



Information Technology University Punjab

Study Group & Peer Tutoring Platform

PROJECT PROPOSAL, EXECUTIVE SUMMARY & PROBLEM DEFINITION

Prepared by:

Hazira Azam Bsse23019

Areeba Shahbaaz Bsse23097

Supervisor:

Sir Zunnurain Hussain

Teaching Assistant: Sir Umair Makdoom

Executive Summary:

1. Overview

The StudyCloud project addresses the growing need for a scalable, reliable, and secure academic collaboration platform. By moving away from traditional, monolithic web architectures, this project implements a Cloud-Native Microservices Architecture on Amazon Web Services (AWS). The result is a high-availability environment capable of supporting seamless student-tutor interactions and study group management with zero-downtime updates.

2. Problem Identification

Modern educational platforms often face three critical challenges that this project sought to solve:

Scalability Bottlenecks: Traditional servers often crash during peak academic periods (e.g., midterms or finals) due to a sudden surge in concurrent users.

Service Interruption: Standard deployment methods require "maintenance windows," where the platform must be taken offline to apply updates or fix bugs.

Data Vulnerability: Many student platforms lack proper network isolation, leaving sensitive user data and session history exposed to the public internet.

3. The Solution: StudyCloud Architecture

The proposed solution replaces local-storage-dependent logic with a robust, backend-driven cloud infrastructure:

Infrastructure as a Service (IaaS): Built within a custom AWS VPC utilizing public/private subnetting. This ensures that the backend logic and the Amazon RDS database remain entirely isolated from the public internet.

Containerization & Orchestration: All services (Auth, Tutoring, Groups) were containerized using Docker and orchestrated via Amazon ECS Fargate. This allows the system to auto-scale horizontally based on real-time demand.

Automated CI/CD: A fully automated pipeline was established using AWS CodePipeline. By integrating GitHub with AWS CodeBuild, the system automatically builds and pushes new container images to Amazon ECR.

4. Strategic Implementation (Blue-Green Deployment)

To eliminate downtime, we implemented a Blue-Green Deployment strategy via AWS CodeDeploy.

Mechanism: Two separate environments (Target Groups) are maintained. The "Green" environment is updated and tested in isolation while the "Blue" environment serves live traffic.

Benefit: Once verified, traffic is shifted to the Green environment via the Application Load Balancer (ALB). If any issues arise, the system triggers an automatic rollback, ensuring 100% service availability.

5. Performance Verification

The implementation was validated through rigorous testing:

Stress Testing: The Application Load Balancer was tested for high-concurrency handling, maintaining a stable "Healthy Host Count" throughout the sessions.

Observability: Using Amazon CloudWatch, we established real-time monitoring for CPU and Memory utilization, ensuring that resource allocation was optimized for cost-efficiency.

6. Conclusion

The StudyCloud deployment successfully demonstrates how modern cloud engineering can transform a simple web application into a resilient enterprise-grade platform. Through the strategic use of AWS managed services and DevOps automation, we have created a secure, scalable ecosystem that provides a seamless experience for the academic community while significantly reducing operational overhead.
