

From Zero to Series C:

# How GlobalDots Became Justt's Infrastructure Backbone for 4+ Years

**Company:**  
**justt**

**Environments:**  
Fintech and payments industry

**Challenges:**

- Complex, fragmented infrastructure
- Manual, untracked deployments
- No consistent testing environments
- Risky, outdated messaging system

**Results:**

- Deployment time cut from 120 → 20 min
- 4,000+ engineering hours saved/year
- \$100K annual infrastructure savings

Throughout, Justt's product engineers stayed focused on what they do best: building the AI chargeback engine.

## About Justt

Justt is an AI-powered chargeback mitigation platform built for financial institutions and merchants. Its technology automates the dispute process across payment networks, recovering revenue that would otherwise be written off.

## Background

### Infrastructure Became a Go-to-Market Bottleneck

By 2021, the product was gaining traction. The engineering team focused on improving the core system, while the underlying cloud environment had been built in layers. One account at a time. One service at a time. One workaround at a time. That approach worked early on. It stopped working once the system had to scale.

**The platform had grown faster than the team's capacity to manage it cleanly. Something had to change. Infrastructure wasn't a technical debt problem. It was a go-to-market bottleneck.**

## The Challenge

### The Platform Became Too Complex to Manage

As Justt scaled, the cracks in the platform became harder to ignore. The **AWS environment** had grown to eleven accounts across multiple teams and use cases. Five separate **Kubernetes clusters** handled production workloads, data pipelines, and internal services. Each had different operational requirements.

**This wasn't just complexity. It was fragmentation.**

Deployments required manual coordination. There was no consistent way to validate changes before they reached shared environments. Infrastructure changes were ad-hoc and untracked, which led to drift between environments that only surfaced during incidents. At the center of the event pipeline sat a **deprecated message broker**. It was deeply embedded and difficult to replace, yet increasingly risky to keep. The result was predictable. Every change carried more risk. Every release required more coordination. The system was slowing the team down. Hiring a dedicated DevOps engineer was one option. But it would take time, and it would concentrate too much responsibility in a single role. The team needed a different path.

## Solution & Strategy

### Hire DevOps or Bring in Experience

Justt chose not to hire in-house. They brought in **GlobalDots**. Hiring was the obvious path. It was also a slow and fragile one. Months to hire. High dependency on a single person. Limited ability to handle complex architectural decisions without additional support.

**The alternative was to bring in a team that had already solved this class of problem. The principle was simple: product engineers should not be blocked by infrastructure complexity. Every decision followed that. Build a platform that is stable, predictable, and requires minimal ongoing attention.**

GlobalDots embedded a senior engineer inside the team from day one, backed by a broader engineering group for deeper architectural decisions. This wasn't a single hire. It was an execution with built-in support.



## Technical Implementation

### Standardizing and Modernizing the Platform

GlobalDots approached the engagement in two phases, each with a clear mandate.

#### Phase 1

#### Platform Stabilization

Infrastructure changes were moved to **Terraform**, with **Atlantis** enforcing review and approval before every change. What had been ad hoc became controlled and traceable. Deployments shifted to a **GitOps** model using **ArgoCD** as the single source of truth across all clusters.

**Releases became automated, with a consistent path from staging to production.**

Two changes directly improved the developer workflow:

- 1 Feature-branch environments** allowed developers to spin up isolated environments for every pull request, connected to controlled staging data and enabled testing without team coordination.
- 2 ConfigHub** allowed each service to define its own resources, permissions, and scaling as code. Infrastructure was provisioned automatically from those definitions.

Together, these changes moved the platform toward a self-service model, reducing reliance on centralized DevOps. Also, Node autoscaling was migrated to **Karpenter**, which provisions capacity in real time based on demand, reducing scaling delays and unused capacity.

Phase 2

### Platform Modernization

The focus shifted from stabilizing the platform to removing long-term risks and improving how it scales.

- ✓ The deprecated RabbitMQ broker was replaced with managed **AWS services**. This removed a critical dependency that had been flagged as a growing risk, without disrupting the existing event-driven architecture.
- ✓ Deployments moved to **ArgoRollouts**, enabling gradual, controlled releases instead of all-or-nothing deployments.
- ✓ Autoscaling was redesigned using **KEDA**, allowing services to scale based on real workload signals such as queue depth and stream lag, rather than static CPU thresholds.
- ✓ Additional improvements were made across the platform, including cluster upgrades, networking optimizations, and extended access control.

When the GlobalDots engineer who built this foundation stepped away in mid-2025, another engineer took over. The transition was seamless. The platform kept running. The roadmap continued. The product team did not feel the change.

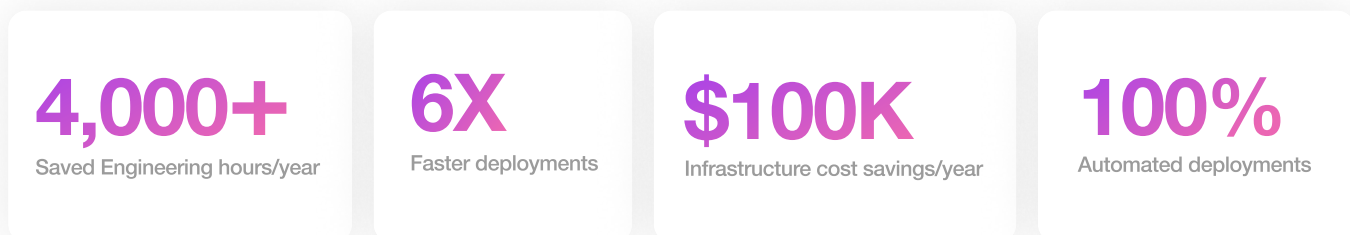
**The knowledge and decision-making did not sit with one person. It was built into the way the system was designed and maintained.**

## Results

### Scaling Without Infrastructure Bottlenecks

Justt tripled its revenue in 2023 and doubled it again in 2024, reaching Series C. The infrastructure scaled with the business at every stage, without becoming a constraint on growth.

#### Operational impact included:




More importantly, the platform became operationally stable. Over four years, the product engineering team maintained uninterrupted focus on the chargeback AI that drives the company's value. Developers could test changes in isolated environments without coordination, and high-risk components like the deprecated message broker were replaced without production incidents.

**Justt scaled from an early-stage startup to a growing fintech company without hiring a single in-house DevOps engineer.**

**Infrastructure was never the bottleneck.**

# Key Takeaways

Hire DevOps or Bring in Experience



**Infrastructure debt does not stay technical.**

At scale, it becomes a growth constraint.

**Complex platforms cannot rely on a single owner.**

Knowledge has to live in the system, not in one person.

**Well-designed infrastructure removes itself as a dependency.**

Product teams move without waiting for it.

## About GlobalDots

GlobalDots is a cloud innovation partner that connects businesses with best-of-breed technologies in web performance, cloud security, cost optimization, and DevOps. With 20+ years of expertise and a multi-vendor approach, GlobalDots serves as an extension of its customers' engineering teams.