# Project: Kubernetes-based Website Deployment on AWS EKS

## Overview

This project involves deploying a website on a Kubernetes cluster hosted on AWS EKS. The cluster was provisioned using Terraform, ensuring infrastructure as code principles. The application stack consists of Tomcat as the application server, an NGINX Ingress Controller, and supporting services such as Memcached, RabbitMQ, and MySQL.

## Infrastructure Setup

### Kubernetes Cluster

- Created an EKS cluster with 2 nodes  
- Configured within a VPC using Terraform  
- Ensured network security and connectivity

### Terraform Configuration Files

- eks-cluster.tf - Defines the EKS cluster setup  
- vpc.tf - Configures the Virtual Private Cloud (VPC) and networking  
- main.tf - Entry point for Terraform execution  
- variables.tf - Stores configurable parameters  
- terraform.tf - Terraform provider and backend configuration  
- outputs.tf - Outputs for cluster details

## Application Deployment

### Services and Components

- Tomcat Service: Hosts the web application  
- Ingress Controller (NGINX): Manages routing and access control  
- Memcached: Caching layer for improved performance  
- RabbitMQ: Messaging broker for async communication  
- MySQL Database: Persistent data storage

### Kubernetes Deployment and Configuration

- Ingress for Tomcat App: Routes external traffic to the Tomcat service  
- Service and Deployment Files:  
 - appdeploy.yaml - Defines Tomcat application deployment  
 - appservice.yaml - Exposes the Tomcat service  
 - dbdeploy.yaml - Defines MySQL database deployment  
 - dbpvc.yaml - Configures Persistent Volume Claim (PVC) for MySQL using AWS EBS  
 - dbservice.yaml - Exposes MySQL database service  
 - mcdep.yaml - Defines Memcached deployment  
 - mcservice.yaml - Exposes Memcached service  
 - appingress.yaml - Configures the Ingress for Tomcat

## Kubernetes Commands

### Creating Kubernetes Resources

kubectl apply -f appdeploy.yaml  
kubectl apply -f appservice.yaml  
kubectl apply -f dbdeploy.yaml  
kubectl apply -f dbpvc.yaml  
kubectl apply -f dbservice.yaml  
kubectl apply -f mcdep.yaml  
kubectl apply -f mcservice.yaml  
kubectl apply -f appingress.yaml

### Checking the Status of Deployments and Services

kubectl get pods  
kubectl get services  
kubectl get deployments  
kubectl get ingress

### Viewing Logs

kubectl logs -f <pod-name>

### Installing NGINX Ingress Controller

kubectl apply -f https://raw.githubusercontent.com/kubernetes/ingress-nginx/main/deploy/static/provider/aws/deploy.yaml  
kubectl get pods -n ingress-nginx

## Checking Application Accessibility

### Creating a CNAME Record

In the hosting zone of kubevpro.azkloud.com, a CNAME record was created pointing to the external ALB of the Ingress Controller.

### Verifying the Website

curl -I http://kubevpro.azkloud.com