Acme Platform — Q4 Engineering Report

Executive Summary

This report covers the engineering progress for **Acme Platform** during Q4 2025. Our team shipped three major features, resolved 47 production incidents, and reduced average API latency by ***38 percent***. The sections below detail each initiative, the technical decisions behind them, and our plans for Q1 2026.

> *"Ship small, ship often, and measure everything." — Acme Engineering Principles*

Acme Logo

Infrastructure Improvements

Database Migration

We migrated our primary datastore from PostgreSQL 14 to PostgreSQL 16, gaining significant performance improvements in parallel query execution. The migration was completed with *zero downtime* using logical replication.

Key metrics after migration:

Metric	Before (Q3)	After (Q4)	Change
Avg query latency	12.4 ms	7.1 ms	-42%
P99 query latency	89 ms	34 ms	-62%
Connection pool util	78%	51%	-27 pts
Daily vacuum time	45 min	18 min	-60%

Deployment Pipeline

The CI/CD pipeline was overhauled to support **parallel test execution** and ~~sequential deployments~~ rolling deployments. Build times dropped from 14 minutes to under 5 minutes.

The new pipeline configuration uses a declarative format:

```
interface PipelineConfig {
  stages: Stage[];
  parallelism: number;
  rollback: {
    automatic: boolean;
    healthCheckUrl: string;
    timeoutSeconds: number;
  };
}

function createPipeline(config: PipelineConfig): Pipeline {
  const stages = config.stages.map((stage) =>
    stage.withParallelism(config.parallelism)
  );
  return new Pipeline(stages, config.rollback);
}
```

Feature Releases

Authentication Overhaul

We replaced our legacy session-based auth with a modern token-based system. The implementation `jsonwebtoken` for signing and `bcrypt` for password hashing. See the Auth RFC for the full design document.

Benefits of the new system:

- Stateless authentication reduces server memory usage
- Token refresh flow eliminates forced logouts
  - Refresh tokens rotate on each use
  - Expired tokens trigger a silent re-auth
- Support for multiple concurrent sessions per user
- API key authentication for service-to-service calls
  - Scoped permissions per key
  - Automatic key rotation every 90 days

Search Improvements

The search backend was rewritten with the following priorities:

1. Relevance scoring using BM25 algorithm
2. Typo tolerance with Levenshtein distance
   1. Single-character edits within 2 distance
   2. Prefix matching for partial queries
3. Faceted filtering by category, date, and author

4. Response time under 100ms at the 95th percentile

Operational Highlights

Our on-call rotation handled ***47 incidents*** this quarter. The mean time to resolution (MTTR) improved from 34 minutes to 19 minutes thanks to better runbooks and automated alerting.

Incident Breakdown

Severity	Count	Avg MTTR	Top Cause
P1	3	8 min	DNS failover delay
P2	11	15 min	Memory pressure
P3	33	22 min	Config drift

The most impactful improvement was adding automated canary analysis. Previously, engineers had to `ploy` ployment — now the system runs `health-check --deep` automatically and rolls back if error rates exceed thresholds.

Q1 2026 Roadmap

Short-term Goals

- Migrate remaining services to Kubernetes
- Implement distributed tracing with OpenTelemetry
- Launch the public GraphQL API

Long-term Vision

We aim to achieve **99.99% uptime** by end of 2026. This requires investment in multi-region failover, automated chaos testing, and a dedicated platform reliability team.

Architecture Diagram

For questions or feedback, contact the engineering team at eng@acme.dev or visit the Acme Developer Portal.