Deployed a Scalable Web Application on AWS

This project demonstrates deploying a highly available and scalable web application infrastructure using Amazon Web Services (AWS). It leverages EC2 Auto Scaling, Application Load Balancer, and CloudWatch Monitoring to ensure resilience, cost efficiency, and fault tolerance.

Step 1: Prepare EC2 Instances

Launched EC2 instances and installed Apache web server. Configured the server to display hostname and IP details dynamically. Created an Amazon Machine Image (AMI) from the configured instance for Auto Scaling.

Step 2: Create a Launch Template

Defined a Launch Template (web-server-template) with the created AMI, t2.micro instance type, security group, and key pair. This template was used for launching new instances automatically.

Step 3: Create Target Group

Created a Target Group (web-target-group) to register instances and route traffic. Configured it to use HTTP on port 80 for communication with the Load Balancer.

Step 4: Application Load Balancer Setup

Deployed an Application Load Balancer (web-alb) with multiple availability zones to distribute traffic. Attached it to the target group to ensure load is evenly balanced across instances.

Step 5: Configure Auto Scaling Group (ASG)

Created an Auto Scaling Group (web-asg) using the launch template. Configured desired capacity of 2, minimum 1, and maximum 3 instances. Implemented a scaling policy based on average CPU utilization (target: 50%).

Step 6: CloudWatch Monitoring & Alarms

Configured CloudWatch alarms to monitor CPU utilization. Set thresholds (CPU > 70% for 2 minutes) to trigger scaling and notifications. Integrated with SNS for email alerts.

Outcome

Successfully deployed a fault-tolerant and scalable web application. The infrastructure is cost-optimized, self-healing, and resilient to traffic spikes. Demonstrated auto scaling by simulating high CPU load and observing new instances launching automatically.