

Maestro Application On-Premise to AWS



□ 11 January 2023 □ amlgo □ Case-Study

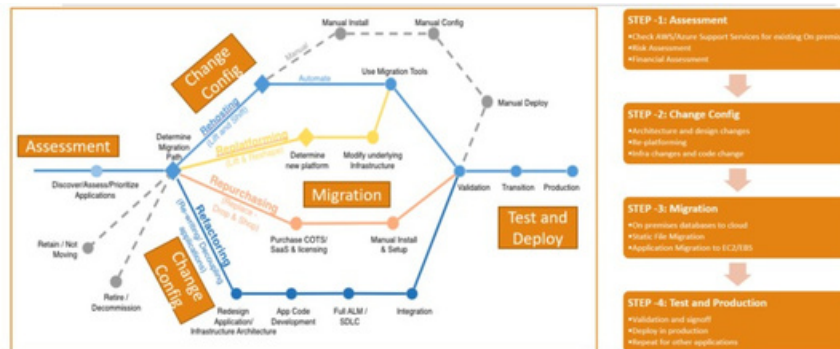
An end-to-end cloud migration by Amlgo Labs

Background:

Our client is one of the top financial services provider firms from Europe having 100 + applications to support day-to-day banking businesses around the globe. The team approached Amlgo Labs for helping them architect and design the migration strategy for one of the applications called *Maestro* from on-premises to AWS cloud. This includes migration of – Applications, Databases, Security, Permissions, etc. with minimum service disruption and migration as efficiently as possible.

Proposed Solution:

Amlgo team's predefined approach has been key to the successful migrations of many projects and played a key role in designing and defining the end-to-end process of migration. The Approach –



Step-1: Assessment

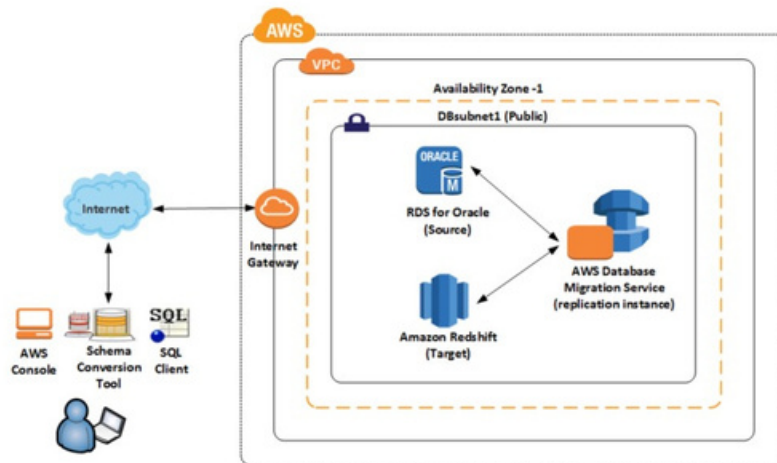
- The AWS Migration Readiness Assessment: a process of gaining insights into how far along we are in the cloud journey, understanding current cloud readiness strengths and weaknesses, and then building an action plan to close gaps identified.
- AWS Cloud Adoption Framework (CAF): a guide to get a holistic view of the transformation initiative that is required for an effective move to the cloud.

Step-2: Change Configuration

During this step of cloud migration, the Maestro application undergoes architectural and design changes. This step is the longest step as it includes many designing and re-platforming sessions at a granular level until the application achieves the desired state. These changes can include implementation change, code change, architectural change, underlying infrastructure change, and so on.

Step-3: Migration

Once the implementation is ready, we first need to migrate the static files, dependencies, and database tables to the respective AWS services. This includes steps like migrating on-premise databases (MySQL, Oracle, Sybase) to Amazon RDS/Aurora, migrating static files and dependencies to Amazon S3.



This is a time-consuming process as legacy applications might have a huge amount of data in their databases. Once the cloud migration is done, a comparison check is required to validate if the data between the on-premise database and Amazon RDS match.

Post that actual migration of the application and deployment happens on Amazon services like EC2/EBS along with allocating load balancers to them and maintaining their scalability. Since the application is containerized, we also need to make an entry in ECR and deploy on AWS Fargate.

Step-4: Test and Production

The final step involved testing the sustainability of the Maestro application, so this is essentially a validation step where we test the application to ensure that they work seamlessly. Here we compare the performance of cloud-migrated applications against their on-premise

version to determine if the cloud migration was successful, few other steps were processed during this –

Agile development practices (DevOps, test automation, CI/CD, observability).

Infrastructure automation (elastic infrastructure, containers, AI/ML).

Future Extension Planning to the Cloud-native architectural patterns (stateless, microservices, serverless, data lakes).

Windows modernization, database modernization, and product-based operating models.

Conclusion:

Amlgo team successfully transforms the Maestro application from on-premise to AWS with an exceptional scalable design and enhanced security implementations.

As we always emphasize that cloud migration doesn't have to happen all at once. You can migrate services in phases or waves grouped by service or user. A phased cutover approach allows for rollback points if things don't go according to plan, reducing migration risk.

Ensuring data security in the cloud may require extra steps. Cloud security challenges may require extra focus in your cloud migration effort. During this transformation, we created the additional security layers their business required.

Cloud capabilities can improve disaster recovery planning. As we wanted to ensure that the client should be better prepared to quickly recover customer data in the event of a disaster. AWS' reliable, redundant, and robust data backups help make this possible.