Nguyen Huu Bao Duy

CloudDevOps Engineer · Fresher

I am a final-year Network and Systems student at PTIT. I am developing practical experience in deploying and operating cloud services and automating system workflows.

duybao09012004@gmail.com

083 418 4916

GitHub: https://github.com/Crozn1812 · LinkedIn: https://www.linkedin.com/in/nguyenhuubaoduy

Experience

Cloud Engineer Intern · HPT Corporation

07/2026 – present

Contributed to Microsoft Cloud deployments at HPT Corporation. On the partner data sharing project, I worked with Azure API Management to manage API access and Azure Functions to connect, normalize and filter data from cloud and on-premises sources. I also worked with Microsoft Fabric for data workflows, Power Automate for workflow automation and Microsoft Foundry while exploring and deploying AI services.

Education

Posts and Telecommunications Institute of Technology (PTIT)

Network and Systems Engineer

Expected graduation: 2027 · in progress

Skills

Cloud Platform · AWS

- EC2
- S3 — sales CSV processing pipeline
- Lambda — CSV processing triggered by S3 events
- DynamoDB
- CloudWatch
- AWS CDK
- EKS — CMS capstone
- ECR — CMS capstone
- ALB — CMS capstone
- RDS PostgreSQL — CMS capstone

Cloud Platform · Azure

- Azure API Management — Partner Data Gateway
- Azure Functions — Partner Data Gateway
- Microsoft Entra ID — Partner Data Gateway
- Application Insights — Partner Data Gateway
- Microsoft Foundry — internship
- Microsoft Fabric — internship
- Power Automate — internship

Languages

- Python — data processing and backend services
- Shell — deployment environments
- C++ — coursework

Network and Systems

- Linux — deployment environments
- CCNA networking fundamentals
- Cisco Packet Tracer — labs
- pfSense — study

Tools

- Terraform
- Docker
- Kubernetes
- Helm
- Jenkins
- GitHub Actions
- Argo CD
- Prometheus — observability
- Grafana — observability
- Postman — API testing

Certificates & learning

CCNA

Cisco Certified Network Associate, earned 03/2026.

Certificate of Completion: DevOps, VTI Academy

Completed the DevOps training course, 09/2026. Its graduation project was the CMS DevOps Platform on AWS EKS.

Planned: AWS Solutions Architect Associate

Not taken yet. This stone is still blank.

Planned: a Linux certification or CEH

Not taken yet. This stone is still blank.

Projects

CMS DevOps Platform AWS EKS

A platform that builds and automates delivery of a content management system (CMS) on Amazon EKS, combining Terraform infrastructure as code, Jenkins CI, Argo CD GitOps and monitoring.

Status: completed

Role: Sole author of the graduation project (platform design and delivery pipeline).

Contributions: Designed the whole chain described under architecture alone, from infrastructure as code to monitoring (per the slides).

Results: Self-reported on the slides: Terraform/VPC/EKS with 2 worker nodes; private Multi-AZ RDS plus Redis; Jenkins, ECR and Argo CD GitOps; private by default with a Pod security baseline; rolling update, probes and rollback; CloudWatch, Prometheus and Grafana. Independent evidence (screenshots, logs, repository): none yet. Metrics: not measured.

Secure Partner Data Sharing Portal

A secure way to share approved data from two main data flows, cloud and on-premises, with external partners through Azure API Management and Azure Functions: each partner has its own subscription key, and an Azure Function returns only the records and fields that partner may see. Built at HPT in two phases; both are complete.

Status: completed

Role: Cloud & DevOps intern at HPT; presenter of the phase 1 proof of concept.

Contributions: Created an individual APIM subscription and key per partner; developed an Azure Function that reads the approved data and filters it by partner ID; implemented rate limiting, correlation IDs, key rotation and subscription revocation; tested authentication, error handling and isolation between partners (per Duy's CV). Phase 2, with a Kong gateway and an on-premises data service, was completed as well.

Results: Phase 1 worked end to end: the endpoint returned 200 and the test partner received exactly the 2 records assigned to it. Phase 2 (Kong, on-premises) is complete according to Duy. Latency, volume, cost and adoption were not measured. The public repository is a demo with mock data, not the company system.

Repository: https://github.com/Crozn1812/azure-api-gateway-dsr-ai · demo

InsightHub RAG

InsightHub is a local RAG notebook: upload .txt, .md or .pdf files, ingest them asynchronously, ask questions and review citations. Duy wrote the whole implementation, including the DevOps, security, observability and cost-tracking layers, on a repository supplied by the instructor.

Status: completed

Role: Wrote the whole implementation on the instructor's repository.

Contributions: The web, API and ingestion worker; the PostgreSQL plus pgvector data model; Terraform and Helm, GitHub Actions and Checkov for the local path; guardrails on input, context and output; Prometheus and Grafana monitoring; a chat-ops bot; per-workflow budget keys.

Results: A real incident was fixed: embedding requests failed with HTTP 400 because chunks exceeded the model's 512-token context. Reducing the conservative chunk budget from 800 to 400 let three failed documents be reprocessed to ready (124 chunks each) without losing IDs or payload, with the same model and data. The local runtime works; the AWS runtime was not executed. Answer quality: not measured. Local model cost is zero.

Repository: https://github.com/Crozn1812/insighthub_devops

AWS Sales Data Pipeline

An event-driven AWS pipeline that validates, cleans and transforms retail sales CSV files uploaded to S3, separates bad records, and tracks file metadata in DynamoDB; infrastructure is defined with AWS CDK.

Status: completed

Role: Author of a personal project.

Contributions: Used AWS CDK to deploy S3, Lambda and DynamoDB; configured an S3 event source so that new objects in the raw bucket invoke the Lambda function directly; wrote Python and Pandas scripts to clean, transform and partition CSV data; wrote Pytest tests for the data logic (per Duy's CV and the README).

Results: An example run on the sample file produced 99 clean records and 4 error records with status COMPLETED (sample data, not a performance metric). Bad records are separated into the error bucket and logs are available in CloudWatch. Throughput, latency and cost were not measured.

Repository: https://github.com/Crozn1812/AWS-Sales-Data-Pipeline