

# Shaquille Cahill

## DEVOPS ENGINEER

**E-mail** : shaqcahill@gmail.com

**Phone Number** : (641)583-9086

**LinkedIn** : <https://www.linkedin.com/in/shaquillecahill/>

**Address** : Des Moines, IA

**GitHub** : <https://github.com/Shaq-C>

### TECHNICAL SKILLS

Python | SQL | HTML | CSS | **Linux** | Git | GitHub | **Cloud (AWS)** | CloudFormation | **EC2** | ECR | DynamoDB | CloudFront | S3 | API GW | **Lambda** | EFS | Boto3 | CloudWatch | **RDS** | MariaDB | **ALB with ASG** | Apache | **Nginx** | Snapshots | Volumes | VPC | Route53 | Certificate Manager | IAM | AMI | Jenkins | **Docker** | **Kubernetes** | Jira | Slack | **Ansible** | **Terraform** | Scrum/Kanban | Javascript

### PROFESSIONAL EXPERIENCE

**Nioyatech LLC**

Oct 2022 - Present

**DevOps Engineer**

- Hands-on experience in AWS provisioning and solid knowledge of AWS services like **EC2**, **S3**, **ELB**, **RDS**, IAM, Route 53, VPC, Autoscaling, CloudFront, Cloud Watch, Cloud Formation, Security Groups, **EKS**, **ECR**, and coding applications with **Python**.
- Built and Managed AWS infrastructure As A Code with **Cloud Formation** in 10+ projects.
- Worked on **Containerization** and Virtualization Projects with **Docker** and **Kubernetes**.
- Created and maintained fully automated **CI/CD pipelines** for code deployment using **Jenkins** and **Git**.
- Wrote **Ansible** playbooks and modules to deploy, configure, and manage 30+ servers.
- Automated the build & deployments for 20+ applications from end-to-end using **CI/CD** strategies with various DevOps tools and technologies.
- Monitored, analyzed, and reported performance statistics for cloud-hosted environments by using **AWS Cloud Watch**, **Prometheus**, **Grafana**.
- Built and configured 200+ EC2 instances on AWS cloud platform, the configuration of elastic load balancer (ELB) for traffic control for the EC2 instances and S3.
- Used **Maven** as build tools on Java projects for the development of build artifacts on the source code.

**Source Allies**

July 2022 - Present

**Teammate Services/ Apprentice**

- Assisted in building applications in languages such as Python and Javascript.

- Assisted in interviewing potential candidates and assessing technical knowledge.
- Organized AWS and Kubernetes trainings
- Assist in recruitment of potential candidates.

## SOFT SKILLS

Analytical Thinking | **Problem-Solving Abilities** | Self-Motivated | **Strong Communication** | Project Management | Planning | **Time Management** | Adaptability | **Continuous Learning** | **Collaboration & Teamwork** | Coaching Mindset | Resilience | Curiosity | Emotional Intelligence | **Creativity**

## PROJECTS

### ML-HUB Application

ML-Hub is an AI and NLP for healthcare project that provides software, models, and data to help healthcare and life science organizations build, deploy, and operate AI projects.

- Dockerized Web Application developed in React, Java Springboot, and Spring Cloud Frameworks and integrated with PostgreSQL database.
- Use Gateway API and Generic Data API for Project dockerized microservices.
- AWS Cognito and EKS are the backbones of our Project.
- Used Gitlab as a project management tool.
- Created full microservices, Gitlab CI/CD Pipelines, and deployment on a Kubernetes cluster with monitoring.
- Built Dockerfiles for each microservices.
- Used Terraform as Infrastructure as a Code (IaaS) to automate the creation of infrastructure in each pipeline.
- Handled the dynamic inventory that includes EC2 instances into the inventory by checking their tags.
- Load Balancers and Auto Scaling Groups for high availability and scalability.
- Automated the build & deployments for many applications from end-to-end using CI/CD strategies with various DevOps tools and technologies.

### Project Boss

- Modular Customer Relationship Management (CRM) application for end-to-end production.
- Deployment on K8s cluster to AWS.
- Used Gitlab Issues as an agile implementation tool.
- Created full microservices, CI/CD Pipelines with GitLab, and deployment on a Kubernetes cluster with monitoring.
- Code developed in NodeJS.
- Built Dockerfiles for each microservices.
- Built a Docker Compose file to work in a development environment.

- Used Terraform as Infrastructure as a Code (IaaS) to automate the creation of infrastructure in each pipeline.

### **Microservice Project:**

In this project, fully automated microservices application is used. Dockerized Web Application is developed in Java Springboot and Spring Cloud Frameworks and integrated with MySQL database. The project is to create full microservices, CI/CD Pipelines, and deployment on a Kubernetes cluster with monitoring. Kubernetes cluster is created and managed with Rancher. Jenkins is used as the CI/CD automation tool and created all the infrastructure on AWS EC2 Service. Also, GitHub is used as the version control system during the whole process. Base branches namely master, dev, and release are prepared for the DevOps cycle.

### **Kubernetes Microservice Phonebook:**

Phonebook Microservice Web Application aims to create a web application with MySQL Database using Docker and Kubernetes to give the understanding of Microservice architecture. In this application, there is a frontend service and a backend service to interact with database service. Each service will be managed by a Kubernetes deployment. The backend service will be a gateway for the application, and it will serve the necessary web pages for create, delete and update operations while the frontend service will serve a search page in order to conduct read operations. To preserve the data in the database, persistent volume and persistent volume claim concepts adopted.

### **Bookstore API:**

Bookstore Web API Application with MySQL created a bookstore web service using Docker for containerization of an application. The application code is deployed as a RESTful web service with Flask using Dockerfile and Docker Compose on AWS EC2 Instance using Terraform.

### **AWS Capstone Project:**

A web application written by Django Framework on AWS Cloud Infrastructure and deployed as a Blog Page Application. It has ALB with ASG of EC2 Instances and RDS on defined VPC, around which S3, Lambda and DynamoDB services are set. In addition, the Cloudfront and Route 53 services are located.

## **EDUCATION**

**Des Moines Area Community College**

Sep 2016 - July 2018

Associates in Arts