Migration: Assessment & Planning

Customer: One of the largest low-cost airlines

Summary

The customer is UAE's aviation corporation catering over 70 million passengers as of date. Passenger service system (PSS), their ticket booking application was a legacy system that they intended to migrate to a cloud environment while ensuring they manage to leverage administered services of cloud by conducting a Migration Readiness Assessment & Planning (MRAP).

Problem Statement

Passenger Service System (PSS) was the existing ticket booking application for the customer. The objective was to understand this legacy system and then recommend how it can be migrated to AWS while leveraging the cloud-native capabilities via an MRAP assessment. The focus would be application modernization rather than a lift & shift migration to the cloud. The customer team intends to leverage managed services of cloud and go serverless, containers, open source etc. wherever possible. The customer team also wants to move away from the commercial Oracle database to a more open-source AWS Aurora PostgreSQL database due to the high licensing costs imposed by Oracle.

MRAP is critical to any organization that plans to adapt to the cloud as this tool-based assessment checks their application's ability to cloud. Powerup was approached to perform MRAP on their existing set up to propose a migration plan as well as a roadmap, post its analysis.

Proposed Solution

The customer's MRAP Process

To begin with, the RISC Networks RN150 virtual appliance, an application discovery tool that poses as an optional deployment architecture was configured and installed on the customer's existing PSS Equinix data centre (DC) to collect data and create a detailed tool-based assessment to understand the existing set up 's readiness to migration.

Application stacks were built for the applications in scope and assessments as well as group interviews were conducted with all stakeholders. Data gathered from stakeholders were cross-verified with the information provided by the customer's IT and application team to bridge gaps if any. Powerup team would then work on creating a proposed migration plan and a roadmap.

MRAP Deliverables

A comprehensive and detailed MRAP report included the following information:

Existing overall architecture

The existing PSS system was bought from a vendor called Radixx International, which provided three major services:

- ***Availability service***, an essential core service mainly used by online travel agencies (OTAs), end-users and global distribution system (GDS) to check the availability of their customer's flights. It's base system contained modules like Connect Point CP (core), payments, the enterprise application (Citrix app) all written in .NET and the enterprise application for operation and administration written in VB6.
- ***Reservation service*** was used in booking passengers' tickets where data was stored in two sessions, Couchbase and the Oracle database. The webpage traffic was 1000:1 when compared to availability service.
- ***DCS System (Check-in & Departure Control Systems)*** is another core system of any airline, which assists in passenger check-in, baggage check-in and alerting the required officials. It is a desktop application used by airport officials to manage passengers from one location to another with the availability of an online check-in module as well.

Existing Database: Oracle is the current core database that stores all critical information consisting of 4 nodes – 2 Read-Write nodes in RAC1 & another 2 (read-only nodes) in RAC2. All availability checks are directed to the read-only Oracle nodes. The Oracle database nodes are heavily utilized roughly at 60-70% on an average with currently 14 schemas within the Oracle database accessed by the various modules. Oracle Advanced Queuing is used is some cases to push the data to the Oracle database.

Recommended AWS Landing zone structure

The purpose of AWS Landing Zone is to set up a secure, scalable, automated multi-account AWS environment derived from AWS best practices while implementing an initial security baseline through the creation of core accounts and resources.

The following Landing Zone Account structure was recommended for the customer:

***AWS Organizations Account*:**
Primarily used to manage configuration and access to AWS Landing Zone managed accounts, the AWS organizations account provides the ability to create and financially manage member accounts.

***Shared Services Account*:**

It is a reference for creating infrastructure shared services. In the customer's case, Shared Services Account will have 2 VPCs – one for management applications like AD, Jenkins, Monitoring Server, Bastion etc. and other Shared services like NAT Gateway & Firewall. Palo Alto Firewall will be deployed in the shared services VPC across 2 Availability Zones (AZ)s and load balanced using AWS Application Load Balancer.
AWS SSM will be configured in this account for patch management of all the servers. AWS Pinpoint will be configured in this account to send notifications to customer – email, SMS and push notifications.

Centralized Logging Account:
The log archive account contains a central Amazon S3 bucket for storing copies of all logs like CloudTrail, Config, CloudWatch logs, ALB Access logs, VPC flow logs, Application Logs etc. The logging account will also host the Elasticsearch cluster, which can be used to create custom reports as per customer needs, and Kibana will be used to visualize those reports. All logs will be pushed to the current Splunk solution used by the customer for further analysis.

***Security Account*:**
The Security account creates auditor (read-only) and administrator (full-access) cross-account roles from a security account to all AWS Landing Zone managed accounts. The organization's security and compliance team can audit or perform emergency security operations with this setup and this account is also designated as the master Amazon GuardDuty account. Security Hub will be configured in this account to get a centralized view of security findings across all the AWS accounts and AWS KMS will be configured to encrypt sensitive data on S3, EBS volumes & RDS across all the accounts. Separate KMS keys will be configured for each account and each of the above-mentioned services as a best practice.

Powerup recommended Trend Micro as the preferred anti-malware solution and the management server can be deployed in the security account.

***Production Account*:**
This account will be used to deploy the production PSS application and the supporting modules. High availability (HA) and DR will be considered to all deployments in this account. Auto-scaling will be enabled wherever possible.

UAT Account – Optimized Lift & Shift:
This account will be used to deploy the UAT version of the PSS application. HA and scalability are not a priority in this account. It is recommended to shut down the servers during off-hours to save cost.

***DR Account*:**

Based on the understanding of the customer's business a Hot Standby DR was recommended where a scaled-down version of the production setup will be always running and will be quickly scaled up in the event of a disaster.

***UAT Account – Cloud-Native*:**

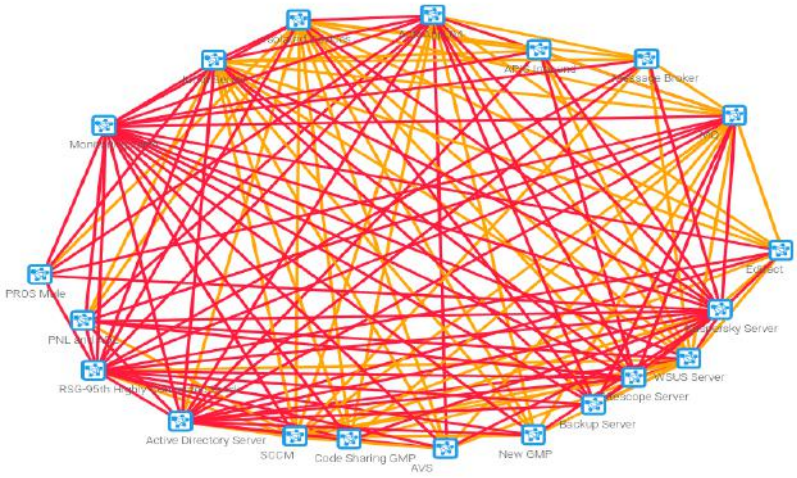
The account is where the customer's developers will test all the architectures in scope. Once the team has made the required application changes, they will use this account to test the application on the cloud-native services like Lambda, EKS, Fargate, Cognito, DynamoDB etc.

Application Module - Global Distribution Systems (GDS)

A global distribution system (GDS) is one of the 15 modules of the PSS application. It is a computerized network system that enables transactions between travel industry service providers, mainly airlines, hotels, car rental companies, and travel agencies by using real-time inventory (for e.g., number of hotel rooms available, number of flight seats available, or number of cars available) to service providers.

- The customer gets bookings from various GDS systems like Amadeus, Sabre, Travelport etc.
- ARINC is the provider, which connects the client with various GDS systems.
- The request comes from GDS systems and is pushed into the IBM MQ cluster of ARINC where it's further pushed to the customer IBM MQ.
- The GMP application then polls the IBM MQ queue and sends the requests to the PSS core, which in turn reads/writes to the Oracle DB.
- GNP application talks with the Order Middleware, which then talks with the PSS systems to book, cancel, edit/change tickets etc.
- Pricing is provided by the Offer Middleware.

Topology Diagram from RISC tool showing interdependency of various applications and modules:



Any changes in the GDS architecture can break the interaction between applications and modules or cause a discrepancy in the system that might lead to a compromise in data security. In order to protect the system from becoming vulnerable, Powerup recommended migrating the architecture as is while leveraging the cloud capabilities.

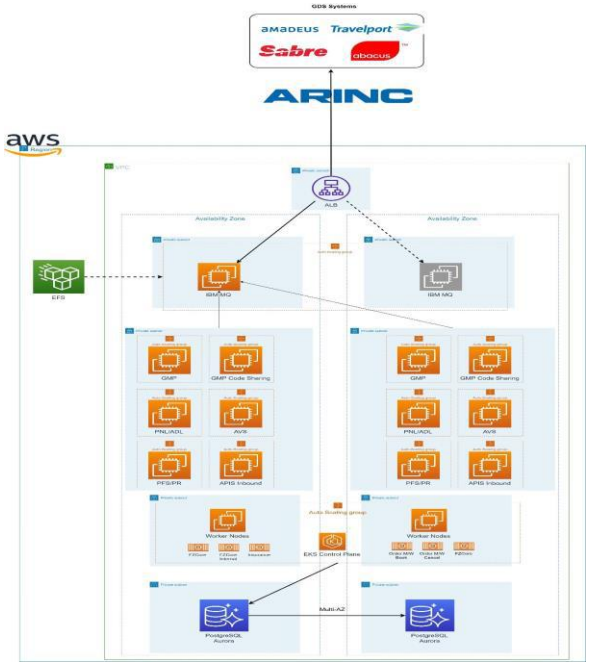
Proposed Migration Plan

IBM MQ cluster will be setup on EC2, and auto-scaling will be enabled to maintain the required number of nodes thus ensuring availability of EC2 instances at all times. IBM MQ will be deployed in a private subnet.

Amazon Elastic File System (Amazon EFS) will be automatically mounted on the IBM MQ server instance for distributed storage, to ensure high availability of the queue manager service and the message data. If the IBM MQ server fails in one availability zone, a new server is created in the second availability zone and connected to the existing data, so that no persistent messages are lost.

Application Load Balancer will be used to automatically distribute connections to the active IBM MQ server. GMP Application and PNL & ADL application will be deployed on EC2 across 2 AZs for high availability. GMP will be deployed in an auto-scaling group to scale based on the queue length in the IBM MQ server and consume and process the messages as soon as possible whereas PNL & ADL to scale out in case of high traffic.

APIS Inbound Application, AVS application, PSF & PR application and the Matip application will all be deployed on EC2 across 2 AZs for high availability in an auto-scaling group to scale out in case of high traffic.

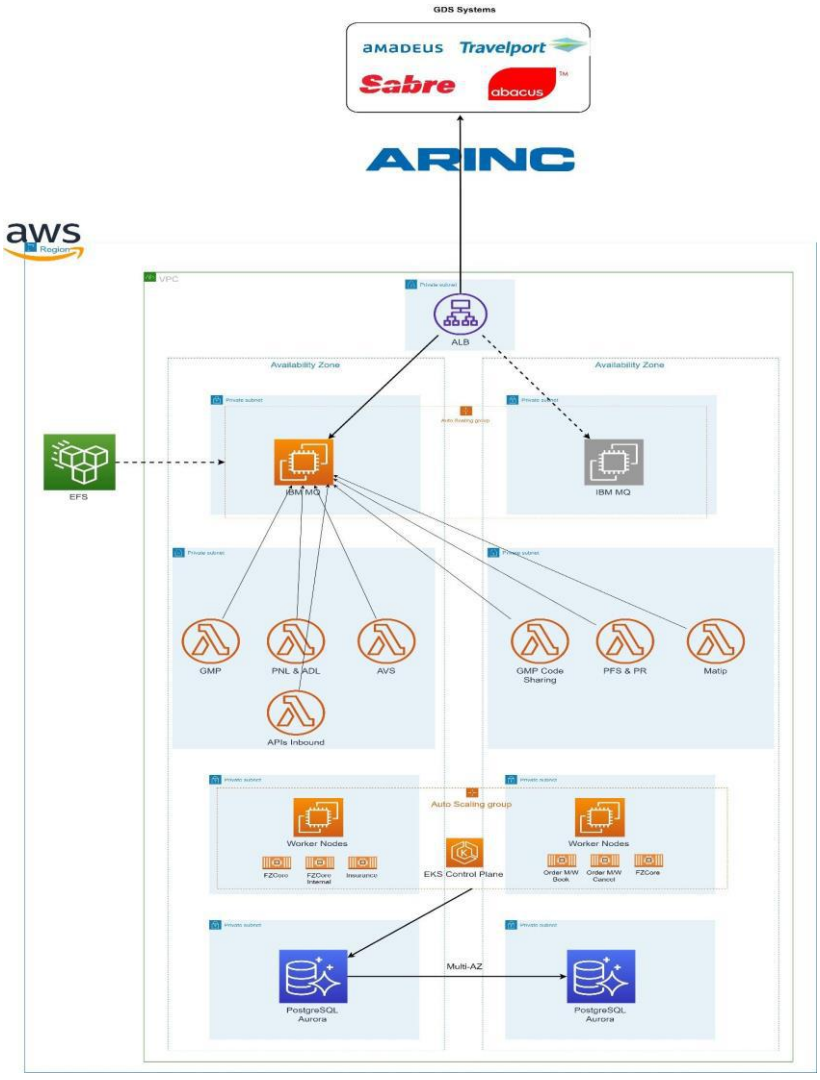


[Optimized Lift & Shift Architecture]

Cloud-Native Architecture

- GMP and GMP code sharing applications will be deployed as Lambda functions. The lambda function will run when a new message comes to the IBM MQ.
- PNL & ADL application will be deployed as a Lambda function and the function will run when there is a change in the PNR number in which case a message must be sent to the airport.
- AVS application will be deployed as Lambda functions where it will run when a message will be sent to the external systems.
- Matip application will be deployed as a Lambda function and will run when a message will be sent using the MATIP protocol.
- PFS & PR application will be deployed as Lambda functions. The lambda function will run when a message will be sent to the airport for booking.
- APIS Inbound application will be deployed as a Lambda function and it will run when an APIS message will be sent to the GDS systems.

For all the above, required compute resources will be assigned as per the requirement. Lambda function will scale based on the load.

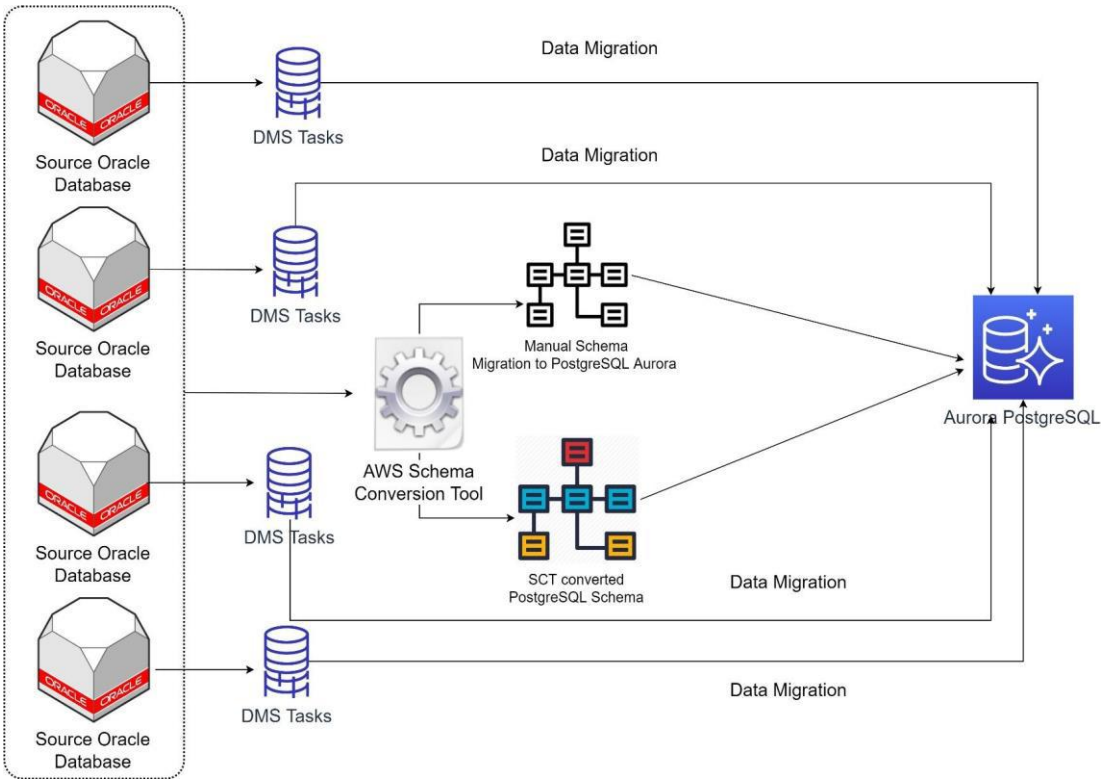


Application modifications recommended

All the application components like GMP, AVS, PNL & ADL, PFS & PR, Matip, etc are currently in .NET. which have to be moved into .NET Core to be run as Lambda functions. The applications are recommended to be broken down into microservices.

Oracle to Aurora Database Migration

AWS schema conversion tool (SCT) is run on the source database, which will generate a schema conversion report that will help understand interdependencies of existing schemas, and how they can be migrated on to Aurora PostgreSQL. The report will contain database objects some that can be directly converted by the SCT tools and the rest, which would need manual intervention. For Oracle functionalities that are not supported in Aurora PostgreSQL, the application team must write custom code to migrate those. Once all the schemas are migrated, AWS Database Migration Service will be used to migrate the entire data set from Oracle to Aurora.



Oracle to Aurora-PostgreSQL Roadmap

- ***Lift & shift:***
  The current Oracle database will be moved to AWS as-is without any changes in order to kick-start the migration. The Oracle database can run on AWS RDS service or EC2 instances. One RDS node will be the master database in read/write mode. The master instance is the only instance to which the application can write to. There will be 3 additional read-replicas spread across 2 AZs of AWS to handle the load that is coming in for read requests. In case

the master node goes down one of the read replicas is promoted as the master node.

- ***Migrate the Oracle schemas to Aurora:***

Once the Oracle database is fully migrated to AWS, the next step is to gradually migrate the schemas one by one to Aurora – PostgreSQL. The first step is to map all the 14 schemas with each application module of the customer. The Schemas will be migrated based on this mapping, wherever there are non-dependent schemas on other modules, it will be identified and migrated first.

The application will be modified to work with the new Aurora schema. Any functionality, which is not supported by Aurora, will be moved to application logic.

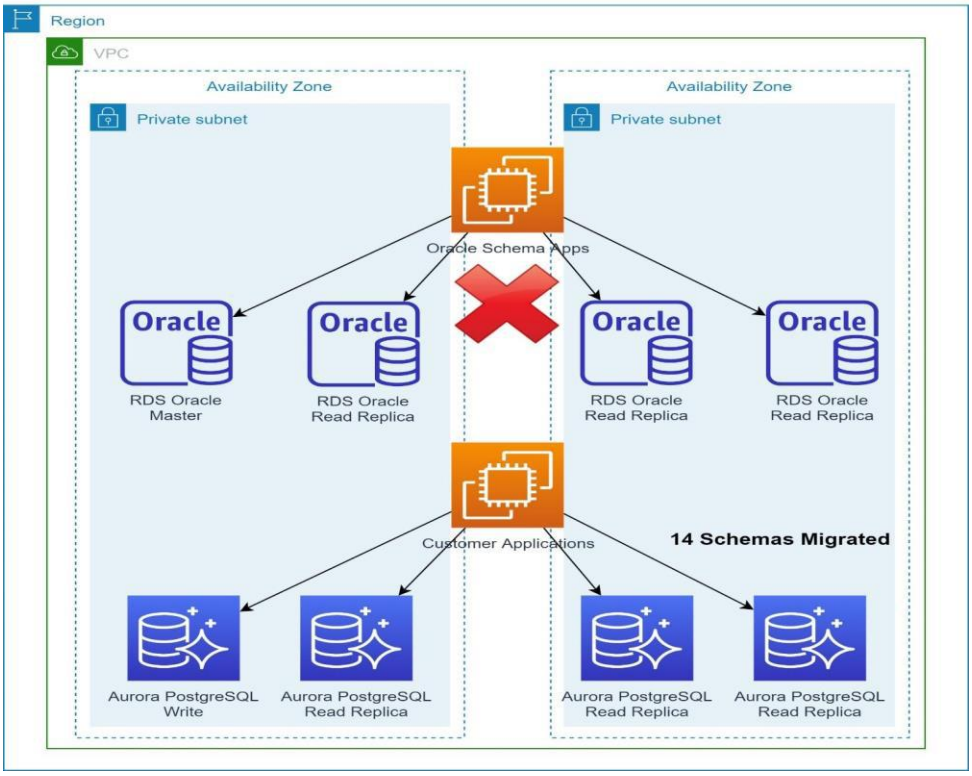
DB links can be established from Oracle to Aurora, however, it cannot be established from Aurora to Oracle database.

Any new application development that is in progress should be compatible and aligned with the Aurora schema.

- ***Final Database:***

Finally, all the 14 schemas will be migrated onto Aurora and the data will be migrated using DMS service. The entire process is expected to take up to 1 year. There will be 4 Aurora nodes – One Master Write & 3 Read Replicas spread across 2 AZs of AWS for high availability.

Final Database Architecture



Key Findings

The assessment posed as a roadmap to move away from Oracle to PostgreSQL saving up to 30% in Oracle License cost. It also provided a way forward for each application towards cloud-native. Current infrastructure provisioned was utilized at around 40-50% and a significant reduction in the overall total cost of ownership (TCO) was identified if they went ahead with cloud migration. Less administration by using AWS managed services also proved to be promising, facilitating smooth and optimized functioning of the system while requiring minimum administration.

With the MRAP assessment and findings in place, the customer now has greater visibility towards cloud migration and the benefits it would derive from implementing it.

Cloud platform

AWS.

Technologies used

EPS, ALB, PostgreSQL Aurora, Lambda, RDS Oracle, VPC.