Sushant Sonbarse

+91 9146570820 | sushantsonbarse07@gmail.com | Nagpur, India | github | linkedin

PROFILE SUMMARY

DevOps Engineer with 9 months of experience working in orchestration, containerization, and scalable CI/CD pipeline implementation.

EDUCATION

Rashtrasant Tukdoji Maharaj Nagpur University *Nagpur,Maharashtra*

Bachelors In Computer Science *2020*-2023

- Top 10% in Academics

SKILLS

Technical skills: Automation, Containerization Tools, Infrastructure as code (Iac), Containerization, Monitoring and Logging

Tools: Linux, Docker, Jenkins, Ansible, GitLab CI, Kubernetes, Terraform, Azure, GCP , Maven, Ansible, Datadog, Bash Scripting, Git, GitHub Actions, Prometheus and Grafana

EXPERIENCE

HisanLabs Pvt Ltd *Pune,Maharashtra*

DevOps Engineer Intern 09/*2024* – 05/2025

- Implemented Angular and Spring Boot for deploying a microservices architecture for a CRM real estate platform with 35 services.
- Used Terraform to automate AWS infrastructure deployment (EKS, S3, RDS), which made our deployments much more consistent and reduced mistakes.
- By streamlining **CI/CD** pipelines and automating infrastructure provisioning, deployment time was cut by **75%**, and release duration was shortened from 4 to 1 hour.
- Worked with dev teams to fix networking issues, debug shell scripts, and solve deployment problems

PROJECTS

End to End CI/CD Implementation

- Reduced deployment time by **85%** by implementing an end-to-end DevOps Automation Pipeline for an e-commerce platform.
- Docker images have been optimized to decrease size and increase build efficiency.
- Helm charts were used to simplify Kubernetes deployments, which decreased deployment errors and sped up release cycles by **30%**.
- Developed Python automation scripts to reduce manual labor by **60%** in infrastructure monitoring, log analysis, and deployment.

Three Tier Architecture

- Implemented AWS services, such as EC2, VPC, RDS, S3, Lambda, API Gateway, Load Balancer, Route 53, CloudFront, and CloudWatch, to design a three-tier serverless architecture that is both scalable and economical.
- The **EFK** stack was used to monitor application and infrastructure logs, allowing for real-time insights, quicker issue detection, and a 40% decrease in incident resolution time.

ACHIEVEMENTS

- Reduced onboarding time by 30% by maintaining documentation for CI/CD pipelines, infrastructure, and troubleshooting manuals.