

Hritik Satish Munde

San Jose, CA | +1 930-904-4405 | hritik16munde@gmail.com | linkedin.com/in/hritik-munde-922b43183/ | github.com/hrikimunde | hritikmunde.me/

SUMMARY

DevOps Engineer with 2+ years of experience building, deploying, and operating cloud-native applications using Kubernetes, Terraform, and CI/CD pipelines. Strong focus on automation, infrastructure reliability, and improving developer productivity through standardized deployment and monitoring practices.

SKILLS

CI/CD: GitLab CI, GitHub Actions, Jenkins

Infrastructure & Automation: Terraform, Ansible, Vault, Bash scripting, Git, GitHub

Kubernetes & Containers: Kubernetes (EKS), Docker, ContainerD, Helm, Kustomize, ArgoCD, Nginx

Cloud & Systems: AWS (EKS, EC2, VPC, IAM, S3, RDS, CloudWatch, Route53), Linux (Ubuntu), TCP/IP, DNS

Observability: Prometheus, Grafana, Loki, ELK Stack (Elasticsearch, Logstash, Kibana), Splunk

Reliability Engineering: Incident Response, SLO/SLI Monitoring, On-Call, Runbooks, Postmortems

Programming: Python, Go, Bash, SQL

Certifications: RHCSA, Terraform Associate

EXPERIENCE

DevOps Engineer

Jul 2022 – Jul 2024

UST

Pune, India

- Built and maintained CI/CD pipelines using GitHub Actions and GitLab CI to automate container builds, testing, and GitOps-based deployments, supporting 40+ services across development and staging environments
- Standardized Docker-based application deployments using Helm and ArgoCD across 4 environments, reducing deployment-related issues by 22%
- Provisioned cloud infrastructure using Terraform and Ansible for development and staging environments, reducing manual setup time from 2 days to 6 hours
- Partnered with application teams to improve deployment workflows and release processes, reducing release failures by 20% and improving overall release reliability
- Supported production Kubernetes clusters hosting 45 microservices, contributing to sustained 99.5% service availability
- Built and maintained Prometheus and Grafana dashboards covering pod health, latency, and resource usage, enabling earlier detection of 28% of production issues
- Developed Python and Bash automation scripts for log analysis and health checks, reducing manual troubleshooting effort by 5 engineer-hours per week

Software Engineer Intern

Aug 2021 – May 2022

Tata Motors

Pune, India

- Developed backend APIs in Java and Python for a vehicle diagnostics platform used by 500 daily users
- Optimized SQL queries and backend processing logic, reducing diagnostic data processing time by 25%
- Assisted in debugging Linux-based backend services and system integrations, contributing to a 20% reduction in recurring diagnostic issues

EDUCATION

Indiana University Bloomington

Aug 2024 – May 2026

Master of Science, Computer Science | GPA: 3.9/4

Bloomington, IN

MIT Academy of Engineering

Aug 2018 – May 2022

Bachelor of Technology, Computer Engineering

Pune, India

PROJECTS

Self-heal Rollback System | ArgoCD, Prometheus, Kubernetes, Bash

Dec 2025

- Built rollback trigger that monitors pod restarts and HTTP 5xx rates; auto-reverts to last stable version under 3 minutes
- Triggered application rollback via ArgoCD within ~1–2 minutes when error rates exceeded defined thresholds
- Tested against 5 intentionally broken deployments; caught 4 instantly, 1 required tuning threshold from 10% to 5% error rate

SLO Monitoring Platform | Sloth, Prometheus, Grafana, Kubernetes, ArgoCD

Nov 2025

- Defined SLOs for 3 sample microservices using Sloth and Prometheus recording rules
- Replaced 15+ threshold-based alerts with 3 SLO-based alerts using Sloth; on-call noise dropped from 30 alerts/day to 4
- Designed Grafana dashboard showing error budget burn rate leading team to now delays risky deploys when budget is below 20%

Distributed Logging System | Go, Fluent Bit, Kubernetes API

Sep 2025

- Wrote Go service that tails logs from 20+ pods via Kubernetes API; streams to central Fluent Bit with pod labels
- Forwarded structured logs to Fluent Bit with namespace and label enrichment for easier filtering
- Configured basic alerts for error spikes, reducing debugging time by ~40% in test deployments