AWS First Cloud AI Journey – Dai Viet Blood Donation & Emergency System (DaiVietBlood)

Implemented by: Skyline Team – FPT University Ho Chi Minh City
Date: December 7, 2025

TABLE OF CONTENTS

---

1. BACKGROUND AND MOTIVATION

1.1 EXECUTIVE SUMMARY

Customer Background:
The DaiVietBlood system is designed to serve the community, including voluntary blood donors, patients in need of emergency blood, and healthcare professionals in Vietnam. The primary customers are blood donors, patient families, and medical staff responsible for managing blood inventory and donation schedules. They require a centralized, reliable platform to optimize the matching process between donors and recipients and improve

communication during emergencies. In the context of digital health transformation, DaiVietBlood provides a secure, accessible solution to address localized blood shortages.

Business and Technical Objectives:
Migrating the DaiVietBlood system from a local/on-premise environment to AWS offers superior advantages:

- **Business:** AWS allows the application to scale flexibly according to the user base, reduces hardware infrastructure operational costs, and ensures consistent performance nationwide.
- **Technical:** AWS provides High Availability and medical data security. Adopting a **Serverless Architecture** (AWS Lambda, API Gateway, Cognito, RDS) simplifies backend management, accelerates development, and reduces maintenance costs. The system integrates comprehensive monitoring (CloudWatch) and adheres to strict security standards.

Summary of Key Use Cases:

Role	Key Function	Short Description
Guest	Access Public Information	View donation guidelines, blood compatibility charts, and educational articles without logging in.
Member	Register/Login, Profile Management	Create accounts, update personal information and blood type.
	Book Blood Donation	Select time slots and locations for donation.
	Submit Emergency Request	Submit urgent blood requests; the system automatically finds suitable donors.
Staff	Manage Requests & Inventory	Approve emergency requests, confirm donation schedules, update blood stock.
Administrator (Admin)	System Administration	Manage user accounts, configure donation slots, view overview reports.

Summary of Partner's Professional Services:
The Skyline Team will provide comprehensive digital transformation services, including assessing the current local application, designing a Cloud-native architecture, and executing the migration of the system to an AWS Serverless environment. We commit to delivering a secure, scalable system accompanied by automated CI/CD pipelines and detailed operational documentation.

1.2 PROJECT SUCCESS CRITERIA

1. **Functionality:** 100% of core functions (registration, scheduling, emergency requests, administration) operate stably on AWS with no regression errors.

2. **Availability:** System achieves Uptime ≥ 99.9%, ensuring continuous 24/7 access.
3. **Performance:** Application response time improves by at least 30% compared to the local version. Emergency request processing time is reduced by 40%.
4. **Cost:** Infrastructure costs are optimized by at least 20% thanks to the Serverless model and Auto-scaling.
5. **User Experience:** UAT acceptance rate reaches a minimum of 95% for all user roles.
6. **Security:** Full compliance with data encryption, access management (IAM), and API security requirements.
7. **Operations:** CI/CD pipeline is fully automated with deployment time < 10 minutes. Monitoring system covers 100% of critical services.

1.3 ASSUMPTIONS

Technical & Architectural Assumptions:

- **Source Code:** The current Local application (Frontend & Backend) is functionally complete. The project focuses on Refactoring for the Cloud (Serverless), excluding the development of new features.
- **AWS Region:** The entire infrastructure is deployed in **Singapore (ap-southeast-1)** to optimize latency for users in Vietnam. *Note: During the testing phase, due to limited VPC configurations and Free Tier/Student resources, latency may fluctuate (estimated ~3.5s/request).*
- **Service Limits:** The AWS account uses default limits (Soft limits). Increasing limits to reduce latency will be approved by the Customer when necessary.
- **Third-party Integration:** The system uses the Gemini API for AI support features.
- **Access Rights:** The Skyline Team is granted Admin access (IAM Role) to provision resources.

Operational & Financial Assumptions:

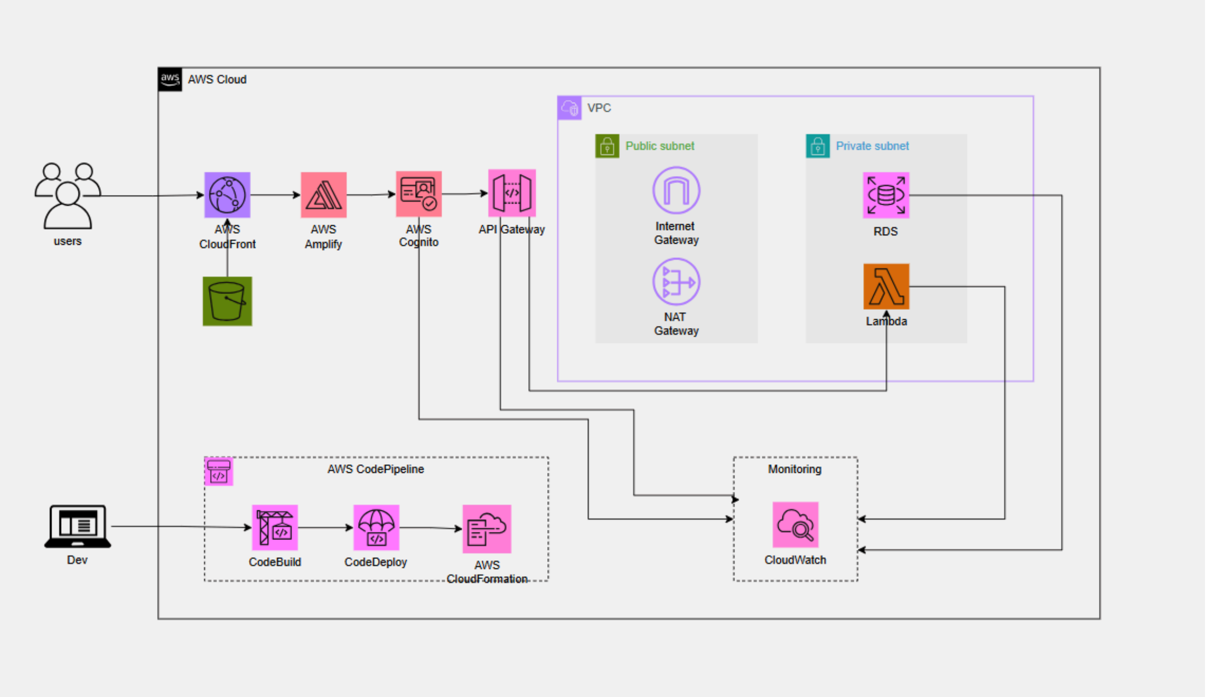
- **Domain:** The Customer owns the domain name (e.g., daivietblood.com) and DNS configuration rights.
- **Cost:** The cost estimate is based on an assumption of approximately 50,000 API requests/month. Actual costs depend on usage levels (Pay-as-you-go).

---

2. SOLUTION ARCHITECTURE / ARCHITECTURAL DIAGRAM

2.1 TECHNICAL ARCHITECTURE DIAGRAM

The DaiVietBlood system utilizes a **Serverless-First** architecture on AWS Cloud, prioritizing scalability, security, and operational optimization.



Key Components:

1. **Network Infrastructure (VPC):**
   - *Public Subnet:* Contains Internet Gateway and NAT Gateway.
   - *Private Subnet:* Contains AWS Lambda and Amazon RDS to isolate and secure data, preventing direct Internet access.
2. **Application & Data:**
   - *Frontend:* Hosted on **AWS Amplify**, distributed via **Amazon CloudFront** (CDN), and assets stored on **S3**.
   - *Authentication:* **Amazon Cognito** manages identity and issues JWT tokens.
   - *API & Compute:* **Amazon API Gateway** receives requests and routes them to **AWS Lambda** for business logic processing.
   - *Database:* **Amazon RDS** stores structured data, located in the Private Subnet.
3. **DevOps & Monitoring:**
   - *CI/CD:* Uses **AWS CodePipeline, CodeBuild, CodeDeploy** to automate the deployment process.
   - *Monitoring:* **Amazon CloudWatch** centrally collects logs and metrics.

2.2 TECHNICAL PLAN

The technical implementation process follows the Infrastructure-as-Code (IaC) methodology:

- **Infrastructure Automation:** Use **AWS CloudFormation** to provision VPC, Lambda, RDS, and API Gateway, ensuring consistency across environments (Dev/Staging/Prod).
- **Application Development:** Refactor backend into modular Lambda functions (NodeJS/Python). Environment variables and sensitive information (DB credentials) are securely encrypted.

- **CI/CD Process:**
    - Source (GitHub) -> Build (CodeBuild) -> Deploy (CloudFormation/CodeDeploy).
    - Includes a Manual Approval step before deploying to the Production environment.
- **Testing Strategy:** Unit Tests for Lambda, Integration Tests for API, and Load Tests to ensure capacity.

2.3 PROJECT PLAN

The project applies the Agile Scrum model over 8 weeks (4 Sprints):

- **Sprint 1 (Foundation):** Set up AWS Account, VPC, RDS.
- **Sprint 2 (Backend Core):** Develop Lambda, API Gateway, Cognito.
- **Sprint 3 (Integration):** Deploy Frontend (Amplify), finalize CI/CD Pipeline.
- **Sprint 4 (Stabilization):** UAT, Performance Optimization, Handover.

2.4 SECURITY CONSIDERATIONS

1. **Access Management:** Use Cognito for user authentication and IAM Roles for service authorization (Least Privilege).
2. **Network Isolation:** Database and Lambda are located in the Private Subnet, accessing the Internet only via NAT Gateway.
3. **Data Protection:** Data encryption At-rest (on RDS/S3) and In-transit (via HTTPS).
4. **Security Monitoring:** CloudWatch Logs record all activities for auditing and intrusion detection.

---

3. ACTIVITIES AND DELIVERABLES

3.1 ACTIVITIES AND DELIVERABLES

Phase	Timeline	Key Activities	Deliverables	Estimate (Man-days)
Analysis & Design	Week 1	Assess Local state, design Cloud architecture, plan migration.	SRS Document, Architecture Diagram, API Specs.	5
Local Development	Week 2-3	Build backend logic, database schema, local unit tests.	Backend Prototype, Database Schema.	10

Frontend & Integration	Week 4-5	Develop Frontend, integrate local APIs, prepare code for refactoring.	Completed Local Application.	10
AWS Infrastructure Setup	Week 6	Write CloudFormation scripts, provision VPC, RDS, IAM.	IaC Templates, Secure VPC Environment.	5
Refactor & Deploy Backend	Week 7-8	Convert to Lambda, configure API Gateway, Cognito.	Serverless Backend active on AWS.	10
Deploy Frontend & CI/CD	Week 9-10	Host Frontend on Amplify, set up automated Pipeline.	Production URL, CI/CD Pipeline.	10
Testing & Go-live	Week 11	UAT, Security Testing, Performance Optimization.	UAT Report, Security Report.	5
Handover & Training	Week 12	Transfer accounts, operations training, handover documentation.	Operations Manual, Acceptance.	5

3.2 OUT OF SCOPE (MVP PHASE)

Due to time and resource limitations of the MVP phase, the following items are not included:

1. Optimal user search algorithm based on real-time Geo-location (currently using simplified logic).
2. Complex Auto-scaling for the Database layer (currently using basic RDS).
3. Deep Latency Optimization for regions outside Singapore.
4. Advanced Security Compliance standards such as HIPAA/PCI-DSS.

3.3 PATH TO PRODUCTION

To upgrade from the current MVP to a large-scale Production system, the following are required:

1. **Environment Strategy:** Strictly separate Dev/Staging/Prod environments across different AWS accounts (Multi-account strategy).
2. **Database Scaling:** Migrate to Amazon Aurora Serverless or use Read Replicas to increase read/write capacity.
3. **Enhanced Monitoring:** Integrate AWS X-Ray to trace requests and identify performance bottlenecks.
4. **Strengthened Security:** Deploy AWS WAF with rules to block DDoS and automated bots; use Amazon Inspector for periodic vulnerability scanning.

4. EXPECTED AWS COST BREAKDOWN

Region: Asia Pacific (Singapore)

Category	Service	Estimated Configuration	Monthly Cost (USD)
Network	NAT Gateway	1 NAT Gateway (Required for Private Subnet) + Data Processing	~$43.13
	VPC	Subnets, Security Groups	~$13.14
	CloudFront	5GB Data Transfer (Utilizing Free Tier)	~$3.00
Compute	Lambda	1,000 requests, 512MB RAM (Free Tier)	~$0.00
	API Gateway	1,000 requests	~$0.00
Database	RDS	db.t3.micro, 20GB Storage	~$21.74
Storage	S3	5GB Storage, 200 requests	~$0.14
Hosting	Amplify	Build & Hosting, WAF enabled	~$16.77
Ops	CloudWatch	Logs, Metrics, Alarms	~$9.41
CI/CD	CodePipeline	1 Active Pipeline	~$1.05
Total			**~$108.38 / Month**

5. IMPLEMENTATION TEAM

- **Mentor (AWS FCJ):** Nguyen Gia Hung - Head of Solutions Architect.
- **Project Manager (PM):** Nguyen Duc Lan - Coordination, schedule management, cost optimization, and UAT strategy.
- **Technical Lead:** Nguyen Cong Minh - In charge of CI/CD, Infrastructure (CDK), Security, and Lambda.
- **Solution Architect:** Do Khang - Serverless architecture design, AI Chatbot integration, Service Policies.

- **Fullstack Developer:** Le Hoang Anh - API development, Frontend UI/UX, and application security.
- **Data Engineer:** Nguyen Quach Lam Giang - RDS administration, VPC/Subnet design, and CloudWatch monitoring.

---

6. RESOURCES & ESTIMATED PERSONNEL COSTS

Time Allocation (Man-hours):
The project mobilizes a total of **750 man-hours** distributed equally among 5 members across phases: Foundation, Core Development, Data Analysis, Testing, and Handover.

Cost Allocation:

- **Personnel:** $0 (Performed by students as part of an internship/capstone project, counted towards academic credits).
- **AWS Infrastructure:** ~$15 (Actual costs incurred during dev/test after deducting Credits).
- **Total Project Cost:** Highly optimized, relying mainly on internal resources and support from the AWS FCJ program.

---

7. ACCEPTANCE

7.1 Submission of Deliverables:

Upon completion of the "Handover" phase, the Skyline Team will submit all Source Code, Architecture Documentation, Admin Accounts, and Operations Manuals to the Customer/Mentor.

7.2 Acceptance Period & Process:

The Customer has 05 business days to review and perform UAT. If the product meets the Success Criteria (Section 1.2), the Customer will sign the acceptance confirmation.

7.3 Defect Remediation:

If critical errors arise or features are missing compared to the committed scope, the Skyline Team is responsible for fixing and resubmitting for acceptance as soon as possible.