Migrating CI/CD pipelines for a futureproofed infrastructure

With a strategic decision to migrate the majority of their workloads to AWS, a large pan-African telecommunications company required BBD's assistance in planning, architecting and modernising the workloads for their digital channels while leveraging CI/CD pipelines, microservices and managed AWS services that are not available on-premises.

Objectives

- Migrate an existing BBD-built system and CI/CD pipelines to Amazon Web Services (AWS)
- Leverage microservices in the migration using an Elastic Kubernetes Service (EKS) cluster that hosts the CI/CD pipeline that is used by hundreds of engineers for deployments across the innumerable projects
- Ensure scalable workloads in order to manage the ongoing, and sometimes, unexpected demand
- Deploy a fault-tolerant and highly available environment that meets defined security goals

Benefits

Time to deploy:
Using EKS we were able to deploy the environment very quick timeframe, while the built-in functionality of EKS allowed us to automatically scale the cluster to generate additional nodes as more resources are required to meet user demands

Cost savings:
Leveraging SQS/Lambda we were able to drive down operational overhead and costs. SQS/Lambda offered the performance of items per second while proving the durability needed

Recovery time from failure:
With minimal to no user interaction, we were able to make the recovery time from an AZ failure far faster than what would have been possible in a traditional setup

Overview of the solution

The overarching goal in this project was to move the existing CI/CD stack to AWS infrastructure, because re-architecting the working CI/CD pipeline itself was unnecessary.

The migration of this system was from on-premises statically sized VMs and Docker images to an EKS cluster that is hosted in AWS. The approach was to ensure that the creation, upgrades and maintenance of the cluster is done through infrastructure-as-Code (IaC) principles with Terraform being the main tool to manage AWS resources.Utilising IaC allows for less errors, automated changes, standardised configurations ad the ability to track infrastructure.

Container management for the project is handled in two parts:

- Cluster components such as ArgoCD and logging operators are cached in and pulled from ECR (an automated, real-time case reporting exchange system)

- Custom-built artefacts are stored in JFrog Artifactory which itself runs in the CI/CD cluster. Here, containers for all applications built via the CI/CD toolset are stored and managed. The containers are then pulled by other clusters to host the applications. JFrog allowed the team to host various types of repos such as Docker, Nuget and Maven, all within a single application. This was the right decision for this project as it meant that the team does not have to manage multiple repos and the security associated with each, thereby allowing for increased productivity. Highly available to boot and supportive of all possible back-end databases, JFrog made the perfect solution for these mission-critical workloads.

All container orchestration was handled via AWS EKS as it takes in account all other scheduling and management of containers. It also has the added benefit of making scaling for effective while making the handling capacity of issues easier.

Impact of BBD's partnership

In a short timeframe, utilising AWS infrastructure, microservices, EKS clusters and managed services, we were able to effectively migrate and modernise the workloads in the AWS cloud. The project resulted in a faster recovery time from failures for the client software engineering teams across their projects, reduced operational overheads and costs, improved scalability and futureproofing.

bbd
software development

Get in touch

If you'd like to engage with us, we'd love to hear from you.

✉ info@bbdsoftware.com   www.bbdsoftware.com