

# Felicia Ebikon

## DevOps Engineer

+2349031392428 [feliciaebikon@gmail.com](mailto:feliciaebikon@gmail.com) Portfolio: [Portfolio](#)

LinkedIn: <https://www.linkedin.com/in/felicia-ebikon> GitHub: <https://github.com/Pearlicia>

### SUMMARY

Experienced **DevOps engineer** with over 5 years of experience in cloud infrastructure management and automation. Proficient in **AWS** with hands-on experience in Terraform, Ansible, Jenkins, Github Actions, Docker, and Kubernetes. Skilled in optimizing release cycles and reducing infrastructure costs through automation and continuous delivery. I have experience working in agile environments and have strong collaboration and communication skills.

### PROFESSIONAL EXPERIENCE

#### DevOps Engineer

##### *Yebox Technologies, Lagos Nigeria*

October 2023 to August 2024

- Proficient in defining Ansible Master Server for managing and configuring nodes, as well as developing **Ansible Playbook** and Modules for **system configuration management**.
- Developed and scheduled **bash shell scripts** for various tasks, including environment verification, running database scripts, file manipulations, and version control with Subversion.
- Developed **Ansible playbooks** for the installation and auto-healing of various tools like **Jenkins, SonarQube, MySQL, and Nexus, PostgreSQL**, specifically designed for Linux platforms.
- Implemented and managed **Docker** and **Kubernetes** infrastructure, running **Jenkins** in Docker containers with EC2 slaves in AWS.
- Managed, maintained, and deployed to test, acceptance, and production environments. Designed **highly available, cost-effective, and fault-tolerant** systems using multiple EC2 instances.
- Defined dependencies and plugins in **Maven's pom.xml** for various activities, and integrated Maven with GIT for project tag management and deployment.
- Employed **Amazon IAM** to manage fine-grained access to AWS resources for users, and handled user **roles and permissions** through IAM.
- Leveraged **CloudFront** to deliver content from **AWS edge locations** to users, reducing the load on front-end servers. Experienced in automating the deployment of EC2 instances across data centers and availability zones.
- Implemented and maintained monitoring and alerting for production and corporate servers like EC2 and storage solutions like **S3 buckets** using **AWS CloudWatch**.
- Developed AWS CloudFormation templates to create custom-sized **VPCs, subnets, NAT Gateway, EC2 instances, ELB, and Security Groups**.
- Managed Linux virtual servers on **AWS EC2 using Ansible Server**, and monitored distributed and multi-platform servers with Ansible.
- Expert in designing and implementing solutions using AWS infrastructure services such as **EC2, S3, EBS, Elastic Load Balancer (ELB), VPC, Amazon RDS, CloudFormation**, and more.
- Experience with **scrum methodologies** and using the **Jira ticketing system** to manage tasks. Familiar with scrum ceremonies, backlog refinement, and sprint planning.
- Experience using Grafana and Prometheus for monitoring and visualization.

**Environment:** AWS, GIT, Jenkins, Nginx, SonarQube, Maven, Nexus, Tomcat, Docker, Kubernetes, AWS ECS, MySQL, YAML, Shell Scripts, XML, UNIX, Linux (CentOS), Ansible, PostgreSQL, Ubuntu, Prometheus, Grafana.

#### DevOps Engineer

##### *Sellani, Port-Harcourt Nigeria*

January 2023 to August 2023

- Successfully migrated legacy systems to modern cloud-based architecture, **resulting in improved performance, reliability, and reduced maintenance overhead**.

- Conducted internal training sessions to **educate team members on new technologies, tools, and best practices.**
- Implemented Git workflows and continuous integration/continuous deployment (CI/CD) pipelines, **enabling faster code iterations and delivery.**
- Implemented automated deployment pipelines using tools like Jenkins and Github Actions CI/CD, **resulting in faster and more reliable software releases.**
- Developed infrastructure using IaC tools like Terraform and CloudFormation, **leading to consistent and reproducible environments.**

## DevOps Engineer

*Unideals, UK. Remote*

November 2022 to February 2023

- Created comprehensive documentation, guides, and tutorials to facilitate **knowledge sharing among team members.**
- Established efficient collaboration between development and operations teams, **fostering a culture of transparency and continuous improvement.**
- Tuned database performance through indexing, query optimization, and caching, **resulting in reduced latency and improved response times.**
- Optimized cloud infrastructure costs by **rightsizing resources**, implementing reserved instances, and utilizing auto-scaling effectively.
- Set up robust monitoring using tools like **Prometheus and Grafana**, enabling proactive issue detection and resolution.

## Full Stack Developer

*Speednet LTD, Abuja Nigeria. Remote*

June 2022 to September 2022

- Implemented coding standards, code reviews, and automated testing, **resulting in a 30% reduction in bugs and faster feature delivery.**
- Improved database performance by optimizing queries, implementing indexing strategies, and **reducing query execution times by 40%.**
- Created well-documented and robust APIs that facilitated seamless integration between various frontend and third-party systems, **enhancing the overall user experience.**
- Designed and developed a scalable backend architecture that allowed the application to handle a **3x increase in user traffic** without degradation in performance.
- Successfully transitioned a monolithic application to a microservices architecture, leading to **better maintainability, scalability, and more focused development teams.**

## Full Stack Developer

*Toroma Innovation Hub, Yenagoa, Nigeria*

September 2018 to January 2020

- Mentored junior developers, providing guidance on **best practices, design patterns, and coding standards, resulting in improved team productivity and skill development.**
- Collaborated closely with frontend developers, designers, and product managers to ensure **alignment and smooth feature development cycles.**
- Increased test coverage by implementing unit tests, integration tests, and end-to-end tests, **resulting in more reliable code and fewer post-release issues.**
- Created comprehensive API documentation using tools like Swagger, making it **easier for external developers to integrate** with the platform.
- Analyzed and optimized the backend's performance bottlenecks, **resulting in a 30% reduction in API response times** and improved user experience.

## SKILLS

- **Programming Languages:** Python, Javascript, Ruby, SQL
- **Backend:** Flask, Nodejs, Ruby on Rails
- **Tools:** Ansible, Jenkins, Terraform, Sonarqube, Bash, CircleCI, TravisCI, Github Actions, Honeycomb,

- Rollbar, Vagrant, VirtualBox, Docker, Kubernetes, Maven, Nexus, Nginx, Prometheus, Grafana.
- **CLOUD:** AWS Cloudformation, Lambda, Load Balancer, CodeDeploy, CodePipeline, SAM, Cognito, EKS, ECS, Azure, GCP
- **Databases:** PostgreSQL, MySQL, DynamoDB, MongoDB, Firestore, Redis, Momento
- **Testing:** Pytest, Rspec, Selenium, Mocha, Jest, Junit
- **Soft Skills:** Ability to learn fast and google well, Ability to pick up new things quickly, Continuous learning, Problem Solving, Communication, Teamwork, Adaptability
- **Microsoft:** Word, Excel, Powerpoint, 365

## TECHNICAL PORTFOLIO

### TERRATOWNS

[Github](#)

TerraTowns is a community website that acts as a hub to discover and connect terraformers to each other's self-hosted personal websites in the style of Geocities of 2023.

I wrote the infrastructure as Code (IaC) to launch my Terra House. A Terra House is a simple Content Management System (CMS) that allows me to author my own personal website and connect it to the TerraTowns network. I chose my **topic of interest** and built my page around existing community hubs.

- **Tech Stack:** Terraform, Ruby, Golang, HCL, S3, AWS Cloudfront, HTML, CSS, Bash, GIT

### CRUDDUR

[Github](#)

A micro-blogging platform that emphasizes privacy and the present moment. Cruddur platform allows users to post updates, thoughts, and photos that automatically expire after a period of time, ensuring that your personal information and conversations stay relevant and in the moment. Perfect for busy professionals, students and anyone who wants to stay connected without the pressure of maintaining a permanent online presence.

- **Tech Stack:** React, Python, AWS lambda, DynamoDB, postgres, Cognito, ECS, Cloudformation, codePipeline, codeBuild, x-ray, rollbar, momento, route53, honeycomb

### MICROSERVICES SECURITY HARDENING AND INCIDENT REPORT

[Github](#)

I was presented with the challenge to build a secure Microservice environment, threat modeling and hardening the container image, run-time environment and application itself. For purposes of the project, I was instructed to use a secure base opensuse image, covering considerations for the importance of using trustworthy base images and verifying the baseline. I was provided with instructions to build, harden, ship and run an environment analogous to the company's new microservice application, simplified for project purposes. In the project I defined and built a new environment from the ground-up. In a real-world scenario, I may have an existing environment that needs to be hardened or may decide to rebuild parts or all net-new, regardless, the tools and techniques in the project are directly applicable. The beauty of microservices vs a monolith architecture is that all core components (image, container, run-time, application) are abstracted allowing for isolation boundaries and iterative development. In the real-world, I could choose to harden and redeploy all base-images as one project phase and tackle docker container security, kubernetes hardening and the software composition analysis, as individual project phases. The best approach is to bake these requirements and security hardening into the build and deploy process. In an enterprise setting, much of this can be enforced with security units tested via CI/CD prior to deployment. Hardening the base image and baking security into the CI/CD is beyond the scope of this project. For this project, once the Microservice environment is hardened and provisioned, I configured sysdig Falco to perform run-time monitoring on the node, sending logs to a Grafana node for visualization. To demonstrate to the CTO that the company can respond to a real security event, I simulated a tabletop cyber

exercise by running a script to introduce an unknown binary from the starter code that disrupts the environment! My goal was to evaluate Grafana to determine what the unknown binary is, contain and remediate the environment, write an incident response report, and present it to the CTO.

- **Tech Stack:** Flask, Python, Sysdig Falco, Grafana, Kubernetes, Vagrant, K3s, Docker, Bash

## TECHTRENDS

[Github](#)

This project demonstrates a streamlined and automated approach to deploying containerized flask application using a combination of GitHub Actions, k3s, and ArgoCD. It provides a complete DevOps workflow for building, pushing, and deploying Docker images to a Kubernetes cluster.

- **Tech Stack:** GitHub Actions, Docker, K3s, ArgoCD, Helm, Flask, Vagrant, Kubernetes

## UDAGRAM IMAGE FILTERING APPLICATION

[Github](#)

Udagram is a simple cloud application developed alongside the Udacity Cloud Engineering Nanodegree. It allows users to register and log into a Web client, post photos to the feed, and process photos using an image filtering microservice.

The project is split into four parts:

1. Frontend - Angular Web application built with Ionic Framework.
2. Backend RESTFUL API - Node-Express application.
3. The Image Filtering Microservice - It is a Node-Express application that runs a simple script to process images.
4. Infrastructure as Code with Terraform and Elastic Kubernetes Service

- **Tech Stack:** Angular, Ionic, Nodejs, Express, REST API, PostgreSQL, Docker, Kubernetes, CircleCI, EKS, Terraform, S3.

## DASHBOARD METRICS

[Github](#)

This project is a full-fledged demonstration of microservices orchestration and observability in a complex architecture. It harnesses the power of various cutting-edge technologies to create a scalable and manageable system. It revolves around a Flask-based microservices application that employs Docker for containerization and utilizes k3s as a lightweight Kubernetes distribution. This project showcases end-to-end observability and monitoring capabilities with MongoDB as the database, Jaeger for distributed tracing, Prometheus for metrics collection, Grafana for data visualization, and OpenTelemetry for tracing instrumentation.

- **Tech Stack:** OpenTelemetry, Docker, K3s, ArgoCD, Helm, Flask, Vagrant, Kubernetes, Jaeger, Grafana, Prometheus, MongoDB

## UDACONNECT

[Github](#)

For a company that was building an app that uses location data from mobile devices. The company has built a POC application to ingest location data named UdaTracker. This POC was built with the core functionality of ingesting location and identifying individuals who have shared close geographic proximity. Management loved the POC, so now that there is buy-in, they want to enhance this application. I was tasked to enhance the POC application into an MVP to handle the large volume of location data that will be ingested. To do so, I refactored the application into a microservice architecture using message-passing techniques. It's easy to get lost in the countless optimizations and changes that can be made: My priority was to approach the task as an architect and refactor the application into microservices.

- **Tech Stack:** Flask, Python, SQLAlchemy, PostGIS, PostgreSQL, Vagrant, VirtualBox, K3s, Kubernetes,

Docker, Kafka, gRPC

## EDUCATION

**National Open University of Nigeria** - B.S.c Computer Science

December 2014 to March 2019

## CERTIFICATIONS AND DEGREES

**ExamPro** - Certificate of Completion, Terraform Beginner Bootcamp

September 2023 to October 2023

**ExamPro** - Certificate of Completion, AWS Cloud Project Bootcamp

February 2023 to July 2023

**Udacity** - Nanodegree, AWS Cloud Engineer

May 2022 to July 2022

**Udacity** - Nanodegree, Cloud Native Application Architecture

June 2021 to December 2021