



Insurance Company DevOps Transformation

Professional Technology Consulting

The Origami Corporation

18.09.2025

Table of Contents

1

From Fortnightly Releases to Daily Flow: Modernizing an Insurance Portfolio with GitOps and IaC

1

1.1

Executive Summary

1

1.2

The Challenge

1

1.3

Our Approach

2

1.4

Implementation

2

1.5

Results & Impact

4

1.6

Technologies & Services Used

4

1.7

Conclusion

5

About The Origami Corporation

1 From Fortnightly Releases to Daily Flow: Modernizing an Insurance Portfolio with GitOps and IaC

1.1 Executive Summary

An established insurance provider managing 50+ legacy applications faced slow, error-prone releases and high operational overhead. Deployments occurred every two weeks, frequently resulting in production issues and rollbacks. Manual server and application configuration consumed 30 hours per week, limiting the engineering team’s ability to innovate.

We delivered a comprehensive transformation anchored in DevOps-as-a-Service, CI/CD implementation, infrastructure-as-code (IaC), DevSecOps training, and a targeted cloud migration. By introducing GitOps workflows, automated testing pipelines, and standardized Terraform-based infrastructure, the company’s 40-person development team now deploys 15 times per day with 90% fewer rollbacks. Manual configuration work fell from 30 hours to 2 hours per week, freeing engineers to focus on product value rather than maintenance.

1.2 The Challenge

The insurer’s application estate—comprised of monolithic services, batch processors, and customer-facing portals—had grown over a decade without standardized build, test, or deployment practices. Key pain points included:

- Release drag: Two-week deployment cycles, with change windows that constrained delivery and increased risk.
- Production instability: Frequent misconfigurations and drift between environments led to outages and urgent hotfixes.
- Manual toil: Infrastructure and application configuration consumed 30 hours weekly across teams, delaying work

on new features. - Compliance friction: Audit trails for changes were fragmented across systems, complicating regulatory reporting typical in financial services. - Skills gap: Teams needed guidance to adopt modern DevOps, security, and cloud practices safely and consistently.

The objective was to accelerate delivery without compromising reliability or compliance—standardizing tooling and processes while modernizing a complex, legacy-heavy portfolio.

1.3 Our Approach

We designed a pragmatic, phased program leveraging five service pillars:

- **DevOps-as-a-Service:** A dedicated cross-functional team embedded with client squads to define standards, build the platform, and coach developers. We aligned delivery practices to DORA principles and introduced platform guardrails.
- **CI/CD Implementation:** Unified pipelines standardized build, test, and release logic across repositories. Automated gates enforced quality, security, and compliance checks.
- **Infrastructure-as-Code:** Terraform codified cloud networking, compute, IAM, and application dependencies. Policy-as-code enforced governance from day one.
- **DevSecOps Training:** Hands-on enablement in secure coding, threat modeling, pipeline triage, and secrets management to shift security left and reduce rework.
- **Cloud Migration:** A targeted move to a modern cloud foundation, balancing rehost and replatform strategies for legacy workloads to reduce operational burden and enable elastic scaling.

Central to the transformation was a GitOps operating model—treating everything (infrastructure, application manifests, and configurations) as code, with pull-request (PR) workflows, automated testing, and continuous reconciliation.

1.4 Implementation

We executed in iterative waves to minimize risk and build confidence:

1. Assessment and Target Architecture

- Mapped 50+ applications by criticality, dependencies, and modernization feasibility.
- Defined a reference architecture for a major public cloud, including network segmentation, identity federation, and baseline observability.

- Selected a container-first approach for services ready to replatform; identified candidates for lift-and-shift as interim steps.

2. Terraform-Driven Cloud Foundation

- Built a “landing zone” using Terraform modules: VPC/subnets, security groups, IAM roles/policies, encryption, logging, and CI/CD runners.
- Implemented remote state with locking, versioning, and drift detection; enforced changes via PRs and Terraform plans.
- Adopted policy-as-code (OPA/Conftest) to enforce tagging, encryption-at-rest, and least-privilege IAM.

3. CI/CD Standardization

- Introduced standardized pipelines (e.g., GitHub Actions/GitLab CI) with reusable templates:
 - Build and package with reproducible Docker images
 - Unit, integration, and contract tests
 - Static analysis (code quality and SAST), dependency scanning, and container image scanning
 - SBOM generation and artifact signing
 - Infrastructure plan/apply as controlled stages
- Enabled parallelization and caching to reduce pipeline runtime and deliver fast feedback loops.

4. GitOps Adoption

- Externalized environment configuration using Helm and Kustomize overlays; declared the “desired state” per environment in Git.
- Deployed Argo CD for continuous reconciliation, ensuring clusters matched Git at all times; drift triggered automated alerts.
- Enforced approvals via PRs with branch protection, mandatory reviews, and required status checks before promotion across dev, test, and prod.

5. Application Modernization and Migration

- Containerized priority services; for complex monoliths, extracted high-change components first while keeping the rest stable.
- Staged migration in waves: non-critical apps first, followed by customer-facing and core policy systems.
- Introduced blue/green and canary strategies to lower risk; feature flags separated deployment from release for controlled rollouts.

6. Security by Design and Training

- Integrated secrets management (e.g., Vault/SM) and rotated credentials tied to workload identity.
- Embedded SAST, DAST, and license compliance scanning into pipelines; security failures blocked merges automatically.

- Conducted DevSecOps workshops, threat modeling sessions, and “game days” to build muscle memory for incident response.

7. Observability and Operations

- Standardized logging, metrics, and traces with centralized dashboards and alerts.
- Implemented golden signals (latency, traffic, errors, saturation) and SLOs to guide reliability decisions.
- Documented runbooks, playbooks, and on-call rotations for production readiness.

Throughout, we partnered closely with product owners to prioritize migration candidates and with compliance teams to ensure audit-ready evidence: every change traceable to a ticket, a PR, and an automated pipeline outcome.

1.5 Results & Impact

The transformation delivered measurable and sustained improvements:

- Deployment frequency: From scheduled releases every two weeks to on-demand delivery—15 production deployments per day across teams.
- Quality and stability: 90% fewer rollbacks due to robust automated testing, progressive delivery, and GitOps-driven consistency across environments.
- Operational efficiency: Manual configuration work dropped from 30 hours to 2 hours per week, thanks to Terraform automation and standardized pipelines.
- Developer productivity: Engineers shifted from firefighting to feature development; platform self-service reduced wait times and handoffs.
- Risk reduction and compliance: Immutable audit trails for infrastructure and application changes simplified regulatory reporting and improved change governance.
- Customer impact: Faster iteration cycles enabled rapid delivery of product enhancements and defect fixes, improving user satisfaction and competitive responsiveness.

These outcomes reflect not only tooling improvements but also new ways of working—codified standards, PR-centric collaboration, and a culture of continuous improvement reinforced by training and coaching.

1.6 Technologies & Services Used

- Services:
 - DevOps-as-a-Service
 - CI/CD Implementation

- Infrastructure-as-Code
- DevSecOps Training
- Cloud Migration
- Tooling and Platforms:
 - Terraform for IaC with remote state and policy-as-code (OPA/Conftest)
 - Git-based workflows with protected branches and PR approvals
 - CI/CD runners (e.g., GitHub Actions/GitLab CI) with reusable pipeline templates
 - GitOps controller (Argo CD) with Helm and Kustomize for environment overlays
 - Containerization with Docker; orchestrated deployments on managed Kubernetes
 - Security scanning (SAST, dependency, and container image scanning), SBOM, and artifact signing
 - Secrets management integrated with workload identity
 - Centralized logging, metrics, and tracing for observability

1.7 Conclusion

By aligning engineering practices around GitOps, IaC, and automated CI/CD, the insurer transformed a complex legacy landscape into a modern, high-velocity delivery engine. The shift from two-week release cycles to 15 daily deployments—while achieving 90% fewer rollbacks and slashing manual configuration from 30 hours to 2 hours per week—demonstrates the power of pairing strong platform foundations with enablement and governance.

Our DevOps-as-a-Service model ensured sustained adoption and continuous improvement, while DevSecOps training embedded security into daily workflows. The cloud migration and platform standardization created a scalable, compliant foundation for future growth. The organization now ships faster with greater confidence—turning operational excellence into a strategic advantage.

About The Origami Corporation

The Origami Corporation is a leading technology consulting firm specializing in cloud migration, DevSecOps, cybersecurity, and digital transformation. We help organizations navigate complex technological challenges and achieve their business objectives through innovative, secure, and scalable solutions.

Connect With Us

Follow Us Online

LinkedIn: [The Origami Corporation](#)

Facebook: [@theorigamicorporation](#)

X (Twitter): [@OrigamiGlobal](#)

Telegram: [@theorigamicorporation](#)

Contact Information

Europe & Americas

- **Email:** office@theorigamicorporation.com
- **Phone:** +359 878 686 627

Asia-Pacific, Middle East & Africa (APMEA)

- **Email:** office.apmea@theorigamicorporation.com
- **Phone:** +41 77 924 2245

Ready to Get Started?

SCHEDULE A MEETING

Ready to transform your technology infrastructure? Contact us today to discuss how we can help your organization achieve similar results.