SHARATH BODIGE

◆ 9581556123 ◆ sharathgoud.bodige@gmail.com

CAREER OBJECTIVE

Enthusiastic and self-motivated aspiring AWS DevOps Engineer with strong knowledge of cloud concepts, AWS services, and hands-on project experience. Skilled in designing, deploying, and managing cloud infrastructure using AWS Free Tier. Seeking an entry-level role to contribute to innovative cloud-based solutions and grow in a dynamic technology environment.

TECHNICAL SKILLS

- **Version Control:** Git, GitHub
- **CI/CD Tools:** Jenkins
- **Containerization:** Docker
- Build tool: Maven
- **Scripting Languages:** Bash
- **Cloud Platforms:** Amazon Web Services (EC2, S3, IAM, VPC, Cloud watch)
- **Project & Ticketing Tools:** JIRA
- **Operating Systems:** Linux, Windows

PROJECTS

1. CI/CD Pipeline for Java Application using Jenkins, Maven, Docker & ALB

Project Summary: Implemented a complete CI/CD pipeline using Jenkins, Git, Maven, Docker, and AWS EC2. Configured Jenkins on an EC2 instance to automatically pull code from GitHub, perform Maven build, package the application, build Docker images, and deploy the updated container to a dedicated EC2 application server. Integrated the deployment environment behind an AWS Application Load Balancer (ALB) for high availability, traffic distribution, and health monitoring. Automated rolling updates using Jenkins pipeline, improving deployment speed and consistency. CloudWatch was used for logging and performance monitoring.

Roles & Responsibilities:

- Built a fully automated CI/CD pipeline using Jenkins, Git, Maven, and Docker to deploy a Spring Boot application on AWS EC2.
- Configured Jenkins Master on EC2 with GitHub webhooks to trigger automatic builds and deployments.
- Implemented Maven build automation to generate production-grade artifacts for containerization.
- Created a lightweight Docker image for the application ensuring consistent deployments across environments.
- Automated container lifecycle: stopping old containers, removing them, and deploying a new version via Jenkins pipeline.
- Provisioned a dedicated EC2 Docker host for application runtime and integrated deployment steps through SSH in Jenkins.
- Configured AWS ALB (Application Load Balancer) to evenly route traffic and provide health checks for application availability.

- Secured infrastructure using IAM roles, security groups, and key-based authentication for Jenkins→EC2 communication.
- Integrated CloudWatch Logs & Metrics for centralized monitoring of application/container health.
- Designed a robust deployment strategy to achieve zero-downtime deployments using ALB target group health checks.
- Containerized the entire environment to reduce configuration drift, simplify deployment, and improve reliability.
- Wrote end-to-end documentation including pipeline workflow, infrastructure diagram, and deployment runbook.
- Enabled faster releases by optimizing Jenkins pipeline stages and reducing manual steps in deployment.
- Managed versioning of Docker images and automated cleaning of unused containers and images.
- Ensured high availability and scalability by leveraging AWS ALB and stateless container deployment.

2. CI/CD Pipeline for Web Application using Jenkins, Docker, GitHub & EC2

Project Summary: I built an end-to-end CI/CD pipeline to deploy a two-tier application (frontend + backend) using Jenkins, Docker, and AWS EC2. The application is accessible through a custom DNS using AWS Route53. The entire deployment process is fully automated through Jenkins.

Roles & Responsibilities:

- Created and configured EC2 instance for hosting containers
- Installed Docker & Docker Compose (if needed)
- Allocated Elastic IP to maintain permanent DNS mapping
- Configured Route53 A-record to make the application accessible via domain (e.g., mydevopsproject.in).
- Setup Nginx container to serve frontend over port 80
- Deployed Node.js API in backend container on port 3000
- Ensured cross-container communication using DNS-based API URLs
- Updated frontend code to call backend via domain
- Integrated GitHub with Jenkins using Webhooks
- Wrote a declarative Jenkinsfile for automated build → push → deploy
- Configured Jenkins to build two Docker images: frontend & backend
- Integrated DockerHub for image storage and versioning
- Automated deployment to EC2 via SSH using Jenkins pipeline

SOFT SKILLS

- Problem Solving
- Communication Skills
- Time Management
- Team Collaboration

EDUCATION

Bachelor of Technology in **ECE** from **JNTUH**, 2019.

DECLARATION

I hereby declare that the above information furnished by me is true and correct to the best of my knowledge.