

# AWS DevOPS





# WS, Oracle + SQL server + Weblogic + Fargate Migration from on-premises to Amazon Web Services

## Project Goals:

- Migrate a mission-critical system comprising Java containerized applications, JavaEE Weblogic-based application, Oracle DB and MS SQL Server DB from the company's premises to AWS.
- Migrate databases to AWS RDS.
- Implement serverless container hosting with AWS Fargate.
- Setup system monitoring with Cloudwatch.
- Design and deploy Business Continuity solutions.
- Develop standard support operation procedures.

## Accomplishments:

- The system has been migrated within 2 hours of downtime.
- A cross-region Disaster Recovery solution has been implemented, comprising DB replication, Container/EC2 synchronization, and S3 buckets sync.
- Cloudwatch setup has been customized to suit corporate monitoring requirements. 17 custom metrics written.
- 8 SOP documents have been developed and tested

## Technologies:

AWS EC2, AWS Fargate, AWS RDS, Cloudwatch, AWS S3, Oracle Database, Microsoft SQL Server, Red Hat Enterprise Linux, Windows Server

## Workload:

350 hrs

## Project Team:

Amazon Web Services Architect  
2 x Senior DevOps Engineer  
Senior Microsoft SQL Server DBA  
Senior Oracle DBA



# WS, Jenkins CI/CD build from GIT

## Project Goals:

- Build a CI/CD pipeline for an AWS based web application, using Jenkins and Git repository
- The application comprises: AWS RDS (Oracle and SQL server), Fargate, EC2 Linux servers with Weblogic server

## Accomplishments:

- Build and deploy pipelines have been configured with Jenkins, to fetch the new code from Git and deploy to Weblogic server, Fargate containers, and RDS DB
- User access model has been implemented in Jenkins to ensure secure access to Production build/deploy functions

## Technologies:

Jenkins, Git, AWS EC2, AWS Fargate, AWS RDS, Oracle Database, Microsoft SQL Server, Red Hat Enterprise Linux, Windows Server

## Workload:

90 hrs

## Project Team:

Senior DevOps Engineer



# WS, AWS Infrastructure Cost Optimization

## Project Goals:

- Analyze the client's cloud-based infrastructure, consisting of AWS EC2, Fargate and RDS services
- Analyze the services' resource usage from different angles and find opportunities to optimize costs

## Accomplishments:

- RDS (Oracle and SQL server), Fargate, EC2 CPU/Memory/IO usage has been analyzed – to propose more optimal instance sizes
- A shutdown policy has been implemented to save resources during off-hours
- Savings plan has been implemented to optimize the costs of the existing resources
- RDS Oracle and MS SQL Server SQL queries have been optimized to lower the actual CPU/Memory usage

## Technologies:

AWS EC2, AWS Fargate, AWS EKS, AWS RDS, Oracle Database, Microsoft SQL Server, Red Hat Enterprise Linux, Windows Server

## Workload:

80 hrs

## Project Team:

Senior Oracle DBA

Senior MS SQL Server DBA

Senior AWS Architect



# Ultra low-latency AWS multi-region cloud infrastructure

## Project Goals:

Build low-latency and stable AWS multi-region cloud environment

## Accomplishments:

- Configured channels and routes with minimum latency for the current distances. Redundancy was used.
- Created Network ACLs

## Technologies:

AWS

## Workload:

40 hrs

## Project Team:

Developers, DevOps

## Clients:

FinTech Startup

## Outcome:

Opportunity to run his FinTech business trading from Tokyo in New York via low-latency servers connection





# Cost reduction for Mobile App Cloud environment

## Project Goals:

Implement cloud environment for the Company's App, security optimizations, cost reductions

## Accomplishments:

- Created AWS Environment
- Setted up CI/CD
- Cost Optimization
- Deployed K8s clusters
- Used best security practices
- Deployed HA architecture

## Technologies:

AWS, K8s, CircleCI, Docker, Github

## Workload:

160 hrs

## Project Team:

Developers, DevOps Engineers, PM

## Client:

NFT mobile application with 300,000+ users

## Outcome:

The number of downtimes of the App is reduced several times. Cost Optimized up to 30%. The time to implement the new version of the application has been reduced by 50%.



# Server Security Optimization

## Project Goals:

Security optimization

## Accomplishments:

- All AWS Services have been migrated to private networks
- Lambda functions have been fixed to use build-in environment variables
- Access has been restricted to services and functions
- ETL processes now works in an isolated environment

## Technologies:

AWS Redshift, AWS Glue, AWS S3, AWS IAM, AWS VPC, AWS Lambda

## Workload:

60 hrs

## Project Team:

Developers, DevOps Engineers

## Client:

Analysis and Reporting Software Product

## Outcome:

Best Security Practices was implemented, re-designed AWS Cloud Environment, transformed ETL Processes