Archives available at journals.mriindia.com

International Journal of Recent Advances in Engineering and Technology

ISSN: 2347 - 2812

Volume 14 Issue 02s, 2025

Microservices CI/CD Pipeline on Cloud Computing (QR Code Generator)

[1]Sonali Lokare, [2]Om Kadam, [3]Vaishnavi Dube, [4]Punam Jathar, [5]Digvijay Vitkar

[1,2,3,4,5] Department of Computer Science and Engineering YSPM's Yashoda Technical Campus, Satara, Maharashtra, India

Email: [1]sonalilokare@gmail.com, [2]omvikaskadam3538@gmail.com, [3]vaishnavidube03@gmail.com, [4]punamjathar19@gmail.com, [5]digvijay.vitkar30102001@gmail.com

Peer Review Information

Submission: 21 Oct 2025

Revision: 18 Nov 2025

Acceptance: 05 Dec 2025

Keywords

DevOps, CI/CD, QR Code Generator, AWS, Jenkins, Docker, Ansible, Terraform, Monitoring, Infrastructure as Code

Abstract

This research paper presents the design and deployment of a DevOps-based Continuous Integration and Continuous Deployment (CI/CD) pipeline for a Python-based QR Code Generator web application. The application, developed using Flask and PyWebIO, is containerized using Docker and deployed using Docker Swarm for scalability and high availability. The CI/CD pipeline is orchestrated by Jenkins, with source code hosted on GitHub. Infrastructure provisioning is automated using Terraform, while Ansible manages configuration and deployment. The application is hosted on AWS EC2 instances, ensuring cloud scalability. For monitoring, Nagios is used to track system health and uptime, while Datadog provides real-time performance analytics and alerting. This project demonstrates a full DevOps lifecycle—from code commit to production deployment—leveraging open-source tools and cloud infrastructure. The implementation enhances deployment speed, operational visibility, and infrastructure reliability in a production-like environment, serving as a model for modern, automated web application delivery.

Introduction

In the evolving landscape of software engineering, the demand for faster, more reliable, and scalable application delivery has driven the widespread adoption of DevOps practices. DevOps bridges the gap between development and operations teams by enabling Continuous Integration and Continuous Deployment (CI/CD) through the use of automation, monitoring, and cloud infrastructure. This research project focuses on implementing a robust CI/CD pipeline to automate the development, testing, and deployment of a Python Flask-based QR Code Generator web application using a modern DevOps toolchain.

The application is designed to generate dynamic QR codes based on user input, providing a simple yet effective use case for demonstrating full lifecycle DevOps practices. The CI/CD pipeline integrates multiple tools, including GitHub for source control,

Jenkins for automation, Docker for containerization, and Docker Swarm for orchestrating containers across a distributed cloud environment.

Infrastructure provisioning is handled using Terraform, allowing infrastructure to be defined as code and deployed on AWS Elastic Compute Cloud (EC2) instances. Ansible is used for configuration management and application deployment, ensuring consistency across environments. To achieve high availability and observability, Nagios and Datadog are integrated for real-time service monitoring and infrastructure performance tracking.

This project simulates a production-grade environment where every code commit triggers an automated workflow—from building and testing the code to deploying the updated application. The goal is to minimize manual intervention, reduce human error, and accelerate the software delivery process.
By demonstrating an end-to-end DevOps pipeline for a cloud-hosted web application, this research serves as a practical example of how modern software delivery models can be implemented using open-source tools and cloud technologies. The approach enhances operational efficiency, reduces deployment time, and ensures consistent application performance.

Experimental Setup

Table 1: Experimental Setup

Component	Description	Tools
Project	QR Code Generator with Microservices & CI/CD	Cloud Platform (AWS, Azure)
Version Control	Code Management	GitHub
CI/CD Tools	ate builds, tests, & deployments	Jenkins, Terraform, Ansible
Microservices	pendent services to (QR generation)	CI/CD
Containerization	rize services for consistency	Docker Swarm
Orchestration	containers (scaling, availability)	Kubernetes services
Database	Store QR data and user Info.	Pywebio, flask
Testing	utomate unit and integration tests	Jenkins
Deployment	cloud deployment	AWS, Kubernetes
Monitoring	tem performance and logs	a Dog, Ngions, ELK, Grafana

System Design and Toolchain Overview

A. Project Objective:

To implement a QR Code Generator web application with fully automated deployment on AWS using a DevOps toolchain.

B. Key Components and Tools

- GitHub: Source code version control
- Jenkins: Automates the CI/CD pipeline
- Docker & Docker Swarm: Containerization and orchestration
- Ansible: Configuration management and Docker Swarm setup
- Terraform: Infrastructure provisioning on AWS (EC2, Security Groups)
- Nagios: Service and health monitoring
- Datadog: Infrastructure performance monitoring

Methodology

A. Application Development

The QR Code Generator app was developed in Python using Flask and PyWebIO. It accepts input and generates downloadable QR codes dynamically.

B. CI/CD Implementation with Jenkins

- Jenkins file includes stages for cloning GitHub repo, building the Docker image, pushing to Docker Hub, and deploying via Ansible.
- Jenkins server runs on an AWS EC2 instance.

C. Docker and Swarm Setup

- The application is containerized using Docker.
- Docker Swarm is initialized on provisioned AWS EC2 instances using Ansible.
- The web service is deployed in a scalable manner using docker service create.

D. Infrastructure Provisioning via Terraform

- Terraform code provisions EC2 instances and security groups in AWS.
- Outputs from Terraform are used as dynamic inventory for Ansible deployment.

E. Monitoring and High Availability

- Nagios monitors the health of all EC2 instances and services (CPU, disk, memory).
- Datadog provides visual metrics and alerts for infrastructure and container

performance.

Proposed Work

The proposed work involves developing a Microservices-based architecture for a QR Code Generator application, deployed on a cloud platform. The application will be divided into distinct microservices, each handling a specific function such as QR code generation, user management, and data storage. A CI/CD pipeline will be implemented using tools like Jenkins, GitLab CI, or CircleCI to automate the testing, building, and deployment of microservices. Docker will be used for containerization, ensuring smooth scalability and portability across cloud environments. The CI/CD pipeline will ensure continuous integration and delivery of code, enabling rapid updates and bug fixes. Automated testing and monitoring tools will be integrated to ensure high-quality service. This setup will allow for faster deployment cycles, better resource management, and greater resilience for the QR Code Generator service.

Explainable Importance

The importance of a Microservices CI/CD pipeline on cloud computing for the QR Code Generator project lies in its ability to streamline development, enhance scalability, and ensure high availability. By adopting a microservices architecture, the application is broken into smaller, independently deployable services, making it easier to develop, test, and scale specific components without affecting the entire system. The CI/CD pipeline automates the process of building, testing, and deploying thesemicroservices, ensuring faster releases and reducing human error.

Continuous integration ensures that code changes are tested and merged regularly, while continuous deployment allows updates to be pushed seamlessly to production with minimal downtime. Leveraging cloud platforms like AWS, Azure, or Google Cloud, along with Docker and Kubernetes, provides scalability, fault tolerance, and resource optimization.

Block Diagram

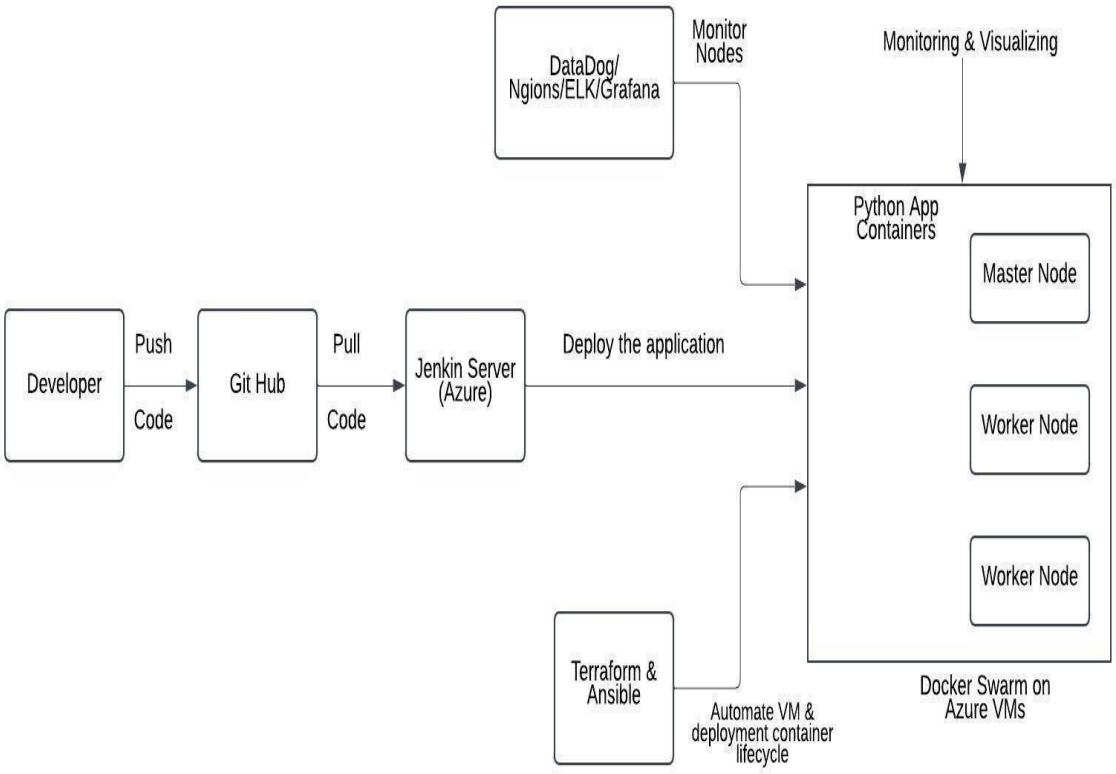

Fig.1 Block Diagram of CI/CD Pipeline

Automated Systems

The "Automated Systems for Microservices CI/CD Pipeline on Cloud Computing (QR Code Generator)" project focuses on automating the development, testing, and deployment of a QR Code Generator application using a microservices architecture in the cloud. The system breaks the application into smaller, independently deployable services, enhancing scalability and flexibility. A Continuous

Integration/Continuous Deployment (CI/CD) pipeline automates the process, ensuring faster iterations and continuous delivery with minimal downtime. Tools like GitHub, Jenkins, Docker, and Kubernetes, combined with cloud platforms such as AWS, Azure, or Google Cloud, are used for seamless orchestration and automation. Docker ensures consistent environments, while Kubernetes manages container orchestration and scaling. The cloud infrastructure ensures dynamic resource allocation, improving reliability, availability, and performance for the QR Code Generator service.

Suggestions and Future Directions

A. Implement Advanced Monitoring and Automation

Integrating tools like Prometheus, Grafana, and the ELK stack for real-time monitoring, centralized logging, and performance tracking can enhance the CI/CD pipeline. Future improvements could focus on AI-driven anomaly detection for proactive issue resolution and faster troubleshooting.

B. Adopt DevSecOps for Enhanced Security

Incorporating security into every stage of the CI/CD pipeline through DevSecOps practices is essential. Tools like Sync and Checkmark can automate vulnerability scanning in code and containers. Future work could involve automating penetration testing and continuous compliance monitoring to maintain security without slowing down development.

C. Explore Serverless Computing for Scalability

Transitioning to a serverless architecture (e.g., AWS Lambda, Google Cloud Functions) for parts of the QR Code Generator can improve scalability and cost efficiency. This future direction would automate resource provisioning, reduce infrastructure overhead, and further streamline the CI/CD pipeline, allowing for more flexible scaling as demand fluctuates.

Conclusion

This research implements a comprehensive CI/CD pipeline for a cloud-native QR Code Generator application using DevOps and microservices architecture. The use of open-source tools and cloud infrastructure significantly improves deployment efficiency, system resilience, and performance. With future enhancements such as serverless computing, DevSecOps, and AI-powered monitoring, this system can evolve into a highly scalable and intelligent web service deployment model.

Acknowledgement

We sincerely thank our guide, the faculty members, and the Department of Computer Science and Engineering at YSPM's Yashoda Technical Campus, Satara, for their constant guidance and support throughout this project.

References

R. Santos, P. Soares, E. Rodrigues, P. H. M. Maia, and A. Silveira, "How Blockchain and Microservices are Being Used Together: A Systematic Mapping Study," in *2022 IEEE/ACM*.

R. S. de O. Júnior, R. C. A. da Silva, M. S. Santos, W. Albuquerque, H. O. Almeida, and D. F. S. Santos, "An Extensible and Secure Architecture Based on Microservices," in *2022 IEEE Int. Conf. Consum. Electron. (ICCE)*, 2022.

A. Chatterjee and A. Prinz, "Applying Spring Security Framework with Keycloak-Based OAuth2 to Protect Microservice Architecture APIs: A Case Study," *Sensors*, vol. 22, no. 5, p. 1703, 2022.

I. Rossi, "Cloud-Native DevOps: Unleashing the Power of Microservices on AWS Infrastructure," *Integrated Journal of Science and Technology*, vol. 1, no. 2, 2024.

H. T. Nguyen, "A Comprehensive CI/CD Pipeline and Google Cloud Deployment for Web Application," 2023.

U. Yilmaz, M. Di Carlo, and P. Harding, "Building a Control System with Cloud Native Technologies: Leveraging Kubernetes and TANGO-controls for CI/CD Practices in SKA Observatory Software," in *Software and Cyberinfrastructure for Astronomy VIII*, vol. 13101, SPIE, 2024.

J. Mulder, *Multi-Cloud Architecture and Governance: Leverage Azure, AWS, GCP, and VMware vSphere to Build Effective Multi-Cloud Solutions*, Packt Publishing Ltd, 2020.

X. Sun, S. Boranbaev, S. Han, H. Wang, and D. Yu, "Expert System for Automatic Microservices Identification Using API Similarity Graph," *Expert Systems*, e12158, 2022.