AWS CICD Delivery

Case Studies - 2

Introduction

The client is an IT consulting firm and focusing on SAAS based entertainment applications. They started their business in 2021 and covered entertainment industries mostly in India and some global customers. Now expanding their business concentrating more on Europe, America and Australia.

Challenges

- The client was in an On premises environment and maintained Databases in MongoDB to server their global customer.
- They faced difficulties in maintaining the physical servers and they felt it was not reliable when connecting in a different location.
- Elasticity is one of the challenges they faced during upgrading the infrastructure.
- Since it is paid services, when the service goes down they get
- TCO was higher on procurement of hardware, shipment and maintenance.
- They faced challenges on updating the applications from version 1 to version 2

Our Solutions

- Virtual Private Cloud created in AWS with Private and Public Subnets with Multiple Available Zones.
- Client's Application uses ReactJS and NodeJS, Java as API, Mysql and MongoDB as Database.Deployed RDS instance for Mysql database in private subnet.
- Created version1 and version2 in separate instance and docker installed and Admin application is hosted in separate ECS cluster and pulled ECR image for each modules(version1, version2 and Admin panel).
- Application Load Balancer deployed in front of ECS clusters.
- Tool configuration in Jenkins for Deployment.
- Created separate slave (server) for every module deployment, then connect slave using the server credentials.

- Each pipeline for every module uses groovy script.
- Set three stages for every pipeline in both V1 and V2 (Checkout, Build, Deployment).
- Delivering and Deploying the project using Docker.
- All GitHub Repositories are directly connected in Jenkins. Every changes made in github, pipeline will be triggered since the Github webhook is enabled in Jenkins setup.
- Any error in build stage, Deployment stage can't impact on that change.
- In production, we are using AWS Code Pipeline, instance Jenkins due to cost optimization since the build will often be tested in the Development phase.
- Completed Deployment using AWS ECS.
- Compiling the programming using AWS Code Build.
- For continuous deployment using AWS Code Deploy and AWS Code Pipeline.
- Enabled CloudWatch monitoring for real time monitoring the resources.

Benefits

- Optimizes costs with an overall reduction of 28%.
- Highly available infrastructure with above 95% SLA.
- High Scalable infrastructure in nature within short time.
- Improved user experience.
- Saved human time to maintain the infrastructure and reduced business downtime for maintenance activities.
- Real time monitoring is straightforward.
- Backup and retrieval process made simpler.
- Reduces transfer cost using CloudFront and endpoints.
- Easy for application version updates.

Services Used

- AWS Virtual Private Cloud.
- AWS Elastic cloud compute.
- AWS CloudWatch.
- Data Lifecycle management.
- AWS Elastic Block Storage.
- AWS Simple Storage Services.
- AWS CloudFront.
- AWS Web Application Firewall.
- GitHub.
- Docker.
- Jenkins.
- Code Build.
- Code Pipeline.