Tej Mandaliya

Building highly scalable and available infrastructure that handles failure on AWS

T

T Tej Mandaliya

Introduction

Objective:

This project demonstrates the creation of an Auto Scaling group that automatically adjusts the number of EC2 instances based on demand. The EC2 instances are launched using a pre-configured launch template in a custom VPC.

Technologies used:

AWS EC2, Auto Scaling, AWS ELB, IAM, Launch Templates, VPC (Virtual Private Cloud), Security Groups, and Cloud Watch.

This project took me...

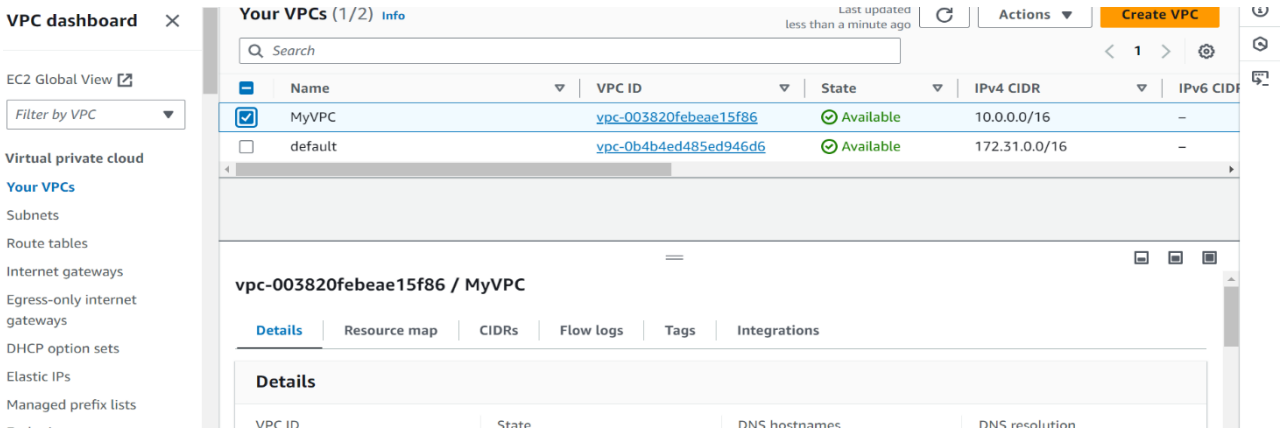
Approximately 1 hour

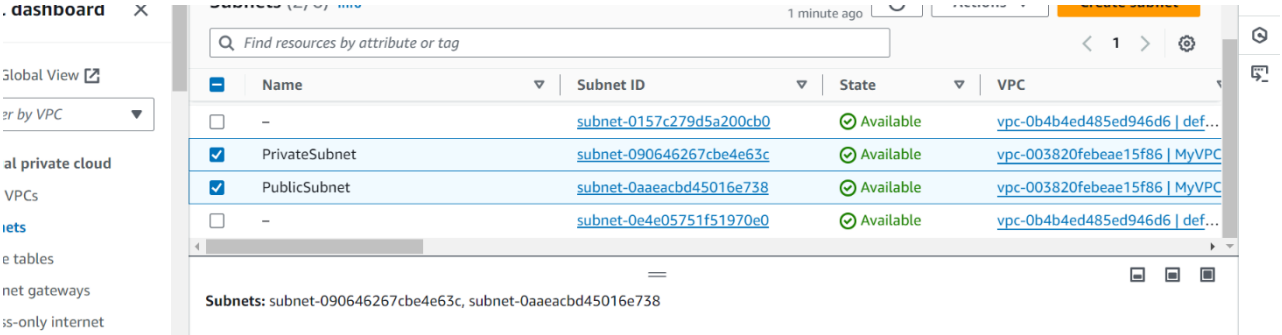
Tej Mandaliya

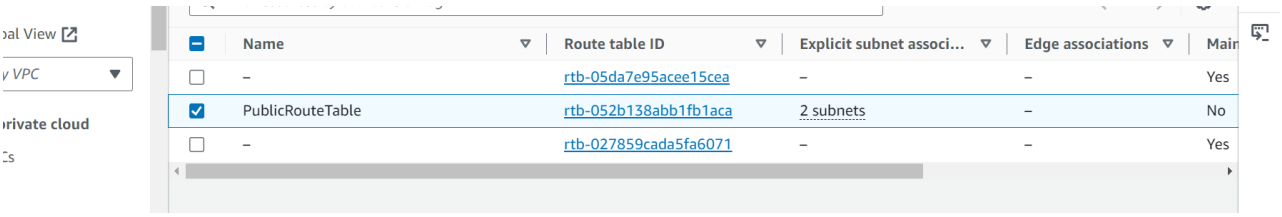
How I Set Up

First Creates a Custom VPC

- **VPC Creation:**
  - **CIDR block: 10.0.0.0/16**
  - **2 Public subnets (in different Availability Zones):**
    - **Subnet 1: ap-south-1a, CIDR block: 10.0.1.0/24**
    - **Subnet 2: ap-south-1b, CIDR block: 10.0.2.0/24**
- **Internet Gateway:**
  - **Attach to the VPC to allow external access.**
- **Route Tables:**
  - **Route traffic to the Internet Gateway for public subnets.**





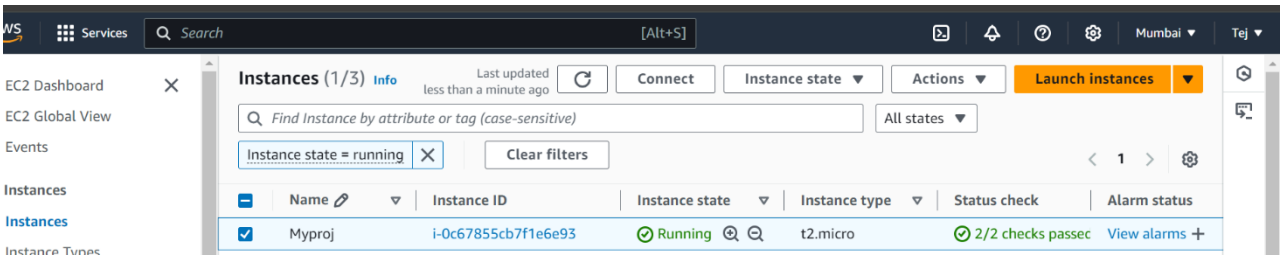


Second Step Launch EC2 Instances

Launch an EC2 Instance:

- **Choose an Amazon Machine Image (AMI) (e.g., Amazon Linux 2).**
- **Select an Instance Type (e.g., t2.micro for free tier).**
- **Click Next: Configure Instance Details.**
  - **Ensure the instance is in the public subnet.**

2. **Configure Security Group:**
   - **Create a new security group (or select an existing one).**
   - **Add inbound rules for:**
     - **HTTP: Port 80 (Type: HTTP, Source: Anywhere or specific IP).**
     - **HTTPS: Port 443 (Type: HTTPS, Source: Anywhere).**
     - **SSH: Port 22 (Type: SSH, Source: Your IP).**
3. **After that Launch Instance.**

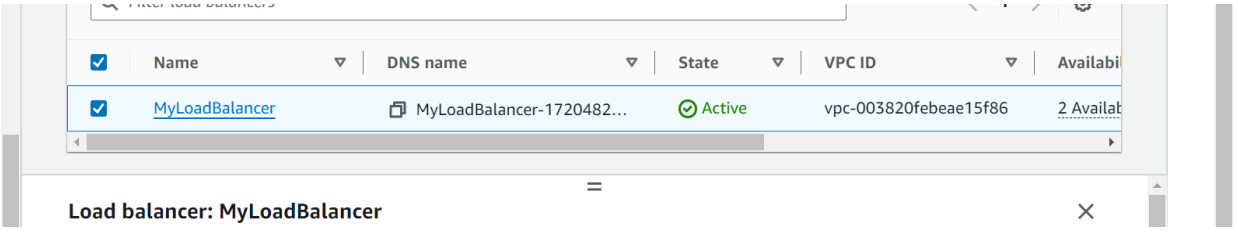


Set Up Elastic Load Balancer (ELB)

1. **Create a Load Balancer:**
   - **In the EC2 dashboard, go to Load Balancers > Create Load Balancer.**
   - **Select Application Load Balancer.**
   - **Name it (e.g., MyLoadBalancer).**
   - **Select Internet-facing and your VPC.**
   - **Choose at least two public subnets.**
2. **Configure Listeners:**
   - **Set the listener (HTTP on port 80).**
   - **Click Next: Configure Security Settings (you can skip HTTPS**

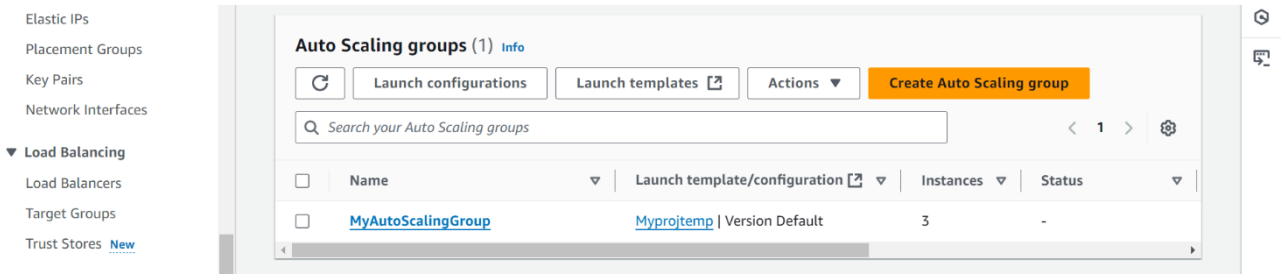
setup for now).

3. Configure Target Groups:
   - Create a new target group (e.g., MyTargetGroup):
     - Target type: Instance.
     - Protocol: HTTP.
     - Port: 80.
   - Click Next: Register Targets.
   - Register your EC2 instance by selecting it and clicking Add to registered.
4. Create Load Balancer:
   - Review your configuration and click Create Load Balancer.



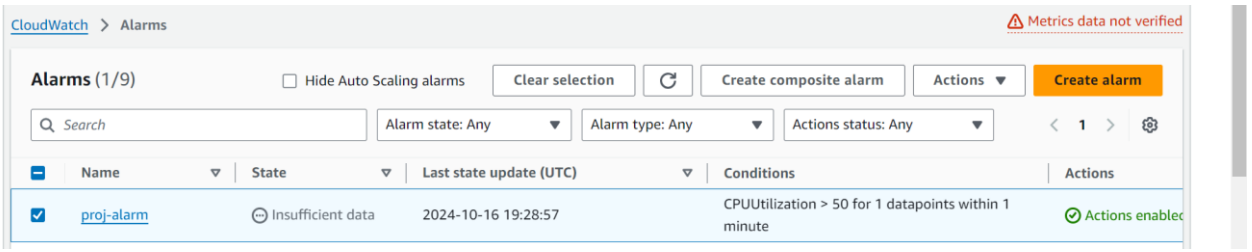
Implement Auto Scaling

1. Created an Auto Scaling Group:
   - Created Auto Scaling Group (MyAutoScalingGroup).
   - Selected the VPC and at least two subnets (one public, one private).
2. Configure Group Settings:
   - Set Desired capacity to 1, Minimum capacity to 1, and Maximum capacity to 3.
3. Configure Scaling Policies:
   - Add scaling policies (e.g., scale out if CPU utilization > 70% and scale in if < 30%).



Created Monitor Your Infrastructure with CloudWatch

1. **CloudWatch Alarms:**
   - **Click on Alarms > Create Alarm.**
   - **Selected metric and choose EC2 metrics.**
   - **Choose Per-Instance Metrics and selected my instance.**
   - **Set conditions (e.g., whenever CPU Utilization is greater than 70%).**
   - **Configure actions (send a notification to an SNS topic).**
2. **Create Dashboards:**
   - **We can create dashboards to visualize your metrics. Click on Dashboards > Create dashboard and select the metrics we want to monitor.**



Last Step to Steup Git for version Control.

Tej Mandaliya