

Implementation Strategy: High-Availability E-Commerce Portal for Africa

1. Executive Summary

This document details the architectural decisions for the new e-commerce customer portal. The proposed solution is designed to address the specific challenges of the **African market**—specifically high network latency and variable connectivity—while adhering to strict **budgetary constraints** and **security requirements**.

2. Architectural Decisions & Rationale

A. The Solution (Solving for Africa)

Component: Content Delivery Network (CloudFront)

- **The Challenge:** The client requires very low-latency for their customers across Africa. Serving content from a single server location would result in slow load times for distant users.
- **The Strategy:** We place a CDN at the edge. This caches static content (specifically the "pictures for products") at edge locations physically closer to the user (e.g., Cape Town, Lagos, Nairobi).
- **The Result:** Users retrieve heavy images from a local server rather than the origin, drastically reducing load times.

B. Cost-Efficient Compute (The Budget Constraint)

Component: Auto Scaling Group with Application Load Balancer

- **The Challenge:** The business is "low on budget" but demands a system that is "fault tolerant and highly available". Buying large, fixed servers is a waste of money during low-traffic periods.
- **The Strategy:** We utilize an **Auto Scaling Group**.
 - **Elasticity:** The system adds servers (EC2 instances) only when traffic spikes and removes them when traffic drops. We pay only for what we use.
 - **Fault Tolerance:** If a server fails, the Auto Scaling Group automatically detects the failure and replaces it instantly, ensuring zero downtime.

C. Storage Strategy (Performance vs. Cost)

Component: Amazon S3 (Object Storage)

- **The Challenge:** Storing "pictures for products" in a database is expensive and slows down retrieval.
- **The Strategy:** We offload all media to **S3 Standard-Infrequent Access** or **S3 Standard**.
- **The Result:** This provides 99.999999999% durability for assets at a fraction of the cost of block storage, satisfying the "low on budget" requirement while ensuring "fast retrieval" via the CDN.

D. Data Integrity & Availability

Component: Amazon RDS (Relational Database Service) - Multi-AZ

- **The Challenge:** The system must store "user relational information" securely and remain available even during a data center failure.
- **The Strategy:** We deploy a Relational Database (PostgreSQL/MySQL) in a **Multi-AZ (Availability Zone)** configuration.
 - **Primary Instance:** Handles all traffic.
 - **Standby Replica:** A synchronized copy sits in a different physical location. In the event of a failure, the system automatically fails over to the standby without manual intervention.

3. Security Implementation

The architecture adopts a "Defense in Depth" approach to meet the requirement for high security "at rest and in transit".

- **Encryption at Rest:**
 - **S3 Buckets:** Enabled with Server-Side Encryption (SSE-S3/KMS) to protect product images.
 - **EBS Volumes & RDS:** All database and disk volumes are encrypted using AWS KMS (Key Management Service) keys.
- **Encryption in Transit:**
 - **SSL/TLS:** The Load Balancer manages an SSL certificate. All traffic from the user to the application is encrypted via HTTPS.
- **Access Control:**
 - **Private Subnets:** Application and Database servers reside in private subnets, inaccessible directly from the internet. Only the Load Balancer is public.
 - **Security Groups:** Acts as a virtual firewall, allowing traffic only on specific ports, Port 443 for Web, Port 5432 for DB from authorized sources.

4. Observability (Monitoring & Logging)

Component: CloudWatch & CloudTrail

- **The Strategy:** To "factor in monitoring and logging":
 - **CloudWatch Alarms:** Set to trigger if CPU usage spikes (scaling event) or if 5xx Error rates increase (application failure).
 - **Access Logging:** S3 and Load Balancer access logs are enabled to track requests for security auditing.

5. Conclusion

This architecture delivers a robust, enterprise-grade solution on a startup budget. By leveraging **serverless storage (S3)**, **edge caching (CDN)**, and **auto-scaling compute**, we meet the strict latency requirements for the African market while ensuring the platform remains secure and highly available.