



AWS First Cloud AI Journey – Project Plan

Bug Hunters – FPT University
IELTS Learning Platform

9/12/2025

TABLE OF CONTENTS

1	BACKGROUND AND MOTIVATION	3
1.1	EXECUTIVE SUMMARY.....	3
1.2	PROJECT SUCCESS CRITERIA	3
1.3	ASSUMPTIONS	3
2	SOLUTION ARCHITECTURE / ARCHITECTURAL DIAGRAM	4
2.1	TECHNICAL ARCHITECTURE DIAGRAM	4
2.2	TECHNICAL PLAN.....	6
2.3	PROJECT PLAN	6
2.4	SECURITY CONSIDERATIONS	6
3	ACTIVITIES AND DELIVERABLES	7
3.1	ACTIVITIES AND DELIVERABLES.....	7
3.2	OUT OF SCOPE.....	8
3.3	PATH TO PRODUCTION	8
4	EXPECTED AWS COST BREAKDOWN BY SERVICES	8
5	TEAM	8
6	RESOURCES & COST ESTIMATES	9
7	ACCEPTANCE	10

1 BACKGROUND AND MOTIVATION

1.1 EXECUTIVE SUMMARY

The customer is developing a next-generation digital learning platform to support students preparing for the IELTS examination. The current approach relies on fragmented tools, manually curated content, and limited automation, resulting in inconsistent learning experiences and minimal performance insights. To address scalability, reliability and the need for modern, AI-supported learning features, the customer aims to standardize its platform on a cloud-native architecture that can grow alongside user demand.

The primary business objective is to deliver a unified, always-available IELTS learning environment that improves learner outcomes while reducing operational overhead. AWS provides the customer with a secure, elastic, cost-efficient foundation capable of supporting high traffic volumes, real-time collaboration, and intelligence-driven learning flows.

Technically, the customer requires a platform that supports rapid feature development, integrates seamlessly with AI services, and leverages managed cloud offerings to minimize infrastructure management. Moving to AWS enables strong architectural governance, multi-AZ resiliency, built-in observability, and scalable compute patterns aligned with expected user growth.

- A learner signs up and accesses consolidated IELTS preparation content, practice tests and progress tracking
- Students join real-time study rooms with WebRTC-based communication for collaborative learning
- Users submit writing samples or speaking recordings, which are evaluated automatically through AI pipelines
- Personalized flashcards are generated from user performance history to reinforce key concepts
- Administrators manage users, content, performance analytics and learning recommendations

These use cases reflect both current functional requirements and a foundation for future expansion into AI-assisted tutoring, adaptive assessments, and multi-exam support.

The partner will design and deliver a fully cloud-native web platform on AWS, including architecture definition, infrastructure provisioning, and application development for all core learning modules. Services include backend and frontend implementation, integration of AI scoring engines, setup of CI/CD pipelines, security controls, and multi-environment deployment. The partner will also conduct testing, performance tuning, documentation, knowledge transfer, and production readiness activities to ensure the platform meets all technical and business objectives.

1.2 PROJECT SUCCESS CRITERIA

- Provide a unified learning platform covering user management, content creation, collaboration, AI evaluation and flashcards.
- Achieve at least **99.9% uptime** using Multi-AZ AWS architecture.
- Support **10,000+ concurrent users** without performance degradation.
- Deliver AI writing/speaking assessment results in **under 10 seconds** for 95% of requests.
- Maintain median backend API latency **under 200ms** under normal load.
- Implement fully automated CI/CD pipelines supporting zero-downtime deployments.
- Maintain an AWS steady-state infrastructure cost of approximately **300 USD per month**.

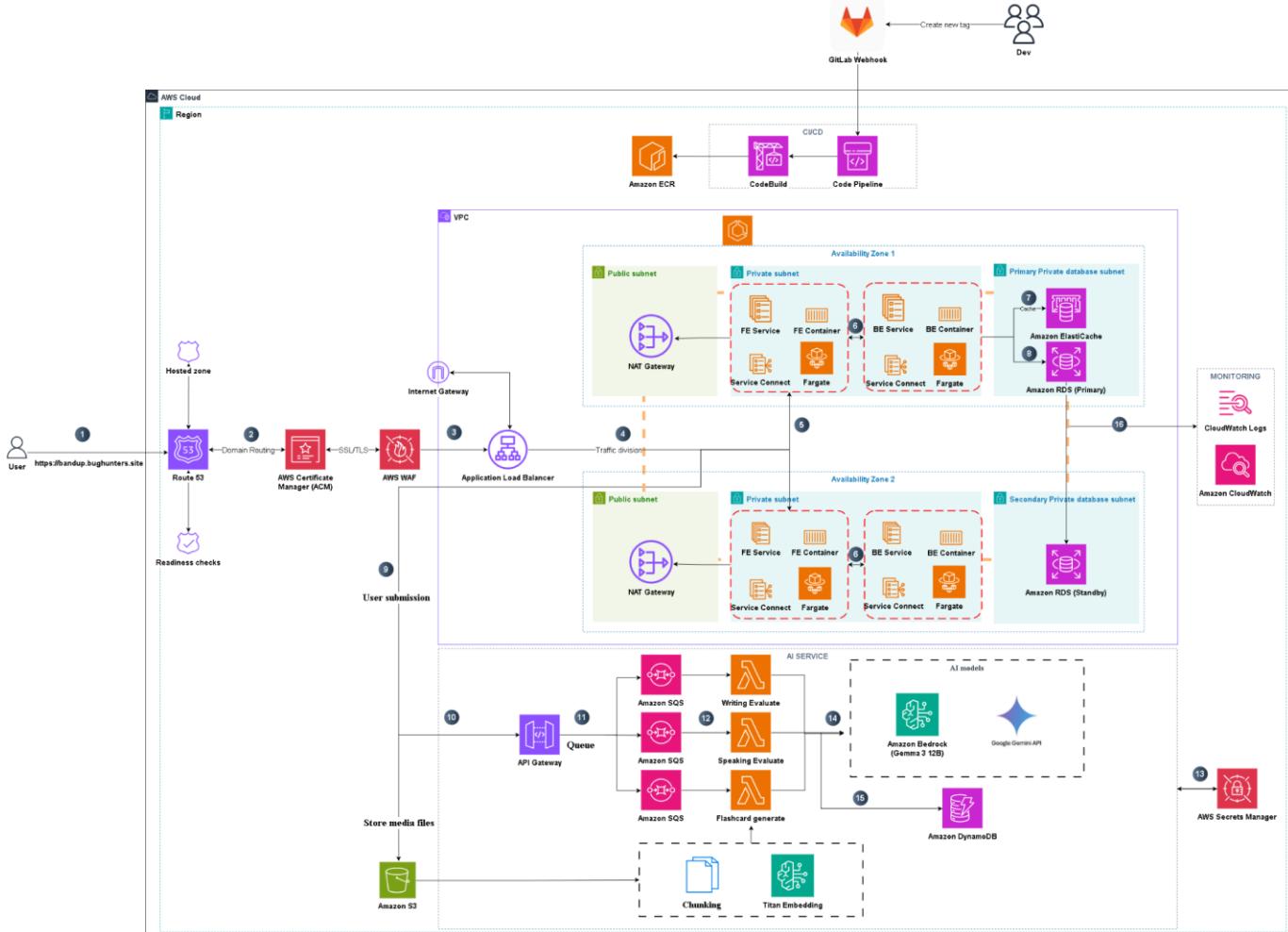
1.3 ASSUMPTIONS

- AWS accounts, IAM roles, VPC and networking foundations are provisioned and accessible to the delivery team
- IELTS content, datasets, and media are provided and licensed for use
- External AI APIs (Gemini Flash, Amazon Bedrock) remain accessible within agreed quotas
- End-users access the platform through modern browsers with stable network connectivity

- The project team and customer follow an Agile project delivery process with two-week sprints
- Scope of delivery is limited to web functionality; no native mobile applications are included

2 SOLUTION ARCHITECTURE / ARCHITECTURAL DIAGRAM

2.1 TECHNICAL ARCHITECTURE DIAGRAM



The proposed solution is deployed into an AWS Virtual Private Cloud (VPC) spanning multiple Availability Zones to provide high availability and fault tolerance. Public subnets host the internet-facing Application Load Balancer (ALB), which terminates TLS and routes traffic to containerized workloads running on Amazon ECS Fargate in private subnets. The frontend is implemented using Next.js and exposed as a service behind the ALB, while the backend is implemented in Spring Boot and exposed as internal APIs, also on ECS Fargate. All application traffic between components is secured using security groups and restricted to least-privilege network paths.

Application data is persisted in Amazon RDS for PostgreSQL configured in Multi-AZ mode to provide resilience and automated failover. Frequently accessed or transient data is cached using Amazon ElastiCache for Redis to reduce latency and offload read pressure from the database. Static assets, such as documents, images, audio files and other learning content, are stored in Amazon S3 and distributed through Amazon CloudFront for low-latency global access.

AI-driven workloads, including automated IELTS writing and speaking evaluations and flashcard generation, are orchestrated using serverless patterns. The backend invokes Amazon API Gateway endpoints, which place requests onto Amazon SQS queues. AWS Lambda functions consume these requests, enrich them where necessary, and invoke external AI services such as Google Gemini Flash and Amazon Bedrock models. The results are written back to the application data stores and made available to learners via the web interface.

Security is enforced at multiple layers. IAM roles and policies follow least-privilege principles for all services. AWS WAF is optionally used in front of the ALB to mitigate common web threats. All data in transit is encrypted using TLS, and all data at rest in RDS, ElastiCache, S3 and DynamoDB is encrypted using AWS Key Management Service (KMS). Monitoring and logging are implemented via Amazon CloudWatch, AWS CloudTrail and AWS Config to provide operational visibility, audit trails and configuration compliance.

Deployment and operations are automated through CI/CD pipelines that build, test and deploy container images to Amazon ECS Fargate. Infrastructure is defined using Infrastructure-as-Code, ensuring that environments (development, staging, production) are reproducible and consistent. Operational dashboards and alarms are configured to monitor key metrics such as latency, error rates, resource utilization and AI workload performance, allowing the operations team to quickly identify and remediate issues.

- **AWS Core Infrastructure**
 - o Amazon VPC, Subnets, Internet Gateway, NAT Gateway
 - o AWS Identity and Access Management (IAM)
 - o AWS WAF (optional) and Security Groups
- **Compute and Application Layer**
 - o Amazon ECS Fargate (Next.js frontend and Spring Boot backend services)
 - o Application Load Balancer (ALB)
- **Data and Storage**
 - o Amazon RDS for PostgreSQL (Multi-AZ)
 - o Amazon ElastiCache for Redis
 - o Amazon S3 for static content and media
 - o (Optional) Amazon DynamoDB for high-speed key-value or metadata storage
- **AI and Integration**
 - o Amazon API Gateway
 - o Amazon SQS
 - o AWS Lambda
 - o Amazon Bedrock (for model hosting where applicable)
 - o Google Gemini Flash API (for IELTS writing/speaking evaluation where permitted)
- **Networking, Monitoring and Security**
 - o Amazon CloudWatch (metrics, logs, alarms)
 - o AWS CloudTrail
 - o AWS Config
- **Development & DevOps Tooling**
 - o Git-based source control (e.g., GitHub or GitLab)
 - o Containerization with Docker
 - o CI/CD pipelines (e.g., GitHub Actions / GitLab CI) to build and deploy to ECS Fargate
 - o Infrastructure-as-Code (e.g., AWS CloudFormation or Terraform)

2.2 TECHNICAL PLAN

The proposed solution will be implemented using a fully automated, Infrastructure-as-Code–driven delivery model to ensure consistent, repeatable, and secure deployments across development, staging and production environments. All foundational components—including VPC networking, ECS Fargate services, RDS PostgreSQL, ElastiCache Redis, S3 buckets, CloudFront distributions, IAM roles and service integrations—will be provisioned via Infrastructure-as-Code (IaC) to ensure traceability and version control.

Application workloads will be containerized using Docker and deployed onto Amazon ECS Fargate through CI/CD pipelines. The backend (Spring Boot) and frontend (Next.js) services will follow a blue/green or rolling deployment model to minimize downtime and reduce operational impact. Configuration values, environment variables, database credentials and API keys will be securely managed through AWS Systems Manager Parameter Store or AWS Secrets Manager.

AI-related components—including the writing and speaking evaluation workflows—will be implemented as serverless pipelines using API Gateway, SQS and Lambda. These workflows will integrate with Amazon Bedrock or Google Gemini Flash to process scoring requests in a scalable, asynchronous manner. Successful and failed AI jobs will be logged in CloudWatch for observability and troubleshooting.

To maintain high operational quality, all critical paths (authentication, study room communication, practice test submission, AI evaluation, flashcard generation) will be covered by automated tests integrated into the CI/CD process. Each pipeline stage will execute unit tests, static code analysis, vulnerability scans and integration tests before deployment. Additional performance and load tests will be executed prior to production release to validate autoscaling thresholds, database performance and AI workload throughput.

The technical plan ensures that the system adheres to AWS Well-Architected principles—operational excellence, security, reliability, performance efficiency and cost optimization. This approach provides the customer with a stable, scalable and maintainable platform suitable for long-term evolution.

2.3 PROJECT PLAN

The delivery follows a 12-week Agile project structure. Early sprints finalize requirements, architecture and infrastructure setup. Subsequent sprints focus on implementing the core functional modules: user management, blog system, study rooms, practice test engine, and flashcard generation. AI evaluation pipelines are integrated after the core modules are stable. Final sprints deliver non-functional testing (performance, load, security), production readiness, documentation and operational handover.

2.4 SECURITY CONSIDERATIONS

- IAM least-privilege enforcement, MFA for privileged roles, and RBAC for application users
- All compute components deployed in private subnets; ingress restricted via Security Groups and AWS WAF
- TLS enforced for all traffic; encryption at rest enabled for RDS, DynamoDB, S3 and ElastiCache
- CloudTrail, CloudWatch Logs and AWS Config enabled for auditing, alerting and compliance
- Regular patching and image updates maintained through CI/CD automation
- Backup and restore procedures defined for database and application state

3 ACTIVITIES AND DELIVERABLES

3.1 ACTIVITIES AND DELIVERABLES

[Provide project milestones with timeline and respective deliverables, corresponding to the items and activities described in the Scope of Work / Technical Project Plan section. Indicate plans on how to govern the project/ change management; communication plans; transition plans]

Project Phase	Timeline	Activities	Deliverables/Milestones	Total man-day
Assessment	Week 1-2	<ul style="list-style-type: none">• Requirements gathering• Business validation• High-level architecture definition	<ul style="list-style-type: none">• Approved technical specification	12 man-days
Setup base infrastructure	Week 3-4	<ul style="list-style-type: none">• Provision VPC, subnets, security groups• Configure ECS Fargate (FE/BE services)• Deploy RDS PostgreSQL Multi-AZ• Set up S3, CloudFront, ElastiCache Redis	<ul style="list-style-type: none">• AWS infrastructure baseline ready (Dev + Staging)	22 man-days
User Management & Blog System	Week 5-6	<ul style="list-style-type: none">• Implement user registration, authentication, and authorization• Configure JWT + role-based access control• Build Blog module (CRUD, categories, tagging)• Develop foundational Next.js UI components and layouts	<ul style="list-style-type: none">• User management module functional• Blog system functional	28 man-days
Study Rooms, Practice Tests & AI Learning Components	Week 7-8	<ul style="list-style-type: none">• Build Study Room module (WebRTC video/voice, shared workspace, chat• Implement Practice Test Engine (Reading + Listening)• Integrate AI Writing Assessment	<ul style="list-style-type: none">• Study Room and Practice Test modules functional• Writing and Speaking AI assessment pipelines operational• Flashcard generation and review system available	36 man-days

		<p>pipeline (S3 → SQS → Lambda → Bedrock/Gemini)</p> <ul style="list-style-type: none"> • Integrate AI Speaking Assessment pipeline (audio transcription + scoring) • Implement Flashcard generation and spaced repetition workflow 		
Testing & Go-live	Week 9-10	<ul style="list-style-type: none"> • Integration testing • Performance testing • Security validation • Production deployment 	<ul style="list-style-type: none"> • Production environment deployed 	18 man-days
Handover	Week 11-12	<ul style="list-style-type: none"> • Knowledge transfer • Documentation & runbooks • Monitoring dashboards 	<ul style="list-style-type: none"> • Project handover completed 	10 man-days

3.2 OUT OF SCOPE

- Native iOS/Android applications
- Offline learning capabilities
- Third-party LMS integration unless separately scoped
- Additional exam types outside IELTS (e.g., TOEFL, SAT, PTE)
- Advanced analytics dashboards beyond the defined scope

3.3 PATH TO PRODUCTION

The production release will be executed following standard deployment procedures, including final performance tuning, autoscaling validation, observability review, backup/restore verification, and incident response preparedness. After go-live, early lifecycle monitoring will identify refinement opportunities, which may be addressed in subsequent enhancement sprints or as a separate engagement.

4 EXPECTED AWS COST BREAKDOWN BY SERVICES

IELTS Learning Platform (Band-Up) Infrastructure Cost per month

5 TEAM

Partner Executive Sponsor

Name	Title	Description	Email / Contact Info

Project Stakeholders

Name	Title	Stakeholder for	Email / Contact Info

Partner Project Team

Name	Title	Role	Email / Contact Info
Dang Phuong Nam	Software Engineer	Frontend Developer	namdangcoder@gmail.com
Dang Minh Hoang	Software Engineer	Backend Developer	hoangsusp@gmail.com
Nguyen Quoc Bao	AI Engineer	AI Engineer	quocbao0905461606@gmail.com
Vo Hoang Minh	Software Engineer	Backend Developer	minh7n3@gmail.com
Le Nhat Anh	Software Engineer	Frontend Developer	lenhatanh2411@gmail.com

Project Escalation Contacts

Name	Title	Role	Email / Contact Info

6 RESOURCES & COST ESTIMATES

Resource	Responsibility	Rate (USD) / Hour
Solution Architects [number of assigned headcount]		
Engineers [number of assigned headcount]		
Other (Please specify)		

* Note: Refer to section “activities & deliverables” for the list of project phases

Project Phase	Solution Architects	Engineers	Other (Please specify)	Total Hours
Total Hours				
Total Cost				

Cost Contribution distribution between Partner, Customer, AWS:

Party	Contribution (USD)	% Contribution of Total
Customer		
Partner		
AWS		

7 ACCEPTANCE
