"ScaleFlow" 3 Tier SaaS AWS

Project Owner: Nathaniel Kay

Status: In-Initiation

Business Case:

“ScaleFlow” is a B2B SaaS startup providing real-time data analytics. They are currently hosted on a single-server setup that experiences frequent downtime during traffic spikes. The manual deployment process is prone to error, and a single hardware failure in the data center can take the entire service offline for hours, resulting in lost revenue and customer churn.

Project Overview & Vision:

Vision: To transition ScaleFlow from a fragile single-node server to a resilient, enterprise-grade 3-tier architecture capable of 99.9% availability and autonomous scaling.

The 3-Tier Architectural Scope:

- **Tier 1:** Presentation (Web) Tier
  - Goal: User interaction and content delivery.
  - Components: Application Load Balancer (ALB) and public-facing entry points.
  - Scope: Must be the only tier accessible from the public internet.
- **Tier 2:** Application (Logic) Tier
  - Goal: Processing business logic and SaaS analytics.
  - Components: EC2 instances in an Auto Scaling Group (ASG).
  - Scope: Resides in Private Subnets. It only accepts traffic from the Web Tier (ALB).
- **Tier 3:** Data (Persistence) Tier
  - Goal: Secure storage of customer data and analytics logs.
  - Components: Amazon RDS (Relational Database Service).
  - Scope: Resides in Isolated Private Subnets. It only accepts traffic from the Application Tier.

Project Objectives:

- **High Availability:** Eliminate single points of failure so the app stays online even if a full AWS Data Center (AZ) goes dark.
- **Scalability:** Implement Auto Scaling to handle a 500% increase in morning traffic without human intervention.
- **Security:** Isolate the Database and Application logic from the public internet using private subnets.

- **Reliability:** Achieve a "Definition of Done" where a manual termination of an EC2 instance results in an automatic recovery within 2 minutes.

High-Level Requirements & Constraints:

- **Regional Deployment:** Must use at least 2 Availability Zones.
- **Networking:** Must use a custom VPC (not the default one).
- **Storage:** Database must be relational (RDS) with automatic failover enabled.
- **Cost Constraint:** Use AWS Free Tier eligible services where possible (e.g., t3.micro), but acknowledge where "Professional" features like NAT Gateways are required for the architecture.

Success Metrics (KPIs):

- **Failover Test:** Manually "crash" the primary database; the app should reconnect to the standby within 60 seconds.
- **Load Test:** Simulate traffic; the Auto Scaling Group should launch a second instance automatically.
- **Security Audit:** A public "Ping" or SSH attempt to the Database IP address must fail.