A logo for a company

AI-generated content may be incorrect.

AWS First Cloud AI Journey – **Project Plan**

Project team – FPT University – Server-less Student Management System

[2025-12-07]

**Table of Contents**

**[1](#_ghona9kr41df)****BACKGROUND and motivation** 3

[1.1](#_i7uchkls85k7) executive summary 3

[1.2](#_lsktm7gdqhr7) PROJECT SUCCESS CRITERIA 3

[1.3](#_b1njzurv7rfo) Assumptions 4

**[2](#_x8p0ymcuyi3y)****SOLUTION ARCHITECTURE / ARCHITECTURAL DIAGRAM** 4

[2.1](#_5kn1dfomfhz1) Technical Architecture Diagram 4

[2.2](#_iuhm1hihj33t) Technical Plan 5

[2.3](#_4utuwf2ef5g9) Project Plan 5

[2.4](#_dgvinz7m5o5l) Security Considerations 5

**[3](#_r1ilk8bt9i9a)****Activities AND Deliverables** 6

[3.1](#_7w8qpwsq0hd1) Activities and deliverables 6

[3.2](#_in5gbsjij0q) OUT OF SCOPE 7

[3.3](#_3f1wf4qh9mcz) PATH TO PRODUCTION 7

**[4](#_av4tkvho8vb1)****EXPECTED AWS COST BREAKDOWN BY SERVICES** 8

**[5](#_u06c6xaw9tf5)****TEAM 6**

# BACKGROUND and motivation

## executive summary

The **Serverless Student Management System** is a student management platform built entirely on AWS Serverless technologies. The goal is to deliver a modern, cost-efficient, scalable solution for managing students, courses, assignments, and grades using AWS services such as API Gateway, Lambda, Dynamo Db, Cognito, and Amplify*.*

**Customer background:**

This project is designed for teachers, students, and anyone learning or working with AWS Cloud who wants hands-on experience with building a real-world serverless system.

**Business & technical objectives:**

- Build a truly 100% serverless student management application

- Optimize operational cost (under $20/month)

- Fully implement CI/CD from GitLab → CodePipeline

- Build a complete full-stack system (React/Amplify + API Gateway + Lambda + DynamoDB)

**Use cases:**

- Student management (CRUD, search, import)

- Course, assignment, and grade management

- Admin/teacher dashboard

- Authentication & authorization with Cognito

**Professional services delivered:**

- Serverless architecture design following AWS Well-Architected Framework

- Backend, frontend, CI/CD implementation

## PROJECT SUCCESS CRITERIA

- One API Gateway with resources `/` and `/{proxy+}` (proxy resource), using ANY and OPTIONS methods, proxy integration forwards all requests to 1 Lambda Function.

- 1 AWS Lambda functions with >80% unit test coverage

- 5 fully functional modules: Students, Courses, Grades, Assignments, Dashboard

- React/TypeScript frontend deployed via Amplify with 15+ pages

- API response time < 500ms

- Uptime ≥ 99%

- CI/CD automated from GitLab → CodeBuild → CodeDeploy/Amplify → Production

- Infrastructure cost maintained < $20/month

## Assumptions

- AWS Free Tier is used throughout the project

- All environments use serverless services only (no EC2)

- Active users < 200 per month

- API calls < 500k/month

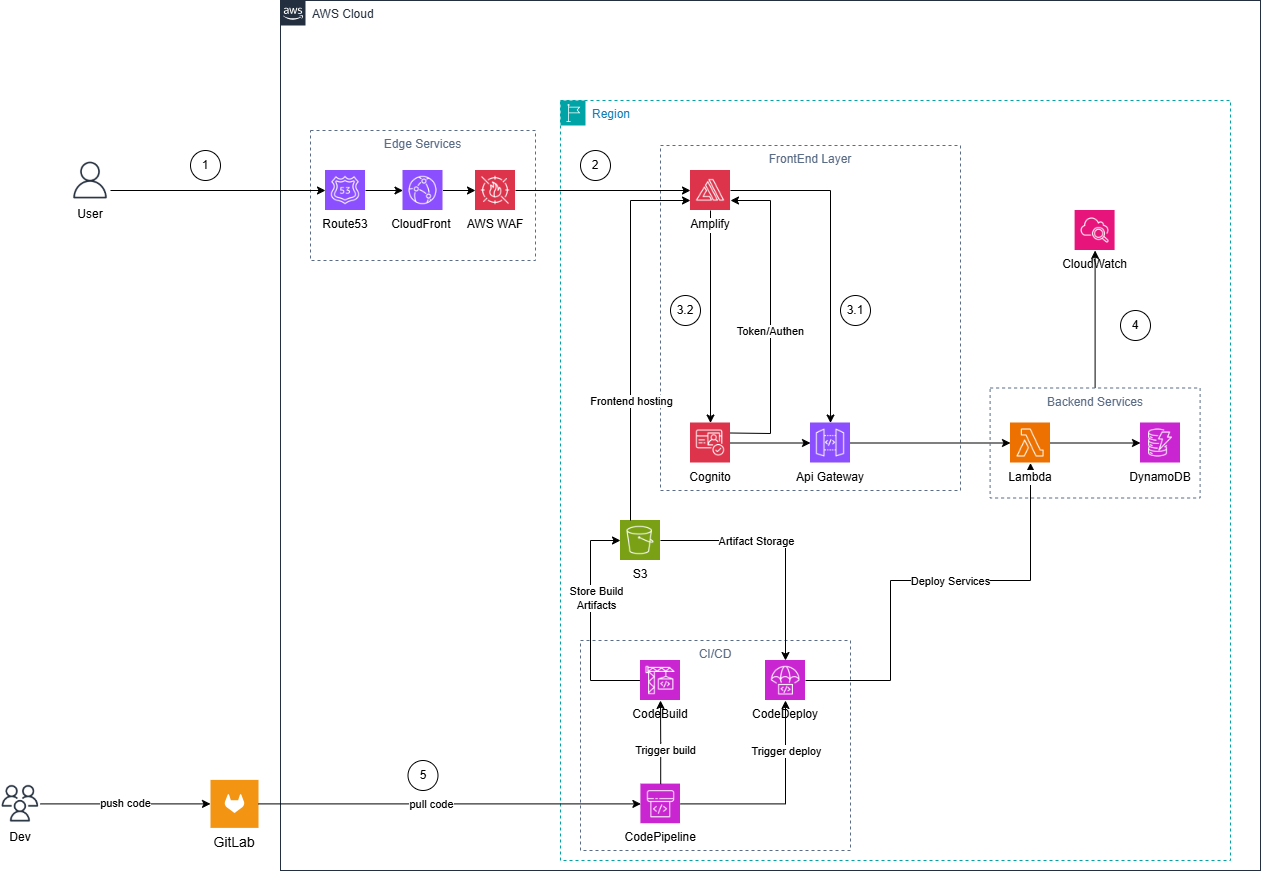
- DynamoDB data volume < 10GB

- Team has basic knowledge of React, Java, and AWS Cloud

- Services like Route53 domain and WAF may incur costs but are included in scope

# SOLUTION ARCHITECTURE / ARCHITECTURAL DIAGRAM

## Technical Architecture Diagram



## Technical Plan

The system is built using AWS Serverless Architecture, consisting of 5 major layers:

- **Edge Layer**:Route53, CloudFront, WAF

- **Frontend Layer**: Amplify, Cognito

- **Backend Layer**: API Gateway, Lambda, DynamoDB

- **Monitoring**: CloudWatch

- **CI/CD**: GitLab, CodePipeline, CodeBuild, CodeDeploy, S3

Main API groups:

- `/students/` (CRUD, search, import/export)

- `/courses/`

- `/grades/`

- `/assignments/`

- `/auth/`

## Project Plan

The project will be implemented in **5 weeks**:

- **Week 1:** AWS setup, IaC, DynamoDB, Cognito, Git repository

- **Week 2:** Backend & Frontend Phase 1 (Lambda + API Gateway + basic UI)

- **Week 3:** Backend & Frontend Phase 2 (15+ pages, integration test, CloudFront + WAF)

- **Week 4:** CI/CD (CodeBuild, CodeDeploy, CodePipeline)

- **Week 5:** Testing, optimization, documentation, demo

## Security Considerations

- IAM least privilege

- Cognito User Pool + JWT authentication

- API Gateway throttling (rate limiting)

- WAF rules (SQLi/XSS, IP block, rate limiting)

- DynamoDB encryption

- CloudWatch alarms

# Activities AND Deliverables

## Activities and deliverables

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Project Phase** | **Timeline** | **Activities** | **Deliverables/Milestones** | **Total man-day** |
| Assessment | Week 1 | * Setup AWS account, IAM, billing alerts * Create DynamoDB + GSI tables * Configure Cognito user groups and groups * Design detailed architecture, sequence diagrams * Initialize IaC structure (CDK/CloudFormation), Git repo | * AWS foundation complete * DynamoDB + Cognito operational * Architecture documentation + workflow diagrams * Repo & IaC ready | [X man-day] |
| Setup base infrastructure | Week 2 | * Backend: build Lambda (Students, Courses, Grades, Assignments, Auth) * Setup API Gateway endpoints * Unit tests 80%+ * Frontend: implement Amplify, routing, UI core layout * Integrate state management, API client | * Lambda functions working * API Gateway stable * Frontend base UI complete * Unit test passed1 | [X man-day] |
| Setup component 1 | Week 3 | * Backend: automatic workflow integration, integration testing * Frontend: build 15+ pages (Students, Courses, Grades, Assignments) * CloudFront connection – Route53 – WAF – SSL/TLS * End-to-end integration FE ↔ BE | * Complete backend workflow * Full operational management page * Domain + CDN + WAF + HTTPS running smoothly * E2E test passed | [X man-day] |
| Setup component 2 (CI/CD) | Week 4 | * Configure CodeBuild for FE & BE * Set up CodePipeline (source → build → deploy) * Configure CodeDeploy to deploy backend & frontend * Test the entire pipeline | * Complete CI/CD pipeline * Automatic build & deploy * Test pipeline report | [X man-day] |
| Testing & Golive | Week 5 | * E2E testing, edge cases, load testing 100+ users * Performance optimization (Lambda, DynamoDB, CloudFront) * Standardization of technical documentation, user manual * Demo preparation and testing | * Load test report * Optimized and stable system * Full documentation (architecture, API, deployment guide) * Complete demo | [X man-day] |

## OUT OF SCOPE

- Mobile Applications (iOS/Android)

- Advanced Email/SMS OTP authentication

- AI recommender system / ML scoring

- Multi-region failover

- Enterprise SSO (SAML, OIDC)

## PATH TO PRODUCTION

- Commit code → GitLab

- CodePipeline (source stage)

- CodeBuild (frontend & backend build)

- CodeDeploy (Lambda deployment + Amplify)

- CloudWatch (logs & monitoring)

- DNS routing via Route53 + CloudFront

# EXPECTED AWS COST BREAKDOWN BY SERVICES

[AWS monthly pricing calculator](https://calculator.aws/" \l "/estimate?id=a32a503d312abf60cdc634a02c8a860d85fd286c)

|  |  |  |
| --- | --- | --- |
| Service | Monthly Estimate | Notes |
| API Gateway | ~$3.8 | 100k–500k requests |
| Lambda | ~$0.5 | Free Tier includes 1M requests |
| DynamoDB | ~$1 | <10GB data |
| Cognito | $0 | <10k MAU |
| S3 | ~$0.13 | Artifacts & hosting |
| CloudWatch | ~$4 | Logs & metrics |
| Amplify | ~$1 | Frontend hosting |
| Route53 | ~$1 | DNS hosted zone |
| WAF | ~$7 | Web ACL |
| CodeBuild | ~$0.9 | ~100 build minutes |
| CodePipeline | ~$1 | 1 pipeline |
| CloudFront | ~$1 | CDN traffic |

**Total Estimated Cost: ~ $7 – $20 / month**

**For 3 months: ~$21 – $60 (can be reduced to $15–30 with Free Tier).**

# 

# TEAM

**Partner Project Team**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Title | Role | Email / Contact Info |
| Nguyễn Hoàng Nam | FCJ Cloud Intern | Project Lead/ Backe-End | namnhse182076@fpt.edu.vn |
| Phạm Bảo Phi | FCJ Cloud Intern | Full Stack | phipbse185046@fpt.edu.vn |
| Nguyễn Nhật Kim Ngân | FCJ Cloud Intern | Frontend /BA | ngannnkse182088@fpt.edu.vn |
| Đặng Huy Hoàng | FCJ Cloud Intern | Back-End/DevOps | hoangdhse184661@fpt.edu.vn |
| Lâm Thanh Phúc | FCJ Cloud Intern | Back-End/DevOps | phucltse184678@fpt.edu.vn |