traffics from ports ssh(22) , http(80) and mysql(3306) for allowing incoming traffics from ec2 instance security group.



7. EC2 Instance is created with a basic web server- Apache



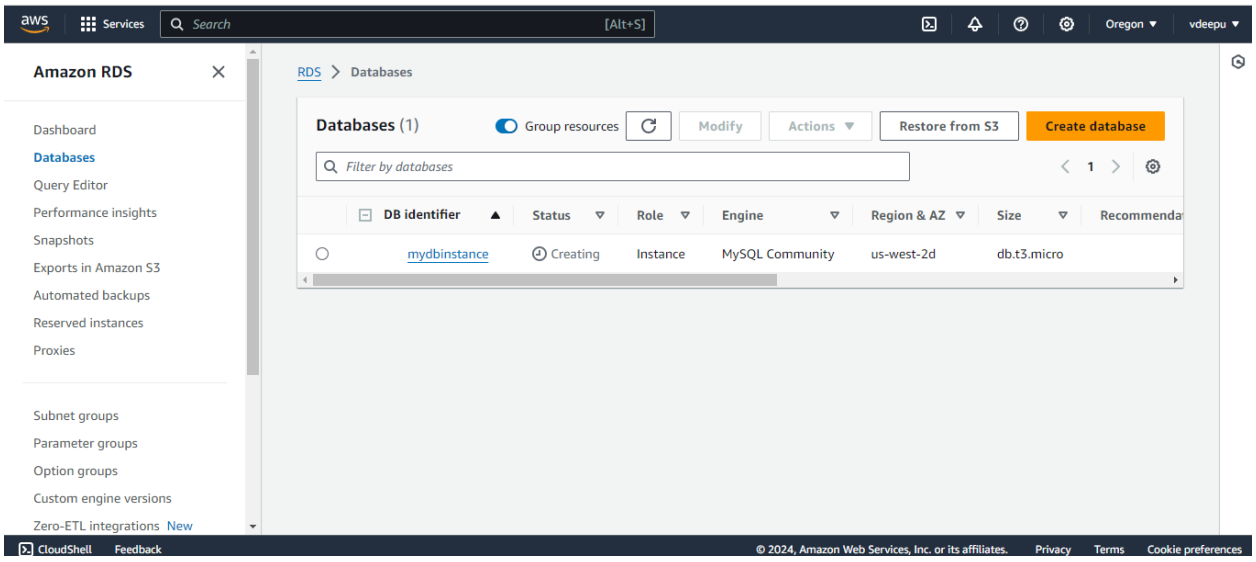


8. Changed the html content of the Apache web server.

Welcome to My Simple Web Page

This is a simple Apache web server.

Note: For RDS Database the db.t2.micro instance type is no longer supported for new RDS MySQL database instance creations as of June 1, 2024, and support will end entirely on December 31, 2024. Instead, i used the db.t3.micro instance type, which offers better performance and cost efficiency.



9. RDS Mysql Database is created with a instance type of db.t3.micro