
CASE STUDY

Application Migration to Cloud

Introduction

EFMFM is an enterprise telematics end-to-end ERP product by NGFV to manage complete employee transportation process for large enterprise clients encompassing fleet management, intelligent routing, live tracking, compliance and security.

Aon Hewitt, a customer of NGFV user of EFMFM from 500 offices in 120 countries, it provided consulting, outsourcing, and reinsurance brokerage services. Employee transportation is an extremely critical service provided by the company in a shift-based operation. There is a need to optimize the cost of providing these services without compromising on the safety of the employees. With growing number of enterprise customers, the team needed a scalable solution with disaster recovery option.

Current System overview:

The clients currently have a web application for managing the fleet management with the DB and application server remaining on-premise. The addition of users, service providers and new routes require changes to be made and validated for performance.

Newt Global successfully replaced current on-prem solution with cloud-based solution along with data migration from current systems to AWS. Currently team is also looking at Aurora as a special requirement for one of the clients needing 5 times faster output than MySQL.

Key Business/Systems requirements:

- Automated application Provisioning.
- Software Installation & Configuration
- Continuous Monitoring
- Scheduled Backup implementation
- High availability of DB and other services

Solution Considerations

Primary objective was to migrate fleet management application from On-Premises to AWS cloud to optimize TCO and utilization of resources. The on-premise legacy application was robust and must perform exactly in the same fashion in the public cloud. Some of the key technical considerations were related to use a SaaS version of the product that can scale-up or down as needed and implement changes in an agile fashion.

Choice of Migration approach

With a deep understanding of the AWS features to address the client requirements, Newt Global execute the sequence of actions all the way up to application migration.

- Launched AWS MySQL relational databases,
- AWS autoscaling with Elastic Load Balancer,

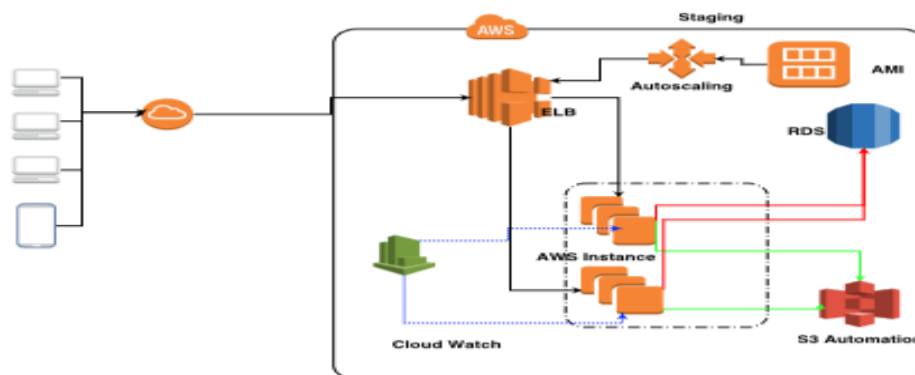
- S3 bucket, Java Spring 5, Android studio, IOS (Objective C), Angular JS for Dev, QA, Stage and Prod environments.
- Multiple availability zones(AZ) were also Created for global rollout (South Africa, Philippines, Mexico and India).

Design Considerations:

Architecture

To build the confidence of the customer, the application was hosted in a staging environment to validate all the functionalities are fully tested with the client data set. For a period of 6 weeks this environment was used to make changes as required and get them validated with live data sets for performance.

Staging Architecture



Staging environment is the replica of Production environment and will also be referred as Pre-Prod. Pre-Prod will be used for

1. User Acceptance Testing
2. Load / Performance Testing
3. Regression Testing
4. Sanity Testing

The Pre-Prod RDS will be snapshot taken out of Prod RDS on some defined interval of time.

Implementation Phase:

Since the application is a SaaS product the initial set-up and provisioning of services is quite fast and follows a templated approach. The configuration of DB and porting of data was done in 2 weeks' time. Additional customization of UI and enhancements tool 4 weeks.

Start of the project: Dec 2017 – Completion of all sites: April 2018.

Timelines

The customer had planned a progressive roll-out plan for 5 sites with 25,000 users spread across. A round robin approach was taken to complete the different phases of the implementation.

- Freezing the customization requirements took 3 weeks.
- Setting-up the staging environment took 1 week
- Developing enhancements took 2 weeks
- UAT with actual data took 2 weeks
- Configuring and launching the production environment took 1 week
- Roll-out to subsequent sites took 1 week each

Team structure

The team size for the first site was 7 people (3 developers, 1 architect, 1 QA, 1 DBA and 1 implementation engineer) all of them relevant AWS certification.

The implementation team for each of the subsequent sites has two-member teams to complete the tasks.

Use of DevOps tools

Git Hub, Jira, Confluence, Ansible and Jenkins were used in the process of migration extensively.

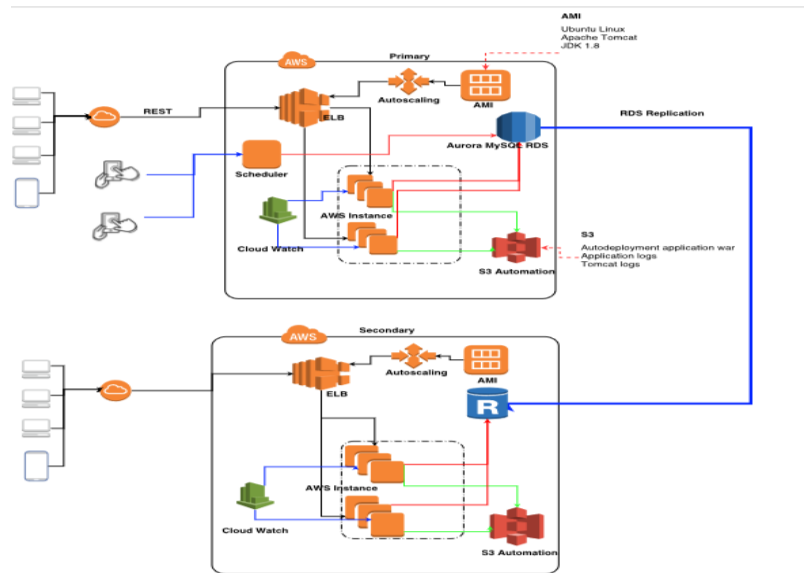
AWS services used are: EC2, S3, ELB, Cloud Watch and RDS.

Operations and Optimization

Some of the key areas for focus in the operations phase are:

- ✓ Reserved Instance usage
- ✓ Horizontal scaling based on OS resources
- ✓ AWS health-checkup to clean-up instances, AMIs, volumes, snapshots
- ✓ Schedule stop/start instances based on triggers

Deployment architecture



Some of the features tested before production cut-off were:

- **Infra Failure**
 - EC2 Failure
 - Server Failure
 - RDS Failure
 - Region Failure
- **Application Failure**
- **HOT Sites**
 - Secondary is the replica of Primary and always in active mode. The Secondary RDS will always sync with Primary RDS. If the DR happened the EFMFM team will change the DNS entry (ELB URL) It will take 5 mins to up the sites.
- **Warm Sites**
 - Secondary RDS will be in active mode. The EC2 instance of Secondary sites will be in passive mode. The AWS team will make EC2 instance active and the EFMFM team change the DNS entry (ELB URL), it will take 15 mins to up the sites.
- **Cold Sites**
 - Secondary RDS will be in read replica. The AWS team will promote the Secondary RDS and make it as Write replica. The AWS team will launch the instance from AMI. The

EFMFM team will deploy the application and start the server. The EFMFM team will change the DNS entry (ELB URL) in Go Daddy. It will take 1 hrs. time to up the sites.

Security considerations and implementation

The focus on security was present right through the process of migration and the enterprise has clear guidelines around this and the project team implemented them.

- ✓ Hardened Image provided by Enterprise AWS AMI Team
- ✓ Enabled with SSO login
- ✓ VPC design, security groups, network access control were implemented with the help of IT team at the client place.
- ✓ Application specific IAM Policy which applies in EC2, S3, RDS in all resources
- ✓ TLS Enabled from application end
- ✓ Encryption of data using the proprietary tool of the client

Business benefits of the migration:

The migration project is envisaged to give the following benefits to the client in a virtual data center environment as and when the application is on AWS.

Availability: 99.99 % availability for business continuity and user satisfaction

Autoscaling: Application able to handle 25000 users at peak time

Disaster Recovery: System can be replicated and recovered quickly into AWS without disruption or data loss

High Read/Write Capability: 3000+ IOPS at peak time, supporting 15000+ bookings simultaneously

Pro-active Monitoring and remediation

Centralized Login mechanism