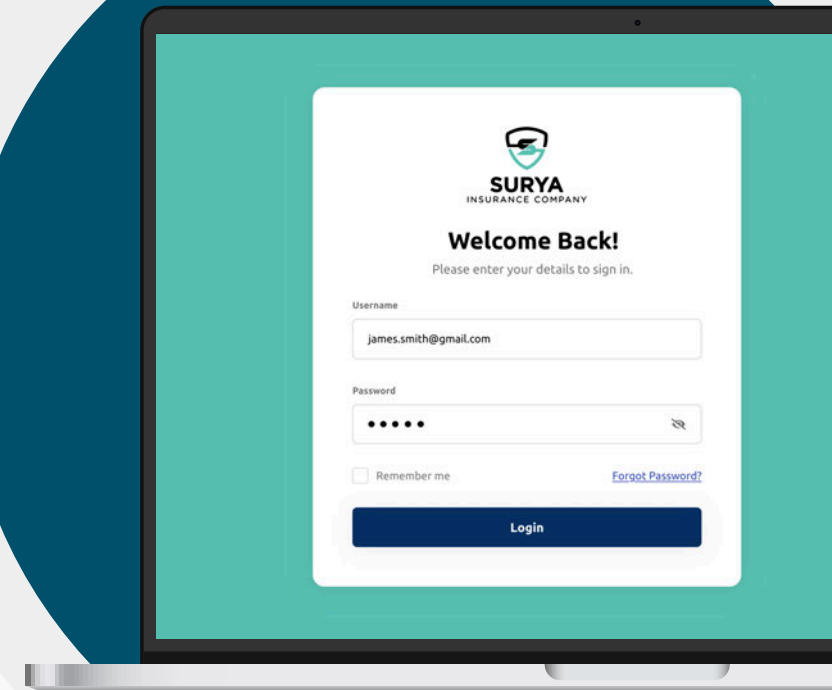


Transforming Application Development with **Surya Insurance**

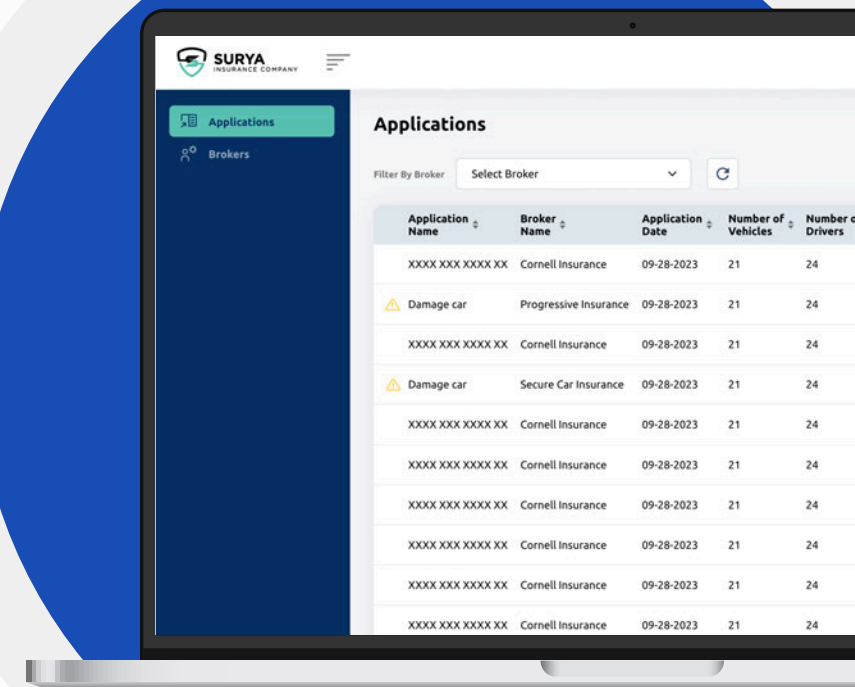
Client Overview

Surya Insurance Company (SIC) is a prominent regional insurance provider offering a comprehensive portfolio of life, health, and general insurance products. As part of its long-term digital transformation roadmap, SIC aims to modernize its core platforms and improve customer experience by streamlining internal operations and embracing cloud-native technologies. The company operates across multiple geographies and has a strong commitment to improving operational agility, reducing go-to-market timelines, and ensuring robust security and compliance.



SIC was struggling with legacy application development practices that hindered their ability to deliver new digital services efficiently. Their form-processing web application, which was critical for capturing policyholder information and issuing policies, was plagued with several issues:

- To meet evolving business and customer demands, SIC needed to adopt a modern DevOps approach that enabled automation, agility, and continuous delivery of reliable and secure applications.



Proposed Solution

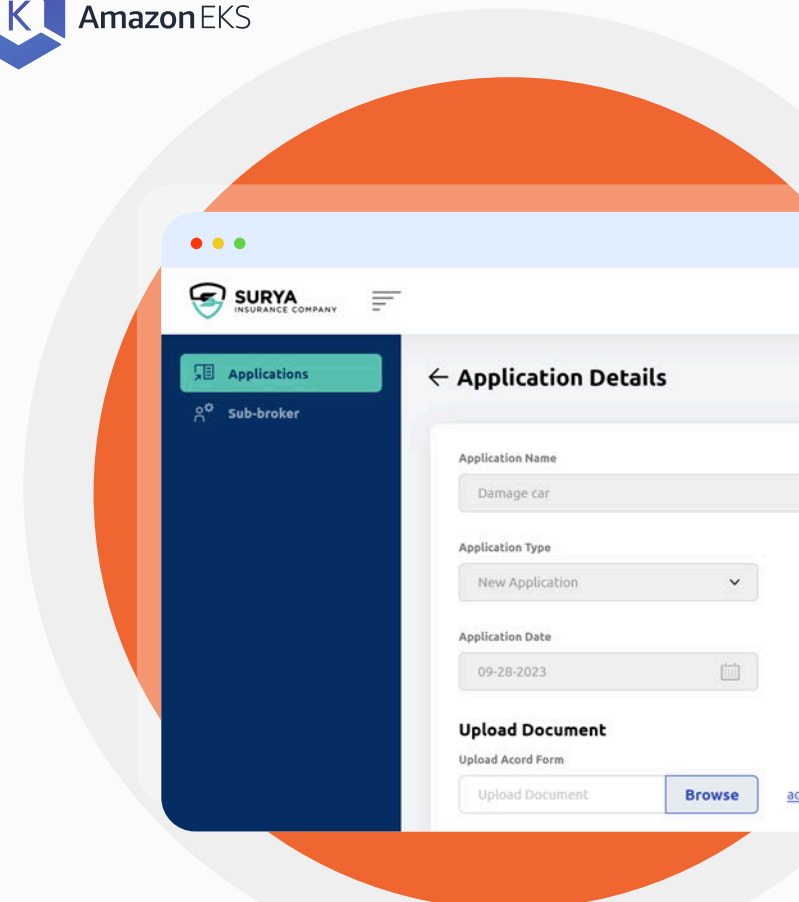
To address these challenges, we implemented a **DevOps-first solution** leveraging AWS-native services and modern toolchains. The goal was to deliver an end-to-end automated pipeline that empowered development and operations teams to work in sync, reduce manual interventions and improve time-to-value.

CI/CD Pipeline Implementation

» **Tools Used:**  **Jenkins**    AmazonEKS

- **Source Code Management:** All source code and environment configurations were version-controlled using GitHub, improving collaboration and traceability.
- **Continuous Integration:** Jenkins was used to automate code builds, unit testing, and integration testing.
- **Code Quality Checks:** SonarQube was integrated into the Jenkins pipeline to perform code analysis and enforce quality gates.
- **Continuous Deployment:** Spinnaker was used to implement canary and rolling deployments on Amazon EKS, ensuring safe rollouts with zero downtime.

This robust CI/CD pipeline enabled frequent, secure, and reliable application releases with minimal manual effort.

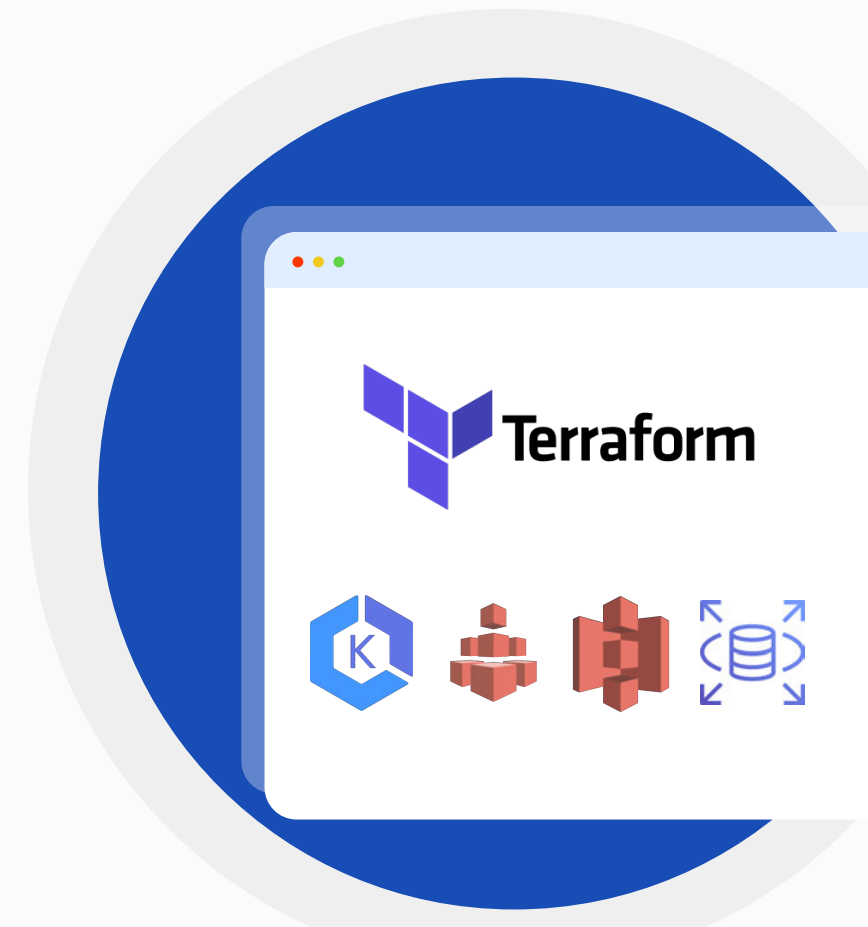


Infrastructure as Code (IaC)

We adopted **Terraform** to manage the entire infrastructure lifecycle as code, allowing repeatable and auditable provisioning of the following key AWS resources across all environments:

- **Amazon EKS** for container orchestration with high availability and scalability.
- **Amazon EFS** for durable, shared storage of form documents.
- **Amazon RDS** for storing application metadata and transactional data.
- **Amazon S3** for storing large form inputs and processed files.
- **AWS KMS** for encryption and key management across services.
- **Secure IAM policies and VPC configurations** to isolate workloads and protect sensitive data.

This modular IaC approach accelerated environment provisioning and made it easier to enforce best practices for security and compliance.



Security and Compliance

Security was embedded across the DevOps pipeline using "**shift-left**" **security practices** to catch vulnerabilities early and automate enforcement:

- **AWS Inspector** continuously scanned container images and EC2 instances for known CVEs.
- **Terraform-based IAM policies** enforced the principle of least privilege across all services.
- **AWS WAF and Shield** were used to secure the public-facing endpoints from OWASP Top 10 threats.
- **Audit logs were** captured through **AWS CloudTrail**, ensuring accountability and aiding incident response.

This proactive security model significantly reduced the risk of exposure and ensured continuous compliance with internal and external policies.



Monitoring and Logging

A comprehensive monitoring stack was implemented to provide deep observability into application and infrastructure behaviour:

- **Amazon CloudWatch** provided metrics, dashboards, and alarms for AWS services.
- **Prometheus and Grafana** were deployed on EKS to collect custom application metrics and visualize them in real-time.
- **Loki** was integrated with Grafana to ingest and query logs from all container workloads.
- **CloudTrail** was used to track changes and access patterns for auditing and troubleshooting.

This observability setup enabled real-time incident detection, faster root cause analysis, and proactive performance tuning.

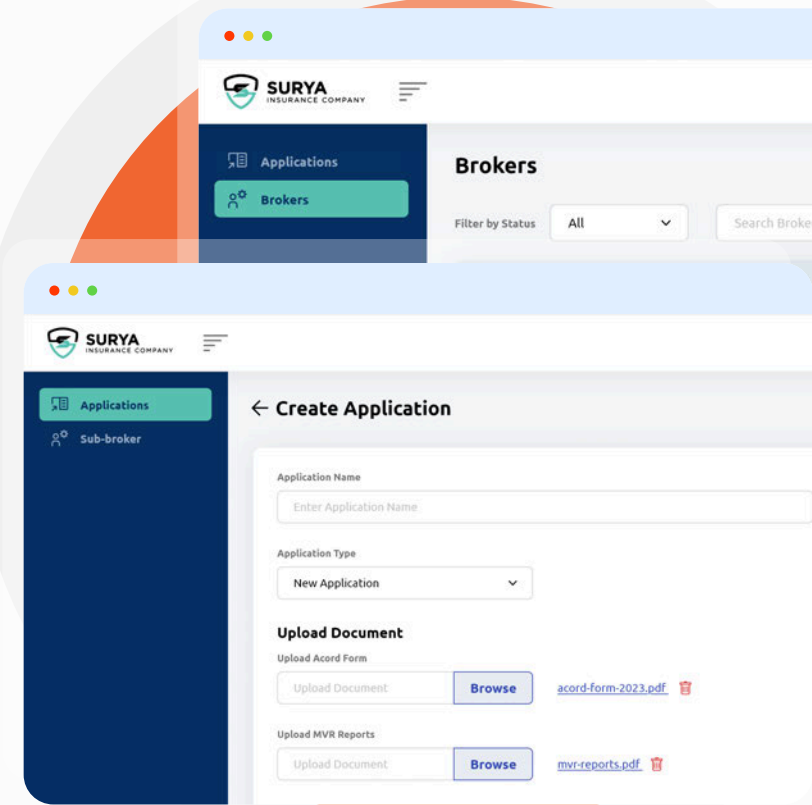


Cost Optimization

We implemented multiple cost-saving strategies aligned with AWS Well-Architected best practices:

- **Auto Scaling** ensured that compute resources matched workload demands, minimizing overprovisioning.
- **Spot Instances** were used in development and testing environments to cut costs by over 60%.
- **Resource tagging and lifecycle policies** helped track ownership and clean up unused assets.
- **AWS Budgets and Cost Explorer** were configured for ongoing financial governance.

These optimizations led to substantial cost savings while meeting performance and availability requirements.



» Operational Efficiency:

- ### » Deployment Reliability:

- » Cost Savings:**

- [illegible]

The DevOps transformation led to tangible business outcomes:

- This engagement positioned Surya Insurance as a forward-looking insurer capable of competing in the digital-first era.



Lessons Learned

- » **IaC is a game changer:** Terraform accelerated the infrastructure lifecycle and simplified multi-environment management.
- » **Security must be proactive:** Embedding security early in the CI/CD pipeline prevented vulnerabilities from reaching production.
- » **Observability is critical:** Centralized monitoring helped identify performance bottlenecks and application errors before users were impacted.
- » **Containerization drives agility:** EKS enabled consistent deployments and simplified rollback mechanisms.

These lessons shaped future DevOps adoption and tooling standardization across SIC's digital initiatives.

A large, stylized orange wavy shape at the bottom of the slide, composed of several concentric, overlapping curved lines.