Le Nguyen Phuc Loc

JUNIOR DEVOPS ENGINNEER

0356 413 729 phuclocdh2017@gmail.com 28/01/2002 District 7, Ho Chi Minh City

locdevops.cloud linkedin.com/in/locledevopsengineer

OBJECTIVE

As a Software Engineering graduate with over a year of hands-on experience as a DevOps Engineer, I am passionate about automating processes, optimizing cloud architectures, and improving system reliability. My goal is to leverage my expertise in cloud technologies and CI/CD practices to contribute to innovative and dynamic teams. I am eager to continue learning and advancing my skills in the DevOps field, ultimately growing into a well-rounded professional who drives efficiency and system performance at scale.

EDUCATION

Sai Gon University 2020 - 2024

Major: Software Engineering

During my studies, I have mastered programming languages such as Python, JavaScript. I also learned about Linux system administration, computer networks. In addition, I participated in development group projects in many languages: Nodejs, Nextjs, Python, operated by Docker, Kubenetes, AWS.

FUNIX 2023 - 2024

Major: Cloud Computing

Here, I was equipped with knowledge and skills about AWS cloud services, I mastered how to deploy, manage and optimize applications on the AWS platform, thereby improving my practical ability and readiness to apply knowledge into real work in the field of software engineering and DevOps.

WORK EXPERIENCE

ATOM Solution – Payment Solution Platform Jun 2024 - Apr 2025

Junior Devops Engineer

- Contributed to key production projects, including Atom Processor x BVBank, Merchant Platform x OCB, ViTrust, and WESAP, helping to build and operate efficient and secure Payment systems.
- Gained a deep understanding of Payment flows, Core Card, and Core Bank systems, improving the operation and efficiency of Payment systems.
- Possess knowledge of PCI-DSS compliance and actively participated in the Development and Maintenance of secure Payment system Infrastructure.
- Led a campaign to Optimize the company's cloud resources, reducing infrastructure costs by 40% while maintaining performance and reliability.
- Built and deployed CI/CD pipelines, managing scalable Microservices on Kubernetes for Payment, ERP, and Web3 systems.
- Migrated Production projects from AWS to CMC and FPT Cloud environments, enhancing scalability and system performance.
- Implemented and managed high availability setups for PostgreSQL (Master-Slave), Redis (Cluster), MongoDB (Replica), Kafka (Cluster), EMQX (Cluster), and Minio (Replica), ensuring fault tolerance and system resilience.
- Automated system monitoring and logging using Grafana Stack (Grafana, Prometheus, Loki, Promtail) and Elastic Stack (Opeansearch, Logstash, Filebeat, Metricbeat) with Ansible, improving incident detection and system reliability.

- Developed and Maintained automation scripts using Bash to alert, backup, sync data, and automate routine tasks, saving time and reducing errors.

TECHNOLOGIES

English	Reading and writting well, Listening and speaking specialized basic. Capable of conducting research and reading technical documentation in English for work purposes.
Programming language	Python, Bash Script, C++, Javascript.
Payment Systems	Experienced with Core Card systems, Core Banking, and Payment system architecture.
CI/CD & GitOps	Jenkins, Gitlab CI, AWS CodePipeline, Kaniko CI, FluxCD.
Container Orchestration	Kubernetes, Docker swarm, Docker compose, AWS ECS.
Security	Knowledge of PCI-DSS Infrastructure construction, Vault, SonarQube, Trivy.
Networking Protocols	OSI Model, DNS, HTTP/HTTPS, SSH, SSL/TLS.
Architecture	Microservice, Serverless
Infrastructure	Postgres, Redis, MongoDB, Kafka, EMQX, Minio.
Monitoring & Logging	Grafana Stack (Grafana, Prometheus, Loki, Promtail) and Elastic Stack (Opeansearch, Logstash, Filebeat, Metricbeat).
Infrastructure as code	Terraform, Ansible, AWS CloudFormation.
Load Balancing	Nginx, HaProxy, Elastic Load Balancing (ELB)
Cloud computing	Familiar with AWS platform services, and Cloud Computing Architectures: IaaS, PaaS, SaaS, FaaS.
Soft skills	Creative thinking, problem solving and independent work, initiative and good teamwork ability.
Work spirit	Listening and learning, progressive spirit, willingness to learn new technology, honesty, responsibility for common work.

PROJECT

Building and monitoring automated CI/CD pipeline for Microservices with AWS Copilot

(02/2024 - 04/2024)

Project Type	Personal Project
Project Description	In this project, I implemented a task management application deployed as applications using AWS Copilot CLI to create CI/CD pipelines for both frontend and backend. The CI/CD pipelines utilized services like AWS CodePipeline, CodeCommit, CloudFormation, and Amazon ECS to automate the process from code commit to deployment of container services.
Project Repository	https://github.com/phuclocdh/aws-cicd-copilot.git
Project Design	https://github.com/phuclocdh/aws-cicd-copilot/blob/main/Readme.md
Project Demo	https://github.com/phuclocdh/aws-cicd-copilot/blob/main/aws_copilot_mov.zip

Technologies	• AWS Copilot CLI: Used for Infrastructure as Code (IaC) to define and manage the application's infrastructure, including environments, ECS services, and CI/CD pipelines. • AWS CodePipeline: Automated the build, test, and deployment stages whenever there was a code change. Integrated with CodeCommit for source control. • AWS CloudFormation: Defined and managed the application's infrastructure. • AWS CodeCommit: Hosted the Git repositories for the application's source code. • Amazon ECS (Elastic Container Service): Deployed and managed containerized applications, running, stopping, and managing containers on a cluster of virtual servers. • Docker: Containerized application services for consistent and isolated execution environments. • AWS DynamoDB: NoSQL database service used to store todo tasks created by the backend service. • AWS CloudWatch: Monitored the health and performance of deployed ECS clusters and services. • IAM (Identity and Access Management): Managed permissions and access for AWS services used in the project. • Amazon VPC (Virtual Private Cloud): Provisioned isolated network environments for application services. • AWS Systems Manager Parameter Store: Stored configuration and environment settings for deployed applications.

Developing and Deploying a scalable TicTacToe website using Serverless AWS Architecture

(12/2023 - 02/2024)

Project Type	Personal Project
Project Description	In this project, I use serverless architecture on AWS with services: Gateway API using WebSocket, DynamoDB, Lambda. In addition to online application on IPv4, I also use EC2, Nginx, technologies. Besides that, you can also use high availability through services: VPC, ALB, ASG..
Project Repository	https://github.com/phuclocdh/tic-tac-toe.git
Project Design	https://github.com/phuclocdh/tic-tac-toe/blob/main/Readme.md
Project Demo	https://github.com/phuclocdh/tic-tac-toe/blob/main/tictactoe.mov.zip
Technologies	• AWS Lambda: Handles functions corresponding to API Gateway routes such as connect, disconnect, send messages, join room, and get room. Receives data from the client and executes the corresponding function when called by the API Gateway. AWS DynamoDB: Serves as the main database for the web application with tables supporting Lambda functions and displaying information on the website. Implements Time to Live and Streams to monitor online and offline players. • AWS API Gateway: Utilizes WebSocket API to receive signals from the client and create routes for functions like send messages, get online status, and notify online status. Calls corresponding Lambda routes to perform functions and return results to the client. • AWS DynamoDB: Ensures high availability and scalability for data storage, supporting distributed and fault-tolerant architecture with its managed NoSQL capabilities.AWS Virtual. • AWS EC2: Hosts the source code for the online website on an IPv4 address using Amazon Linux OS. • AWS CloudWatch: Collects logs and metrics from Lambda, API Gateway. and DynamoDB. Dashboards visualize game traffic, active players, and system health. • AWS SNS: Notifies relevant functions, such as online and offline players, via email when records are deleted, based on checks performed by Lambda on the corresponding table. • NGINX: Acts as a web server and reverse proxy to convert local IP addresses on EC2 to IPv4, allowing users to access the website via HTTP/HTTPS.

	• Auto Scaling Group (ASG): Automatically scales the number of EC2 instances based on load and configuration thresholds, ensuring high availability and reliability for the application. • Application Load Balancer (ALB): Balances the load between EC2 instances in ASG, routing traffic according to preconfigured rules.

CERTIFICATIONS

Devops on AWS at CodeStar Academy	2023
Cloud Computing with AWS at Funix	2024

HONORS & AWARDS

English Olympic Consolation Prize at City Level	2013
Consolation Prize for City level Math Violympic	2014
City-level Young Informatics Consolation Prize	2018
City level Chess Gold medal	2011 - 2013 - 2015 - 2017