**Course-end Project 1**

Designing Applications and Architectures in AWS

David Ahn

# Problem Statement

Your organization wants to deploy a new multi-tier application. The application will take live inputs from the employees, and it will be hosted on a web server running on the AWS cloud. The development team has asked you to set up the web server and configure it to scale automatically in cases of a traffic surge, to make the application highly available. They have also asked you to take the inputs from the employees and store them securely in the database.

Configure and connect a MySQL database instance with a web server and set up the monitoring of the solution.

# Proposed Solution

Per the requirements listed above I propose the following solution:

* A two-tier application design pattern hosted in an AWS VPC in the us-east-1 region split across 3 availability zones for fault tolerance
* Each availability zone will contain one public and one private subnet which will house the Apache/PHP web servers and MySQL database servers, respectively
* The web servers will utilize auto-scaling behind an Application Load Balancer to ensure availability and elasticity of the application should an increase in user traffic and/or failure of an availability zone occur
* The website availability will be monitored using a Route53 health check against the website URL. In this case, the A record of the ALB
* For the purposes of this project, the web servers will be connected to a single MySQL RDS database instance in a single availability zone due to cost restrictions. In an ideal case, a multi-AZ database solution would be preferred.
* SSH access to the webservers will be restricted to a Windows jumpbox hosted in one of the public subnets of the VPC

# Testing Strategy

Testing of the proposed solution will be as follows:

* Once the infrastructure is launched a simulated stress test will be performed on a web server. I will verify autoscaling scales out the web server pool.
* I will verify that the load balancer is balancing web traffic across the web servers
* I will confirm that direct SSH to the web servers is not possible
* I will confirm SSH works from the Windows jumpbox
* I will confirm the Route53 health check shows the application as available throughout the simulated stress test

# Architecture Diagram

A computer screen shot of a diagram

Description automatically generated

# Architecture Provisioning

## VPC

* Created VPC with CIDR range of 10.0.0.0/16 in the us-east-1 region consisting of:
  + 3 AZs
  + 1 public and 1 private subnet per AZ
  + Internet gateway
* Defined security groups
  + Windows-jumpbox
    - Allow RDP traffic from Internet
  + Web-server
    - Allow HTTP traffic from Internet
    - Allow SSH traffic only from the windows-jumpbox security group
  + Application-load-balancer
    - Allow HTTP traffic from Internet

## EC2

* Created launch template for the web server Autoscaling Group
  + Utilized Amazon Linux AMI with t2.micro shape
  + Assigned the default and web-server security groups
  + Populated the user data section with a bash script to perform the following:
    - Install apache and php
    - Write the hostname of the EC2 instance into the /var/www/index.html file
    - Recursively assign more secure permissions to the /var/www folder
      * Add ec2-user to the apache group
      * Assign ownership of the /var/www folder to the apache group
      * Allow group read, write permissions on /var/www
* Configured dynamic scaling policy to scale in/out based on CPU load with a threshold of 50%
  + Tested configuration by stress testing one of the webserver instances
  + Confirmed autoscaling scaled out the infrastructure
* Created an Internet-facing Application Load Balancer
  + Configured a lister to listen for HTTP traffic and forward to a target group assigned to the autoscaling group
  + Confirmed load balancer distributed traffic between the web servers in rotation
* Launched a Windows EC2 instance in a public subnet to serve as a jumpbox
  + Converted private key file from .pem to .ppk using PuttyGen
  + Confirmed SSH from Windows jumpbox to web server via Putty
  + Confirmed direct SSH denied by using cloudshell

## Route53

* Created health check against the ALB public FQDN (A Record) to monitor website availability
  + Confirmed during a stress test of the web servers the site remained available

## RDS

* Created MySQL RDS instance
  + Due to cost considerations, created a single instance in a single AZ
  + Attached the DB to one of the running EC2 web server instances to have RDS create the DB security groups automatically. Once the security groups were created, modified the launch template of the webserver ASG to include the ec2-rds-1 security group and performed an instance refresh to ensure existing and future instances would be able to connect to the DB.