ctac

Cloud Services:

Modernizing Monoliths

Cloud Services: Modernizing Monoliths

Problem:

Our client's mission is to save lives, prevent injuries, and reduce the costs of roadway crashes. To support this mission, they investigate vehicle safety issues and assess their risk to public safety. Their proprietary system serves as the data repository and analysis hub for vehicle complaints, recalls, crash test ratings, crash details, early warning reports, manufacturer communications, and foreign campaigns. This system is crucial, acting as the back-end data store, facilitating the collection and dissemination of safety information for federal investigators, auto manufacturers, and the public. It integrates with various upstream/downstream systems, serving as our client's primary portal for safety and standards in the US.

Over time, the system faced challenges due to increasing data volume and demand. It was originally built as a monolithic infrastructure on aging on-premises servers with expensive legacy software. This outdated setup no longer met the flexibility, maintainability, scalability, or performance requirements of a mission-critical system. The system's performance suffered during recall events with high traffic spikes, and scaling on the old servers was expensive and cumbersome. Our client desired an infrastructure that could scale dynamically based on demand.

Furthermore, the legacy software couldn't support modern upstream/downstream applications, making it difficult to add new features and functionality. Recognizing these challenges, our client sought CTAC's expertise to modernize the application software and migrate legacy workflows to a cloud-based, modular system, ensuring improved performance to meet both current and future needs.

Solution:

The system urgently needed modernization, encompassing both software and hardware aspects. A 'Lift and Shift' migration from on-premises to the cloud wouldn't suffice due to years of technical debt. Instead, our team conducted a comprehensive migration to AWS, replacing legacy middleware and appliances with AWS-managed services. Crucially, the system played a central role for various entities, necessitating uninterrupted real-time information workflows during the migration.

Our partnership with AWS involved close collaboration with their engineers, ensuring alignment with AWS best practices. We transitioned from on-premises physical networks to Virtual Private Networks (VPCs) and subnets. The VPC was designed with public/private subnet separation, service endpoints, and NAT Gateways. Amazon WAF, ACLs, security groups, and IAM enforced security policies. Microservices on Lambda were the target for most workloads, while autoscaling EC2 clusters were used for resource-intensive tasks like zip file processing and threat scanning. Serverless scripts, leveraging Lambda Layers and versioning, ensured runtime flexibility and rollbacks. We established four distinct environments (Development, Staging, Q&A, and Production) as Infrastructure as Code (IaC) using CloudFormation scripts and AWS Serverless Application Model (SAM).

Much of the legacy infrastructure consisted of proprietary middleware, applications, and costly software licenses. The modernization and migration of the system included the following changes:

- The migration of data out of legacy Oracle instances into AWS Aurora PostgreSQL.
- Replacing legacy enterprise Java applications and app servers with modern serverless nodeJS equivalents
- Replacing proprietary middleware and appliances with services including SQS (queued messages), SNS (notification of events), AWS S3, and API Gateway. API Gateways were defined using Swagger.
- Replacing costly Microsoft Windows server instances with more flexible RedHat Enterprise Linux (RHEL)
- Replaced legacy search tool Lucene with AWS OpenSearch Service
- Replace dedicated web servers with S3 and CloudFront

Additionally, our team used AWS Command Line Interface (CLI) and Software Developer Kits (SDKs) for object migration and syncing, while also configuring a DevOps CI/CD pipeline managed within Jenkins. This pipeline is integrated with AWS CodeCommit for version control. AWS CloudWatch provided telemetry metrics and error reporting, visualized through CloudWatch dashboards. The resulting modernized system ensures flexibility, performance, and maintainability for our client's mission of public safety.

Outcome:

The result of this work provided our client with a modernized, modular cloud-native solution that has the scalability and performance to meet the growing needs of their user base for years to come. The system is now built with the scalability to meet the system's reliably unreliable traffic. Its monolithic infrastructure has been replaced with serverless microservices, providing developers with the ability to quickly update workflows and add new features as needed.

Previously, the architecture required queued, multi-day batches to ingest and disseminate information. Now, the system's integration and design allow for real-time, asynchronous data flow between upstream/downstream applications. Dozens of expensive, legacy servers have been replaced by a handful of on-demand cloud computing instances. The migration and modernization of the system have provided our client and its customers with the capability for continued enhancement and growth for years to come.