# Cloud-Native Dockerized Deployment of Full Stack Insurance Application using Azure VM and AKS

## 1. Introduction

In the fast-paced insurance sector, VM-based deployments posed major issues with consistency, scalability, and compliance. To resolve these, we designed and implemented a containerized, cloud-native architecture using Docker and Kubernetes on Azure infrastructure.

This project focuses on containerizing a full-stack MERN application, deploying it efficiently on Azure VM, and migrating it to Kubernetes (AKS) to achieve scalability, fault-tolerance, and rapid CI/CD capabilities.

## 2. Project Objectives

- Resolve fragmentation in VM deployments using Docker containers.
- Achieve reproducible, scalable deployment through Azure Kubernetes Service (AKS).
- Automate development to production workflow using GitHub Actions CI/CD.
- Integrate container monitoring and self-healing capabilities.
- Deliver production-grade, cloud-native architecture suitable for insurance platforms.

## 3. Implementation Overview

Full Stack Setup:

- MERN stack application with authentication, dashboard, and admin panel.
- Verified frontend-backend connectivity and MongoDB operations locally.

Dockerization:

- Multi-stage Dockerfiles for frontend (React with NGINX) and backend (Node.js).
- Docker Compose used for orchestration with MongoDB service.

- Pushed frontend image to Docker Hub: prateek2004/my-frontend

Azure VM Deployment:

- Provisioned Linux VM on Azure.
- Installed Docker, Compose, and configured NSG (Network Security Group).
- Pulled and deployed containers via Docker Compose.

CI/CD Pipeline:

- GitHub Actions triggered builds and pushed Docker images on commit.
- SSH/Webhook-based container restart implemented for Azure VM.

Kubernetes on AKS:

- Created K8s manifests for deployments, services, and ingress.
- Hosted production application on AKS for better scaling and self-healing.

Reliability Enhancements:

- Used NGINX as reverse proxy and fixed SPA routing issues.
- Deployed Watchtower for auto-updating containers from Docker Hub.
- Installed Portainer for GUI-based Docker monitoring.

## 4. Issues Faced and Solutions

SPA Routing Failure:

- Issue: NGINX returned 404 on SPA route reload.
- Fix: Added fallback routing in custom nginx.conf to redirect all paths to index.html.

Out of Memory During Build:

- Issue: Docker crashed on building React app.

- Fix: Increased heap memory with NODE_OPTIONS.

bcrypt Compatibility Error:

- Issue: bcrypt native module error in Alpine container.

- Fix: Rebuilt bcrypt inside container during image build.

MongoDB Connection Refused:

- Issue: Node app couldn't connect to MongoDB.

- Fix: Used Docker Compose service name for DB hostname.

## 5. Real-World Relevance

- Ensures faster go-to-market for insurance platforms.

- Provides a scalable, observable, and secure deployment pipeline.

- Prevents manual errors and downtime via automation and monitoring.

## 6. Project Outcomes

- Deployment time reduced by 60% post Docker adoption.

- Achieved 99.9% container uptime with Watchtower auto-restart.

- Deployment errors reduced by 70% by removing manual VM steps.

- Platform scaled horizontally on AKS.

## 7. Technical Learnings

- Deep understanding of Docker networking, volumes, caching.

- Kubernetes: Deployments, Services, Ingress, ConfigMaps.
- Azure VM & AKS provisioning and networking.
- GitHub Actions workflows, environment secrets, auto-deploy scripts.

## 8. Future Scope

- Add Prometheus + Grafana dashboards.
- Use Helm for production-grade app packaging.
- Add HTTPS with Let's Encrypt.
- Set up AKS GitHub Runner for automatic pull and rollout.
- Centralized logging with Loki or ELK stack.

## 9. Conclusion

The project modernized insurance application delivery by moving from VM-based setups to scalable, resilient, containerized infrastructure. With Docker, AKS, GitHub Actions, and observability tools in place, it provides a template for future-ready cloud-native deployment.

All YAMLs and resources will be uploaded shortly.

GitHub Repository: https://github.com/prateek200445/celebal_final_devops_project

Live App: http://172.171.199.181