KNACKFORGE

Our leading Marketing client witnessed 30% cost savings with Legacy Application Modernization and Migration
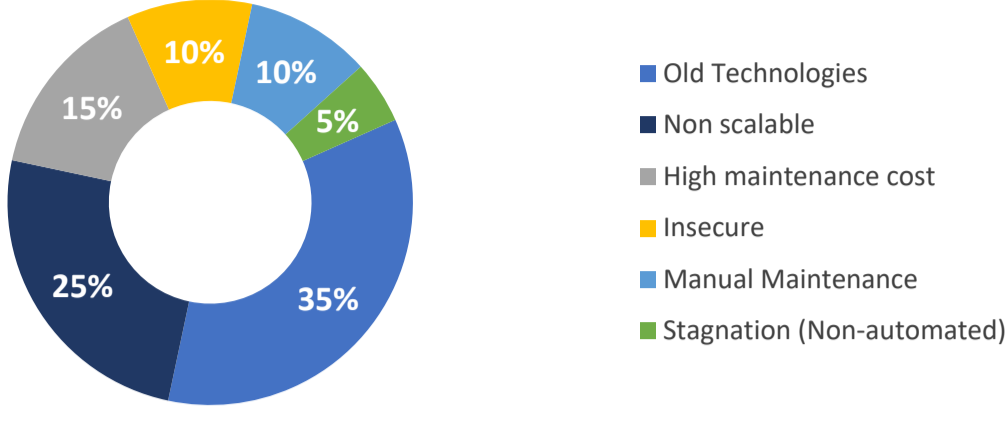
SUCCESS STORY

Legacy applications were a burden and expensive for our client, who was undergoing a digital transformation. Applications built on out-of-date platforms, using old programming languages and tools, hid unmitigated risk. These aging application systems had to be modernized to stay in line with business strategy. US-based leading Auto Repair Marketing client approached KnackForge to modernize its existing application to support newer integrations and flexibility.

CUSTOMER'S PAIN POINTS

Legacy application/infrastructure had below drawbacks such as

- 10% ■ Old Technologies
- 5% ■ Non scalable
- 35% ■ High maintenance cost
- 25% ■ Insecure
- 15% ■ Manual Maintenance
- 10% ■ Stagnation (Non-automated)

MEET THE BRAND

Our client is a leading marketing & loyalty solutions provider for the automobile industry. They offer the independent auto repair shops with the industry's leading Rewards Program. They also provide SMS Text Marketing, Email Marketing, premium Gift Card Mailers, and Predictive Reminder Mailers.

GOAL

To build a scalable application and to leverage Legacy Application Modernization, thus providing enhanced control, security, advanced storage, easy monitoring, and reduced cost.

STRATEGY

KnackForge AWS professionals conducted extensive discovery to articulate a clear assessment of the challenge. After comprehensive planning with the client, we modernized their existing application that didn't support newer integrations and flexibility. Provide the necessary support and maintenance as KnackForge modernized its existing, large and complex application.

CHALLENGES	SOLUTIONS
To build a scalable application	We used AWS autoscaling and serverless method
Modernized application	We followed serverless, AWS QuickSight reporting, code pipeline, SQS
Cost optimization	Adopted pay as per usage model, effective usage of resources
Maintenance free	Automated backups, S3 lifecycle
Easy monitoring	We used CloudWatch dashboards
Secure	Chose the best practices of AWS VPC, gradually access using IAM
Advance storage	We implemented AWS transfer family integration with Amazon S3 for storing file data

PROCESS

Data Source   Migration / ETL process   Production infrastructure
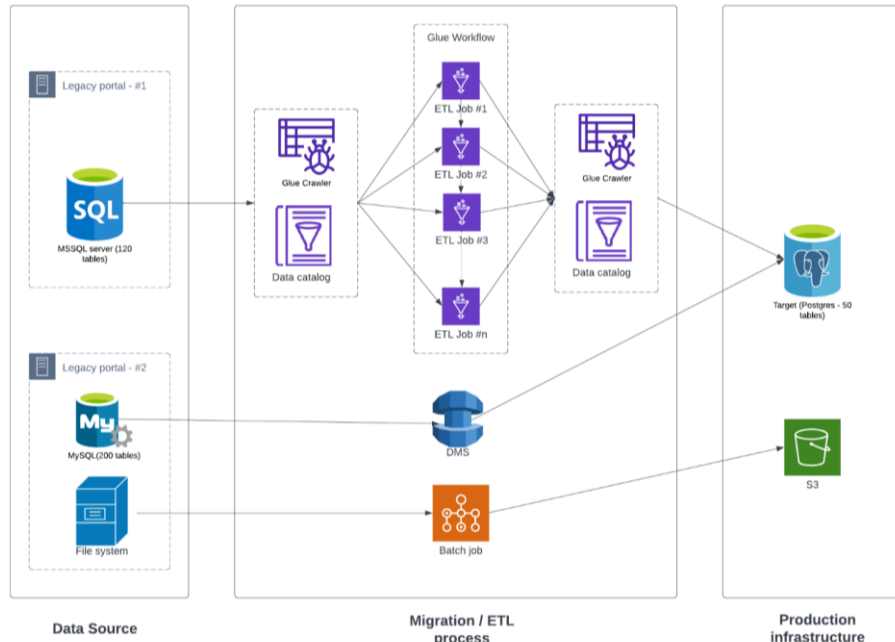
The legacy system had two portals, The Core legacy application portal and the messaging portal, with independent infrastructures. We had to migrate the two portal's data to a single application.

LEGACY PORTAL 1 - The Core Legacy Application Portal

Using AWS Glue, we converted 120 MSSQL tables of data into 40 Postgres tables. We required 40 ETL scripts for the conversion. We had to perform many transformations, joints, filtering, and deduplication to achieve the output.

### LEGACY PORTAL 2 – The Messaging Portal

Using DMS (Database migration service), we migrated MySQL tables from the source to target temp tables. Then we used the AWS Batch Jobs to transform the data within the target.

Using Batch Jobs, we transferred the files from the source (legacy file system) to the target (S3)

DATA MONITORING

- We have continuous 24/7/365 monitoring of solutions and data with the help of our Network Operation Center (NOC) team
- Web application performance, Database load, CPU Utilization - all these are closely monitored in CloudWatch and the alarms help NOC to take immediate action

PERFORMANCE: POST LAUNCH: *All In One Application – Legacy And Messaging Portal*


50%

Period
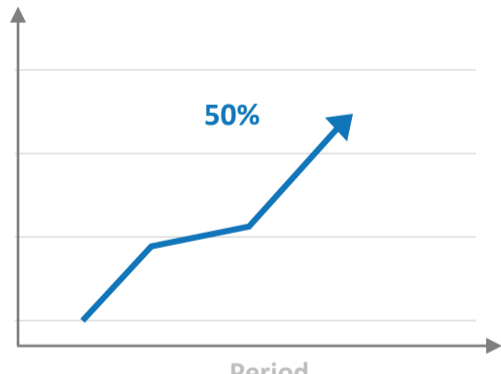
01 Performance Improvement

- Moved to Serverless technology
- Separate reader for analytics
- Effective usage of resources using Auto Scaling

30%

Period
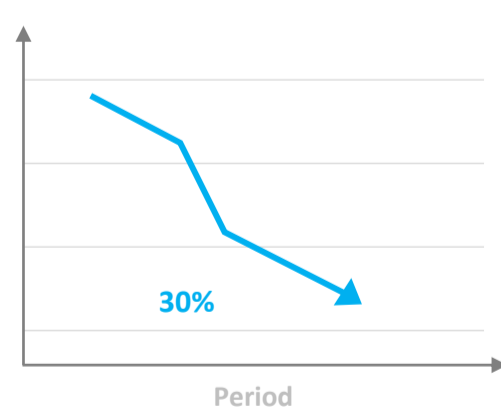
02 Cost Savings

- Requests are handled by OnDemand jobs (Serverless)
- Reserved resources for the long run
- In-house AWS cost optimization tool was used in deriving the cost reduction

Dashboard

$52,661 $47,968 $8,790

0.99999%

Period
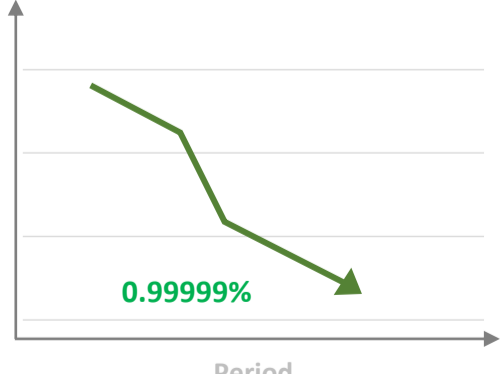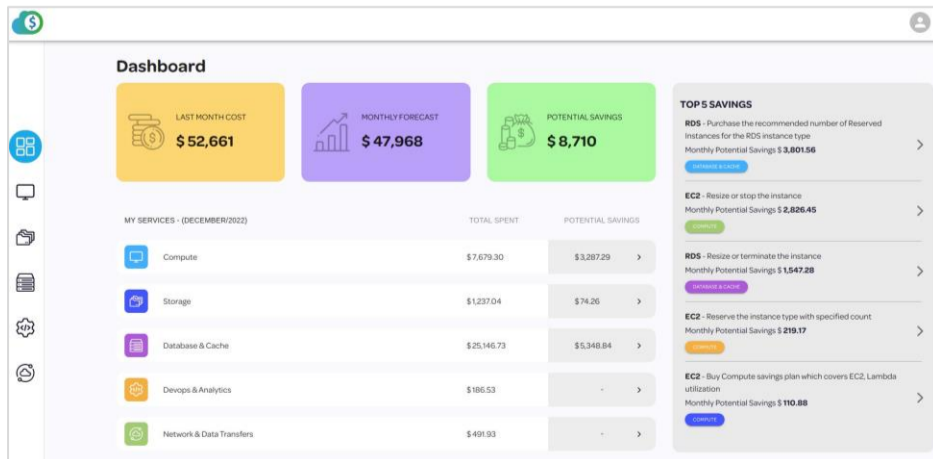
03 Data Loss From 15% to 0.99999%

- Functionalities automated and backups are enabled to prevent data loss
- Reduced 3rd party dependencies
- Automated scheduled jobs
- Modernized error handling (Using CloudWatch)

RESULT

- KnackForge's experience delivered immediate efficiencies and cost reductions: **30% cost savings were achieved.**
- Our developers modernized the legacy architecture and automated the application's monitoring system using the **Serverless method, AWS Quicksight reporting, Code pipeline, and SQS.**
- KnackForge went the extra mile by prioritizing its mitigation. We not only **improved our client's security environment,** but we also **helped them protect significant client revenue.**