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

Strangers – FPT University – **FLYORA**
E-commerce system

06/12/2025

Table of Contents

1 BACKGROUND AND MOTIVATION

1.1 EXECUTIVE SUMMARY

[Customer background] *Flyora is a specialized web application designed to serve bird enthusiasts across Vietnam.*

[Business and technical objectives - drivers for moving to the AWS cloud] *It offers curated products such as bird food, toys, cages, and decorative accessories tailored to species like Chào Mào, Vẹt, Yến Phụng, and Chích Chòe. Built with modern web technologies and hosted on AWS, Flyora ensures scalability, performance, and secure access. The platform aims to become the go-to destination for bird care and ornamentation, combining e-commerce with personalization and community engagement.*

[Use cases] *Flyora addresses current challenges: no centralized platform for bird-specific products, generic pet stores lacking species-specific recommendations, poor mobile responsiveness and outdated UI in existing platforms, and limited backend scalability and search capabilities.*

[Briefly summarize the partner's professional services to be delivered to meet the customer's objectives] *Flyora delivers a responsive, category-driven shopping experience with secure user authentication, real-time product filtering, and a scalable backend. It supports both desktop and mobile users, with future plans for AI-powered recommendations and chatbot support.*

1.2 PROJECT SUCCESS CRITERIA

Success will be measured by the ability to achieve the following:

- *Create a centralized platform for bird lovers in Vietnam.*
- *Provide a responsive, mobile-friendly User Interface/User Experience (UI/UX).*
- *Establish secure user authentication and role management (via IAM).*
- *Build a scalable backend with Lambda/API Gateway.*
- *Support real-time product filtering and chatbot capabilities.*
- *Utilize AWS X-ray for application tracing and analysis.*
- *Lay the foundation for future AI features via Bedrock (Embedding/LLM).*

1.3 ASSUMPTIONS

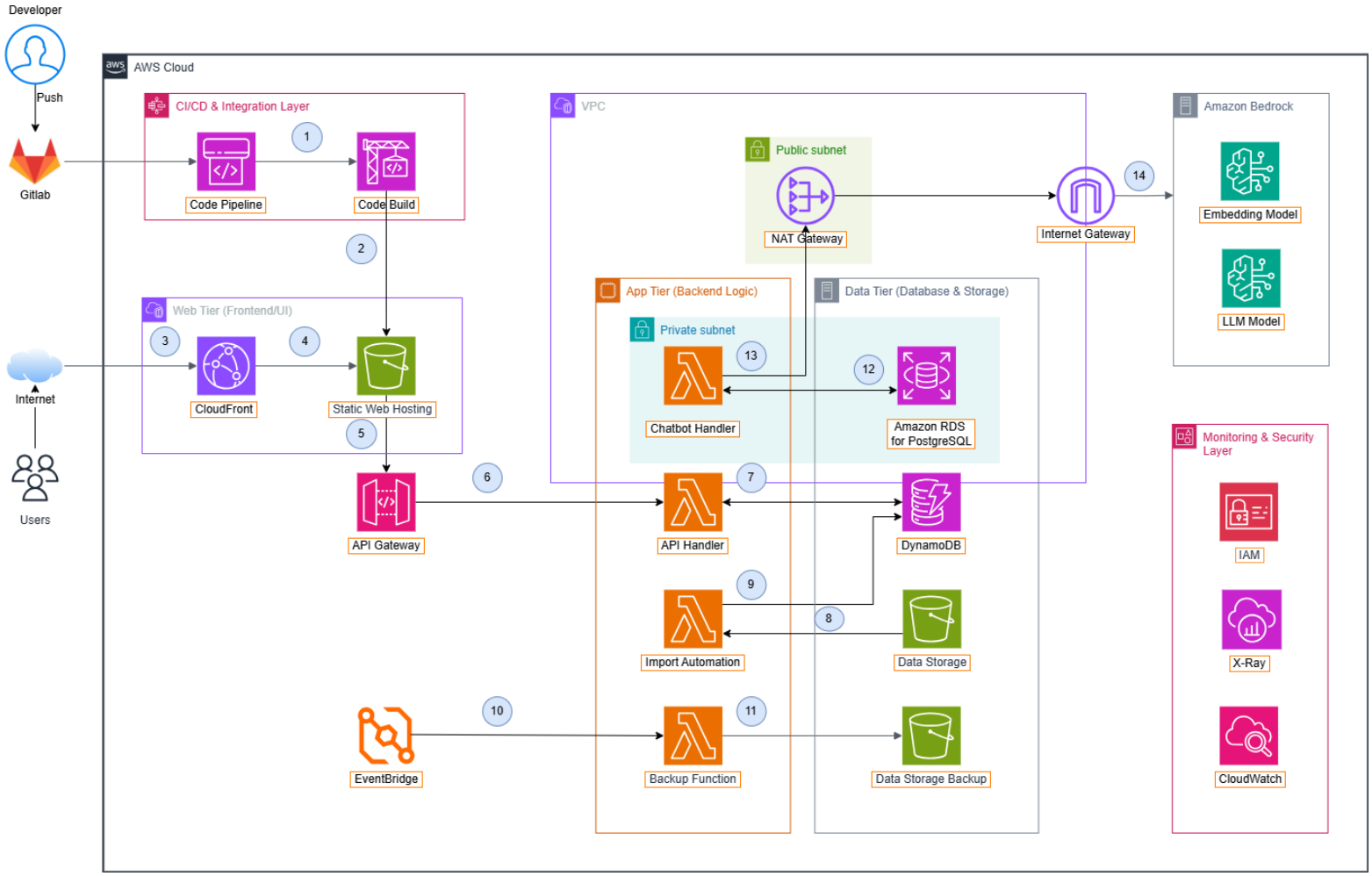
- *Proficiency in AWS services is required for deploying the serverless architecture.*
- *Frontend development will utilize S3 static hosting.*
- *DynamoDB will be used for NoSQL data management, and RDS for PostgreSQL for relational data.*
- *GitHub will be used for version control and CI/CD integration.*
- *Identified risks requiring mitigation include Lambda cold starts (using provisioned concurrency), DynamoDB throttling (using auto-scaling), and RDS downtime (using Multi-AZ deployment and automated backups).*
- *Hardware costs are not applicable as Flyora is a web-only platform.*

2 SOLUTION ARCHITECTURE / ARCHITECTURAL DIAGRAM

2.1 Technical Architecture Diagram

The proposed architecture follows a serverless model:

- ***Web Tier (Frontend):*** *Amazon S3 is used for static web hosting, distributed via CloudFront CDN.*
- ***Application Tier (Backend):*** *Amazon API Gateway manages HTTP APIs. AWS Lambda Functions handle business logic, including the chatbot handler, import automation, and API handlers.*
- ***AI:*** *Amazon Bedrock provides AI-powered features, including the Embedding Model and LLM Model.*
- ***Data Tier:*** *Amazon RDS for PostgreSQL serves relational data, DynamoDB provides NoSQL data storage, and Amazon S3 is used for general data storage.*
- ***Security & Monitoring:*** *IAM handles identity and access management. CloudWatch provides monitoring.* ***AWS X-ray*** *will be used for application tracing and performance analysis.*
- ***CI/CD:*** *GitLab handles version control, and AWS CodeBuild and AWS CodePipeline manage automated builds and continuous deployment.*



2.2 Technical Plan

The [Partner] will develop scripts using **AWS CodeBuild and AWS CodePipeline**. This approach allows for quick and repeatable deployments into AWS accounts. Database modeling will be performed using **MySQL Workbench**. Other tools utilized include various AWS services (S3, Lambda, API Gateway, DynamoDB, Bedrock, OpenSearch, and **AWS X-ray**) and **GitLab**.

2.3 Project Plan

[Partner] will adopt the **Agile Scrum framework**. The development spans three months (approximately 12 weeks), with phases focused on AWS Learning, Development, Integration, and Deployment. Stakeholders are required to participate in **Sprint Reviews and Retrospect**.

2.4 Security Considerations

Security considerations will cover 5 categories, including best practices:

*1. **Access:** [Partner] will implement AWS security best practices such as enabling **MFA on account access**. **IAM** will be used to manage access.*

*2. **Infrastructure:** (Utilizing managed serverless architecture).*

*3. **Data:** Data is secured via storage in RDS, DynamoDB, and S3.*

*4. **Detection: AWS CloudTrail** and **AWS Config** will be configured for **continuous monitoring of activities and compliance status of resources**. **CloudWatch** is also used for monitoring. **AWS X-ray** will be implemented for tracing request flows across services.*

*5. **Incident management:** [Customer] will share their regulatory control validation as inputs for [Partner] to ensure all security objectives are met.*

3 Activities AND Deliverables

3.1 Activities and deliverables

Project Phase	Timeline	Activities	Deliverables/Milestones	Total man-day
Assessment / AWS Learning	4 weeks (Month 1)	• Master AWS fundamentals (S3, Lambda, API Gateway, DynamoDB) • Learn advanced services (Bedrock, OpenSearch).	• AWS fundamentals mastered • Architecture designed • Database schema created.	[X man-day]
Setup base infrastructure / Development	4 weeks (Month 2)	• Build frontend UI; • Connect AWS backend.	• Frontend UI completed; • Lambda functions built; • API Gateway configured.	[X man-day]
Setup component 1 / Integration	4 weeks (Month 3)	● Complete system integration.	● Full system integration; ● Testing completed.	[X man-day]
Testing & Golive	4 weeks (Month 3)	● Final testing and production release.	● Production deployment.	[X man-day]
Handover	Week x-y	● item 1	● item 1	[X man-day]

3.2 OUT OF SCOPE

• Native mobile application development (initial scope is limited to the web platform).

• Implementation of AI features and community expansion beyond basic chatbot and recommendation foundations.

• Hardware costs, as Flyora is designed as a web-only platform.

3.3 PATH TO PRODUCTION

The POC/project will be built with specific and targeted use cases. It will lack core features which are required to be deployed into production. Production setup will require further fine-tuning to optimize all aspects of operational excellence.

4 EXPECTED AWS COST BREAKDOWN BY SERVICES

Reference to the aws calculator: ***https://calculator.aws/#/estimate?id=b930f0fc432907df70029346869ed6423e58ff50***

Estimated monthly and annual costs for the proposed AWS services:

Item	Monthly Cost	Annual Cost	Detail Calculation
Amazon S3	$0.15	$1.8	- Storage: 1GB
AWS Lambda	$0.00	$0.00	- 10.000 request - 512 MB Ephemeral storage - 256 MB Memory - Duration: 150ms
Amazon API Gateway (REST API)	$0.04	$0.48	- 10.000 request
DynamoDB (on-demand capacity)	$0.00	$0.00	- Data storage size: 0.01 GB - Number of writes: 0.01 million - Number of reads: 0.02 million
X-ray	$0.01	$0.12	- 10.000 request - Sampling rate: 10% - Traces retrieved per query: 20
CloudWatch & Logs	$0.00	$0.00	
Amazon Bedrock (Embedding/LLM)	$3.49	$41.88	- Cohere Embed Multilingual (83%), Claude 3 Haiku (17%) - 3.000 request
Amazon RDS for PostgreSQL	$21.01	$252.12	- db.t4g.micro - Storage: 20GB
Data transfer	$0.00	$0.00	- Free tier: 1 GB

CloudFront	$0.10	$1.2	- 10.000 request - Data Transfer Out: Free tier 1 GB (global)
CodePipeline	$0.00	$0.00	- 1 pipeline
CodeBuild	$2.52	$30.24	- arm1.2xlarge - 14 builds in a month - Average build duration: 2 minutes
VPC	$43.07	$516.84	- Hourly Charge: 24h - Data Processing: 3.000 request - 1 Nat gateway
Total Estimate	**$70.39**	**$844.68**	

[Call out assumptions made to create cost estimation] Pricing includes all AWS Services expected to be deployed based on the architecture listed above, including tooling costs (CodePipeline/CodeBuild). Hardware costs are not applicable as Flyora is a web-only platform.

5 TEAM

Partner Project Team

Name	Title	Role	Email / Contact Info
Trịnh Quốc Bảo	AI developer	Integrating chatbot with AWS	baotqse182782@fpt.edu.vn
Bùi Hồ Ngọc Hân	AI developer	Finetuning AI chatbot	buihongochan.lodi@gmail.com
Đỗ Phúc Duy	Backend developer	Integrating CI/CD with CodeBuild and CodePipeline	doychannel1802@gmail.com
Lưu Vĩ Khánh	Backend developer	Integrating API Gateway with Spring project	karlpro812005@gmail.com
Nguyễn Thanh Tân	Backend developer	Main backend architecture	captainparrot13042005@gmail.com
Trần Quang Hiếu	Frontend developer	Integrating React project with AWS	cubul435@gmail.com

Phạm Thành Phúc	Frontend developer	Responsible for integrating React application with AWS API Gateway	phamthanhqb2005@gmail.com

6 ACCEPTANCE

Upon completion of each Project Phase or Deliverable, the Partner ("PROVIDER") will submit the corresponding Deliverables to the Customer together with an Acceptance Form as defined in Appendix B of this SOW.

The Customer will have eight (8) business days ("Acceptance Period") to review, verify, or test the submitted Deliverables to confirm whether they meet the acceptance criteria defined for the project.

- If Deliverables meet all acceptance criteria:
  The Customer will provide written confirmation of acceptance using the Acceptance Form before the end of the Acceptance Period.

- If Deliverables do not meet acceptance criteria:
  The Customer will issue a Rejection Notice, specifying all defects, non-conformities, or missing requirements. PROVIDER must then correct these issues and resubmit the updated Deliverables along with a new Acceptance Form.

- Upon resubmission:
  The Customer will limit their review only to:
  (1) previously reported defects or non-conformities, and
  (2) any new issues directly caused by the fixes.
  The acceptance flow described above will be repeated.

- No response within the Acceptance Period:
  If the Customer does not provide an Acceptance Form or Rejection Notice within the Acceptance Period, the Deliverables will be considered automatically accepted ("deemed accepted").

This acceptance process applies to all phases, milestones, and final project handover.