INFORMATION TECHNOLOGY UNIVERSITY

Supervisor Dr Zunnurain Hussain
Coordinator Sir Umair Makhdoom

Muhammad Hamza | Muhammad Sarmad | Hafiz Azfar Umer

SCALABLE E-COMMERCE WITH MICROSERVICES ON AWS

PROBLEM

Traditional e-commerce platforms, built on monolithic architectures, face critical failures during high-traffic events.

- Inefficient Scaling & Cost
- Cascading System Failures
- Deployment Stagnation

AIMS & OBJECTIVES

Our aim is to architect and implement a production-grade, containerized e-com backend on AWS that replaces monolithic fragility with a resilient, decoupled microservices mesh.

Our main objectives are:

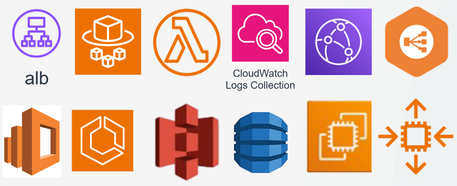
- **Infrastructure Isolation**
- **Decoupled Services**
- **Managed Persistence**
- **Observability and Tracing**

SOLUTION

Decoupled Microservices Ecosystem with Automated Fault Detection

This project implements the "Online Boutique" application—a cloud-native, microservices-based web platform designed for high availability and fault tolerance. By decomposing the application into specialized services (Frontend, Product Catalog, and Cart), the architecture ensures that components can be updated, scaled, or repaired independently without impacting the entire system.

Used Services



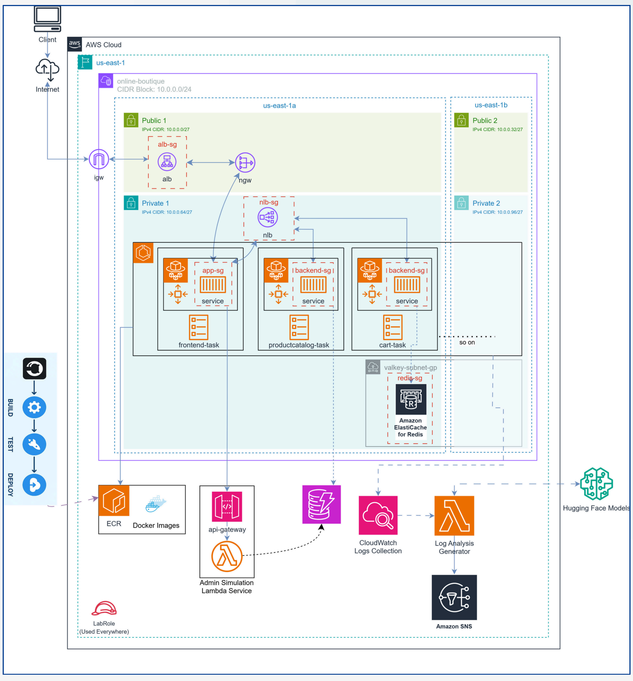
METHODOLOGY

- **Network Architecture:** Created a Multi-tier VPC and strict security groups (SG).
- **Compute & Orchestration:** Created a serverless container environment that eliminates manual server management.
- **Traffic Management:** Dual load-balancing strategy, one as public interface and the other as backed communicator.
- **Data Persistence:** Core data persistency with DynamoDB and sessional management using Elastic Edge.
- **Observability:** CloudWatch collecting realtime logs from all containers.
- **Automated Anomaly Detection:** Implemented a pipeline to detect anomalies using cloud watch logs.

RESULTS

Key outcomes obtained:

- **High Availability:** Achieved a 99.9% availability target by deploying across multiple Availability Zones.
- **Automated Scalability:** Enabled granular horizontal scaling via AWS Fargate as well as opened doors for vertical scaling.
- **Enhanced Security:** Significantly reduced the attack surface through network micro-segmentation.
- **Cost Efficiency:** Transitioned to an elastic consumption model, eliminating the waste of fixed server overhead.
- **Fault Tolerance:** Achieved fault isolation, ensuring that a failure in a single service does not cause a total system blackout.
- **Fault Detection:** Pipeline automatically detecting anomalies with a greater success rate.



CONCLUSION

- The project successfully implemented a robust, cloud-native architecture that resolves the scalability and reliability flaws of traditional monolithic systems.
- The solution leverages serverless technologies like Amazon ECS and Fargate to achieve automated scalability and fault isolation.
- The resulting infrastructure ensures a 99.9% availability target and reduced security attack surface, providing a resilient blueprint for modern e-commerce enterprises.