

● CLOUD ARCHITECTURE · AWS

Delivering High–Availability Business Applications Through a Resilient AWS Architecture

99.99%

Platform availability, up from 99.5%

<60s

Database failover with zero data loss

30–40%

Lower infrastructure cost

<15min

Deploy time, down from 2–3 hours



Executive Summary

THE OPPORTUNITY

The customer needed a **future–ready, cloud–native foundation** to support a rapidly evolving digital platform. As usage patterns and business expectations grew, the underlying infrastructure had to scale effortlessly, stay highly available, enforce strong security standards, and offer clear, real–time visibility into system behavior — all while keeping consistent performance and operational control across variable workloads.

This case study explores how **Crest Data designed and implemented a modern application and observability architecture on AWS** to meet those objectives. It outlines the strategic decisions, technical considerations, and outcomes that enabled the customer to operate a resilient, scalable platform while gaining the insights needed to continuously optimize and grow their digital ecosystem.



About the Customer

The customer is a **premium nightlife marketing and live–events organization** that produces high–profile special events and recurring nightlife experiences across major U.S. metropolitan markets, run from centralized operations with regional teams in key cities.

Known for an upscale, trend–driven audience, they are a trusted source for event and venue information within elite nightlife communities. As their digital presence grew, so did the need for a platform that could scale with their ambitions — something their legacy hosting could not support.

CUSTOMER PROFILE

**INDUSTRY****Nightlife marketing & live events****FOOTPRINT****Major U.S. metro markets****CORE WORKLOAD****Containerized web platform — frontend, APIs, inventory & admin****CRITICAL NEED****Elastic scale with high availability**



The Customer Challenge

LEGACY HOSTING LIMITS

The customer's existing products were hosted on **LiquidWeb**, which had served simpler workloads adequately. But as they planned a new suite of containerized applications — a frontend interface, backend APIs, inventory management, and an administrative panel — the limits of that environment became clear. Rather than expand on LiquidWeb, Crest Data recommended building the new venture on AWS.



Limited scalability. No elastic scaling to absorb unpredictable traffic spikes during major events and promotional campaigns.



Fragmented observability. Minimal visibility into application performance made proactive issue detection nearly impossible.



Manual deployments. Releases were error-prone and slow, with no integrated CI/CD pipeline for consistent, automated delivery.



Inadequate security controls. Managing credentials, API tokens, and access policies was cumbersome, with limited auditability.



Configuration drift. Without infrastructure-as-code, changes to critical infrastructure were difficult to track, audit, or roll back — leaving no centralized secrets management and no reliable change history.



Why AWS Over LiquidWeb

CAPABILITY ASSESSMENT

Crest Data evaluated the customer's current and future requirements against LiquidWeb's capabilities. AWS's **managed-service ecosystem, native automation, and enterprise-grade security** made it the clear foundation for the new venture.

CAPABILITY	LIQUIDWEB	AWS
Elastic scalability	Limited vertical scaling	✓ Auto-scaling , on-demand resources
Continuous deployment	Manual processes	✓ GitLab CI/CD + Terraform
Observability	Basic monitoring	✓ Full-stack: Prometheus, Grafana, Loki
Resource management	Static provisioning	✓ Dynamic, IaC-managed
Auditability	Limited logging	✓ CloudTrail, version-controlled IaC
Authentication & access	Basic controls	✓ IAM, Secrets Manager, least-privilege

The solution combines **Amazon EC2, Aurora PostgreSQL, CloudFront, AWS Lambda, S3, and Secrets Manager** into a secure, resilient, and flexible foundation. A containerized stack on Docker Compose enables modular deployment, with open-source observability — Grafana, Prometheus, Loki, and Fluent Bit — delivering end-to-end monitoring across metrics, logs, and system health.

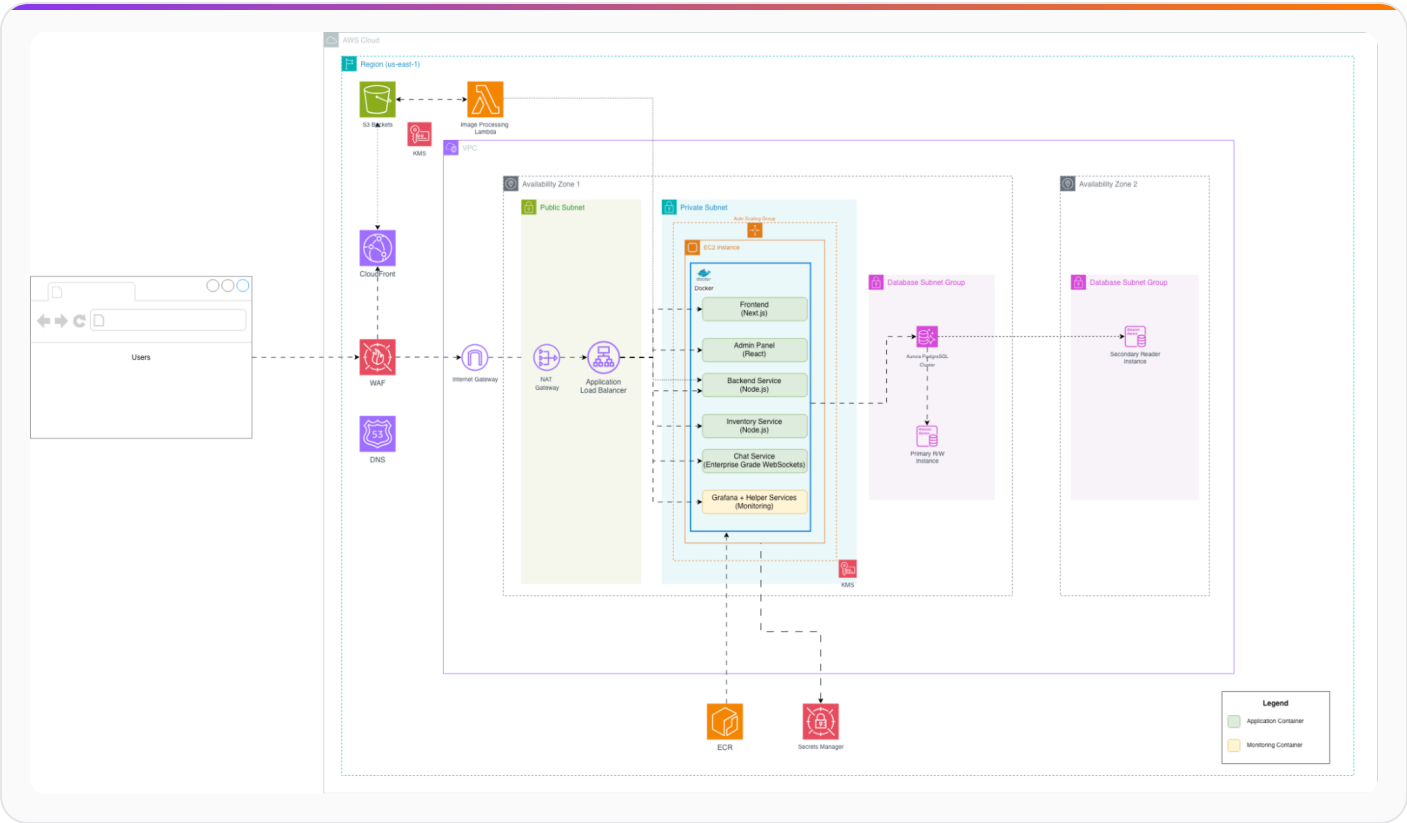


FIGURE 1

High-availability AWS architecture across multiple Availability Zones — edge delivery, containerized application tier, Multi-AZ database, and centralized security & secrets.

ARCHITECTURE AT A GLANCE

- Amazon EC2

Aurora PostgreSQL (Multi-AZ)

Amazon CloudFront

AWS Lambda

Amazon S3
- AWS Secrets Manager

Amazon ECR

IAM

Docker Compose

Nginx

Grafana

Prometheus
- Loki

Fluent Bit



Environment & Networking

- VPC with public & private subnets across **multiple Availability Zones**
- EC2 in a public subnet runs containers via Docker Compose; Elastic IP for static access
- CloudFront serves S3 static assets globally; Nginx handles intelligent routing



Compute & Application

- Containerized stack — **frontend (Next.js)**, admin (React), backend & inventory (Node.js), chat (WebSockets)
- Consistent environments via Docker Compose
- Lambda auto-compresses uploaded images, optimizing storage & delivery



Database & Security

- Aurora PostgreSQL Multi-AZ** with automatic failover
- ECR with image vulnerability scanning; security groups enforce granular access
- IAM least-privilege & Secrets Manager eliminate hardcoded secrets; HTTPS via CloudFront

Case Study · Cloud Architecture · AWS

www.crestdata.ai

P. 03 / 05



The most significant gain over the legacy environment was **comprehensive automation and observability** — transforming how the customer ships, monitors, and secures their platform.



Infrastructure Automation

- Entire infrastructure defined as **code with Terraform**
- All configs stored in Git for complete version history & auditability
- GitLab pipelines deploy Terraform stacks on merge — true continuous deployment
- Every change tracked, reviewed, and auditable



Monitoring & Logging

- Prometheus** collects time-series metrics from all containers
- Loki aggregates logs with Fluent Bit as the collection agent
- Real-time alerting enables proactive detection before user impact



Centralized Observability

- Grafana** unifies metrics and logs from every container
- One pane of glass across application performance, infrastructure health, and business metrics



DevSecOps Best Practices

- Version-controlled IaC ensures reproducibility & audit trails
- Rolling upgrades** minimize downtime during deployments
- Secrets Manager removes credential exposure; IAM enforces auth on every change

WHY AWS — THE DECIDING FACTORS

- ✓ **Managed services** offload patching, scaling & maintenance
- ✓ Enterprise-grade observability stack out of the box
- ✓ CloudTrail + versioned IaC give a complete audit trail
- ✓ Native GitLab CI/CD + Terraform for repeatable deploys
- ✓ On-demand provisioning ends costly over-provisioning
- ✓ IAM + Secrets Manager enforce least-privilege access



Outcomes & Success Metrics

AVAILABILITY · SECURITY · EFFICIENCY



High Availability

- ◆ RDS failover confirmed in **under 60 seconds**, with zero data loss via synchronous replication
- ◆ Architecture delivers **99.99% availability** — up from the previous 99.5% SLA
- ◆ Aurora PostgreSQL Multi-AZ provides automatic failover



Security

- ◆ Least-privilege IAM cut unauthorized-access risk by **~40%** vs. the prior setup
- ◆ Static assets delivered securely through CloudFront + S3 + Lambda
- ◆ Full auditability of infrastructure changes via CloudTrail and Git history



Operational Efficiency

- ◆ Automated CI/CD eliminated **~70% of repetitive deployment tasks**, saving ~40 hours/month
- ◆ Deployment time fell from 2–3 hours to **under 15 minutes**
- ◆ On-demand resources removed over-provisioning waste



Observability

- ◆ **100%** of metrics & logs captured via Prometheus + Loki + Fluent Bit
- ◆ Mean Time to Detection reduced by **60%** through real-time alerting
- ◆ Automated scaling & patching cut unplanned downtime incidents **~25% YoY**

70%

Repetitive deploy tasks eliminated

60%

Faster mean time to detection

100%

Metrics & logs captured

~25%

Fewer downtime incidents YoY



About Crest Data

AI-FIRST · CLOUD-NATIVE

Crest Data is a data and AI-first product engineering and technology solutions company with deep expertise across Agentic and Generative AI, cybersecurity, observability, data analytics, workflow automation, and cloud platforms. With a global team of **1,200+ professionals** and over **5,500 engagements for 150+ customers worldwide** — backed by partnerships with AWS, Google, Microsoft, Datadog, Dynatrace, ServiceNow, and NetApp — Crest Data helps organizations build intelligent, secure, and scalable digital systems.

BUILD ON AWS WITH CONFIDENCE

Ready to architect a resilient, observable cloud platform?

Talk to Crest Data's cloud and observability engineers about a high-availability AWS foundation — CI/CD, full-stack monitoring, and least-privilege security — tuned to your workloads.

Talk to Our Experts →

www.crestdata.ai/contact



SCAN TO EXPLORE

crest(data)

USA · CA

3031 Tisch Way #602
San Jose, CA 95128

INDIA · AHMEDABAD

CDS House, SG Road
Makarba, Ahmedabad 382210

INDIA · PUNE

Amar Madhuban Tech Park
Baner, Pune, MH 411045