Highly Available & Scalable Web Application on AWS (CloudFormation)

This project demonstrates deploying a highly available, auto-scaling web application on AWS using a CloudFormation YAML template.

Architecture Overview

- **EC2 Instances** (Amazon Linux 2)
- **Application Load Balancer** (ALB)
- **Auto Scaling Group** (2 to 5 instances)
- **Launch Template** with UserData to install a web server
- **Public Subnets** in 2 Availability Zones
- **Health Checks** based on ALB / path
- **Scaling Policies**:
    - Scale-out at **>60% CPU**
    - Scale-in at **<30% CPU**

Steps Performed

1. **Launch CloudFormation Stack**
    - Used the provided YAML file with:
        - AMIId: Amazon Linux 2 AMI
        - KeyName: real (EC2 Key Pair already created)
        - InstanceType: t2.micro
2. **Access the Web Application**
    - After successful deployment:
        - Go to **Outputs** in CloudFormation
        - Copy the LoadBalancerDNS value
        - Paste it in your browser to access: <h1>Welcome to Auto Scaling EC2 Instance</h1>
3. **Simulate CPU Load to Trigger Scaling**
    - SSH into one of the EC2 instances: stress --cpu 2 --timeout 300
    - This pushes the average CPU above 60% and triggers scale-out.

4. Allow Scale-In

- Wait for stress to finish (or kill the process)
- Wait 5+ minutes (grace period)
- ASG detects CPU below 30% and scales in
- Note: MinSize is 2, so no scale-in below 2

Outputs

- **ALB DNS Name**: Public URL for accessing the application
- **ASG Events**: Auto scaling logs visible in EC2 → Auto Scaling Group

Files

- **ASG.yaml: CloudFormation template to deploy the entire architecture**

Image: Web Page Served by EC2 Instance

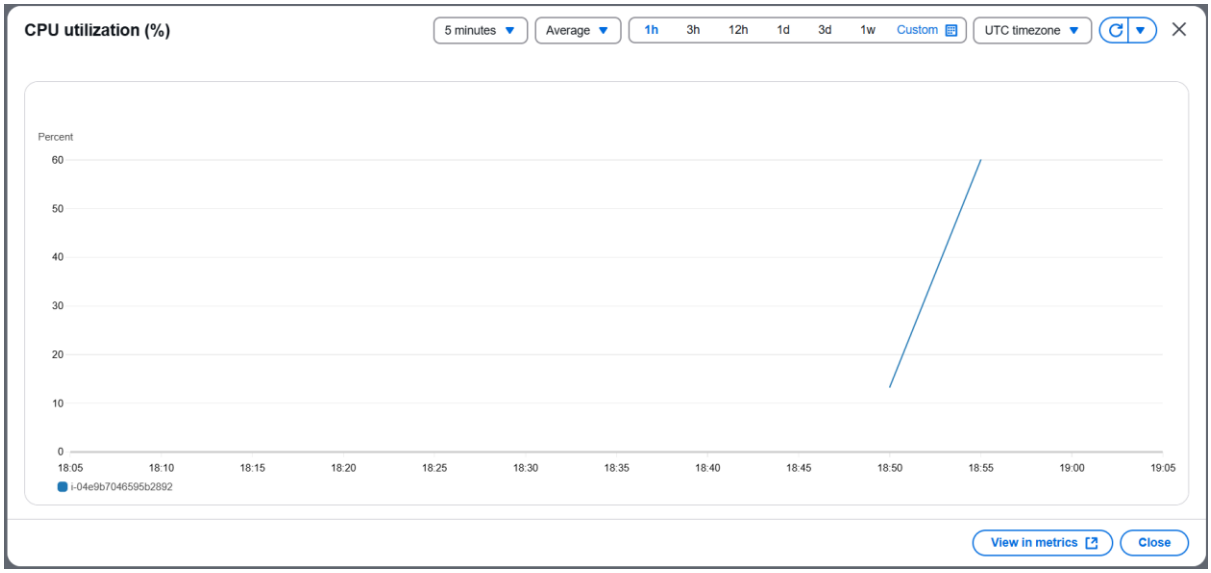

Image: CPU Utilization Triggering Scale-Out

Image: CPU Utilization Dropping After Load