**Deployment of Multi-cloud App**

Cloud Computing Bootcamp Capstone Project

David Ahn

# Problem Statement

A B2C company serving the global audience wants to ensure high availability for its end users using their web app which can be accessed from anywhere in the world. Since the web app gets multimillion traffic on a daily basis, the company cannot afford to have downtime on their app. So, they have decided to deploy the web app on two cloud platforms, on AWS and Azure to ensure that the resiliency is maintained.

# Proposed Solution

Per the requirements above I propose the following solution:

* The web application will be hosted in both AWS and Microsoft Azure
* The web servers will be placed across at least 2 availability zones per cloud platform and placed behind a load balancer to distribute load evenly to healthy instances
* Utilizing a public hosted zone in AWS Route 53, traffic will be directed both to Azure and AWS using a weighted routing policy

# Infrastructure Provisioning

## AWS

* Created a VPC utilizing 2 availability zones with a public subnet in each
* Created a security group to allow HTTP and SSH traffic to the web servers
* Created 2 Ubuntu servers and installed Apache web server
* SSH’d into the web servers and created a new document root folder named “web” under /var/www/html
  + Created an index.html file that identifies the server
* Created /etc/apache2/sites-available/web.conf to change the document root to /var/www/html/web, disabled the default conf file and enabled web.conf
* Restarted Apache to make the changes live
* Created an Application Load Balancer with a target group consisting of the newly created web servers
* Created health checks to ensure only healthy servers will get traffic routed to them
* Validated web traffic is load balanced between the web servers
* Created a Route53 public hosted zone (davidahn.org) with A records that point to both the AWS ALB and the Azure load balancer. The records utilize equal weighting to load balance between both cloud platforms evenly

## Azure

* Created 2 Windows virtual machines and configured their security groups to allow RDP and HTTP traffic
* Installed IIS on both machines and created a new index.html file in the document root that identifies each server
* Created a load balancer and placed the web servers in the backend pool
* Health checks were added to ensure only healthy instances serve traffic
* Validated load balancer serves traffic to both servers

# Testing

* Validated traffic is routed to both Azure and AWS as well as load balanced between all servers behind their respective load balancers by making multiple requests in a web browser