Playo Case Study for Migration

Executive Summary

Introduction

Playo is a SportsTech startup, founded in Bangalore, India. Established in 2014, Playo has quickly emerged as a comprehensive sports experience platform, providing a one-stop solution for sports enthusiasts. Their mobile application caters to users' needs by facilitating connections with like-minded players, discovering suitable venues, improving skills, managing activities, and purchasing high-quality sports gear. With a user-friendly interface, Playo allows users to book football and cricket grounds, rent badminton courts and racquets, and more, all at their fingertips.Playo required a microservices-based migration approach, migrating their infrastructure from GCP to AWS by establishing VPC Peering. They also required the adoption of AWS DevOps best practices, including automated testing, deployment, and monitoring to improve application and infrastructure stability and agility.

Customer Challenges

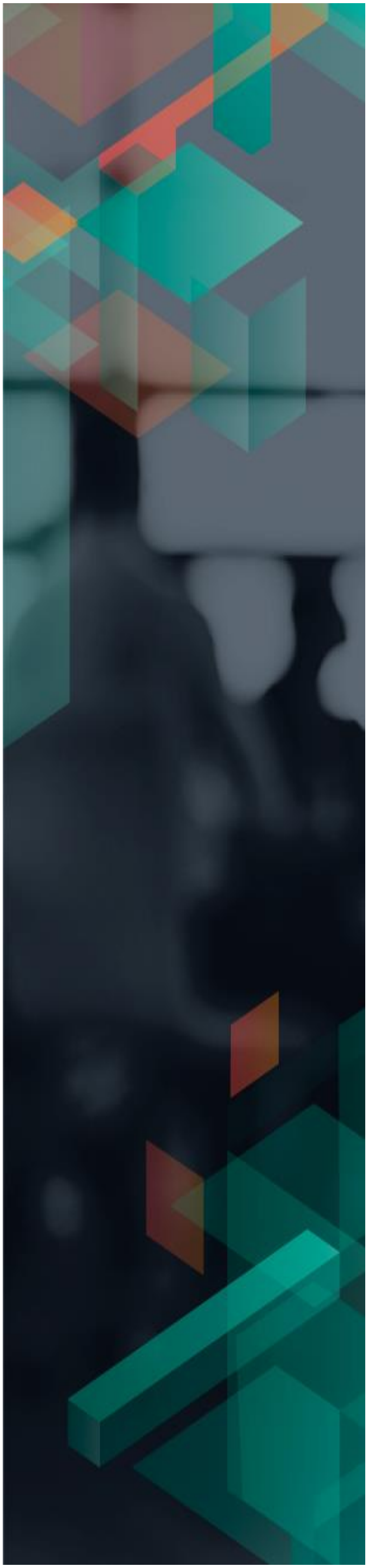
The client faced several critical challenges when migrating from Google Cloud Platform (GCP) to Amazon Web Services (AWS), including transferring a containerized web application with over 30 microservices hosted on Google Kubernetes Engine (GKE). The client faced issues with scalability and observability for the infra and application monitoring with existing GCP architecture. From a DevOps standpoint, the migration had specific primary objectives in mind. These objectives revolved around optimizing system availability, cost reduction, and facilitating a faster development lifecycle through the implementation of continuous integration/continuous delivery (CI/CD) using AWS services. By leveraging the capabilities provided by AWS, the migration sought to enhance the overall availability of the system, streamline the development process, and achieve cost efficiencies. The adoption of CI/CD practices aimed to enable a continuous and automated delivery pipeline, allowing for faster and more frequent releases of software updates, bug fixes, and new features. Ultimately, these efforts were aimed at maximizing system performance, reducing operational expenses, and accelerating the pace of software development.

Solution

- The migration process involved transferring existing resources from GCP to AWS while ensuring minimal downtime across multiple environments.
- The migration also required the adoption of AWS DevOps best practices, including automated testing, deployment, and monitoring to improve application and infrastructure stability and agility.
- VPC design with Bastion Host and NAT gateway was provisioned and implemented to ensure network security

About Playo

Playo is a thriving SportsTech startup revolutionizing the sports industry through its mobile application. Offering a comprehensive range of services, Playo connects sports enthusiasts, facilitates venue bookings, skill enhancement, activity management, and sports gear purchases. Headquartered in Bangalore, Playo has expanded its operations to India, UAE, and Qatar, with pilot projects underway in the UK and Australia. With a customer base exceeding three million user relationships, Playo powers over 150,000 monthly sports activities.



- MongoDB database migration was done by setting up multi-node failover MongoDB with active-active configuration on AWS EC2 instances from GCP VM.
- AWS ECR was used as a container registry and the container images deployed on AWS ECR were specifically used to deploy them on Amazon EKS.
- Application performance monitoring was implemented using Xray with CloudWatch.
- Open search and CloudWatch log streams achieved centralized logging solutions across environments.
- Monitoring and alerting for CPU & memory utilization of Nodes and infra monitoring for EKS and EC2 were implemented using Prometheus (self-managed) and CloudWatch.
- AWS WAF was used to monitor HTTP and HTTPS requests forwarded to protected web application resources through the load balancer.
- Third Party applications used:
    - **Argo CD** is a Kubernetes controller responsible for continuously monitoring all running applications and comparing their live state to the desired state specified in the Git repository.
    - **Bitbucket Cloud** is a Git-based code and CI/CD tool optimized for teams using Jira.
    - **Prometheus** is an open-source system monitoring and alerting toolkit that supports features like having a multi-dimensional data model with time series data identified by metric name and key/value pairs. In addition, we have enabled metrics from EKS Fargate to have metrics on memory CPU and disk utilization and alerting system integrated with email.
    - **Grafana** is the most prominent tool in application monitoring and analytics. It is an open-source analytics and interactive visualization web application.

Architecture Diagram

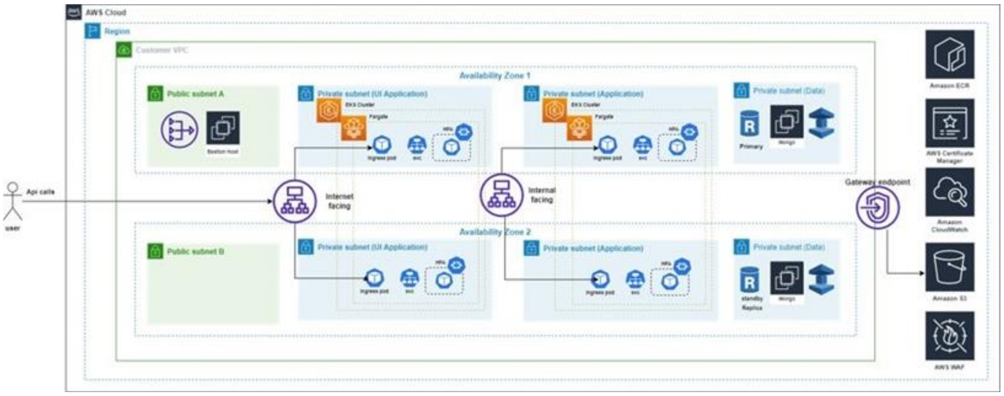


Fig 1: Playo AWS Infrastructure diagram

The above architecture diagram explains a three-tier architecture implementation where the bastion host is in a public subnet, and application servers and database servers are hosted in private subnets on EKS. The above approach provides a modular, scalable, highly available, fault-tolerant secure system on the EKS Containerization platform.

Conclusion

Successfully migrated all from GCP to AWS in 14 working days for the dev environment and successfully replicated the stage and prod environments. The multi-environment setup for the web application platform is deployed, which satisfies the given organizational SLAs and internal process framework to help business growth. Deploying containerized workload on EKS, which is 25% cheaper than the existing architecture in each environment, by leveraging spot instances With no downtime, we have migrated and deployed a highly available failover setup for MongoDB on the AWS EC2 instance. DevOps best practices implemented to support new feature releases with zero downtime and deployments spanning within minutes. Around 15 releases are made by each service with zero time and save about 30 minutes of manual effort for every deployment. Fast, accurate, and highly available, secure application for high customer traffic on mobile applications, which is 20% more than existing traffic. Successfully incorporated microservices with DevOps Best Practices in coordination with developer teams. Better observability with application performance monitoring tool leveraging AWS X-ray provided real-time traceability, which was not available.