

VEERA REDDY RAVURI

+91 9908201599 • veerareddy.ravuri01@gmail.com • [LinkedIn](#) • [GitHub](#) • [Portfolio](#)

Cloud & DevOps Engineer

Vijayawada, AP • Open to Relocation

PROFESSIONAL SUMMARY

Production-focused Cloud and DevOps Engineer with 2+ years of production engineering experience on enterprise-scale billing infrastructure. Proficient in AWS, Terraform, Docker, GitHub Actions, and Linux systems administration. Caught a billing defect affecting 24,700+ customers preventing ~\$200K in revenue loss, and reduced MTTR by 30% across 100+ production incidents. Seeking a Cloud or DevOps role where I can apply production-proven reliability engineering and infrastructure automation to build systems that scale.

KEY IMPACTS

- Prevented ~\$200K revenue loss affecting 24,700+ customers by identifying and resolving a critical billing defect through production log analysis
- Reduced deployment failure impact window to under 2 minutes by implementing automated health-check-based detection and rollback mechanisms in CI/CD workflows
- Reduced MTTR by 30% across 100+ production incidents by implementing structured log analysis and RCA workflows
- Eliminated 90% manual infrastructure provisioning effort by codifying a 14-resource AWS environment using Terraform with remote state management and locking

SKILLS

- **Cloud:** AWS (EC2, VPC, IAM, Lambda, S3, CloudWatch, SNS, DynamoDB, ALB)
- **IaC & Automation:** Terraform, Ansible, GitHub Actions, GitOps
- **Containers:** Docker, Docker Compose, Nginx, Blue/Green, Canary Deployment
- **Security:** Trivy, Least-Privilege IAM, Immutable Image Tagging
- **Observability:** CloudWatch Alarms, SNS Alerting, JSON logging, Webhook Alerting
- **Languages & Databases:** Python, Bash, MySQL, PostgreSQL
- **Systems & Networking:** Linux/Ubuntu, TCP/IP, DNS, HTTP/S, systemd, journalctl, SSH, Git, YAML, JSON, Jira
- **Building:** Kubernetes, Helm, Prometheus, Grafana, AlertManager

Projects

Secure CI/CD Pipeline with Automated Rollback | *GitHub Actions, Docker, Python, FastAPI, AWS (S3, EC2, CloudWatch), Trivy*

- Reduced deployment blast radius during releases by routing 10% initial traffic via Blue/Green canary strategy using AWS ALB, enabling controlled validation before full rollout
- Maintained service reliability during failed releases by enabling automated rollback within 2 minutes using error-rate-based validation loops sampling 20 ALB requests per cycle and triggering dual rollback, ALB traffic shift back to Blue plus GitOps manifest revert when error rate exceeded 15%
- Secured the deployment pipeline by integrating Trivy into GitHub Actions to block critical CVEs, utilizing immutable commit-SHA tags to ensure every running container traces back to an exact source commit
- Improved deployment traceability and post-incident auditability by implementing GitOps-based version-controlled manifests with S3-backed deployment state tracking

AWS Production Infra Blueprint | *Terraform, Ansible, AWS (EC2, VPC, IAM, S3, DynamoDB, CloudWatch), Docker*

- Provisioned a 14-resource AWS environment entirely as code using Terraform, implementing S3 remote state and DynamoDB locking to enable safe concurrent operations
- Enforced infrastructure immutability by detecting and resolving state drift from manual console changes, creating a recovery runbook validated against 5 failure scenarios including state corruption
- Prevented observability blind spots by enforcing least-privilege IAM roles and resolving a critical CloudWatch logging failure caused by missing log stream permissions
- Validated full Ansible configuration idempotency across 4 roles achieving 0 changed tasks on second playbook run, and documented a deliberate NAT Instance vs NAT Gateway (\$8/mo vs \$32/mo) cost tradeoff with reasoning

Containerized Microservice Stack | *Docker, Docker Compose, FastAPI, PostgreSQL, Nginx*

- Reduced Docker image footprint by 46% (225MB to 120MB) and enforced non-root execution by implementing multi-stage builds on an Alpine base, accelerating CI/CD pull times
- Diagnosed and resolved a production-pattern failure where Nginx cached upstream DNS at startup and silently routed all traffic to one of two API replicas, fixing by configuring resolver 127.0.0.11 valid=5s to force 5-second re-resolution via Docker's embedded DNS server
- Prevented request loss during container restarts by implementing SIGTERM-based graceful shutdown with a 3-second drain window, ensuring safe termination of in-flight traffic

Linux Reliability Toolkit | Python, psutil, systemd, journalctl, YAML, cron, Git

- Reduced system failure detection time to under 5 minutes by building a Python-based monitoring daemon with automated service recovery and webhook-based alerting
- Eliminated a silent failure mode by engineering a startup privilege self-check, ensuring the daemon actively validates recovery permissions during initialization rather than failing silently during a live incident
- Standardized telemetry by implementing structured JSON logging, enabling seamless query compatibility with log aggregation pipelines like CloudWatch and ELK

Automated Incident Alert Pipeline | Python, AWS Lambda, S3, CloudWatch, SNS, GitHub Actions

- Constructed an event-driven serverless alerting pipeline (S3 → Lambda → CloudWatch → SNS), delivering high-severity log notifications within CloudWatch's 5-minute evaluation window
- Automated Lambda function deployments using GitHub Actions, achieving a 17-second continuous deployment lifecycle from code push to live AWS function updates

EXPERIENCE

HCLTech

Chennai, TN

Software Engineer – Production Systems & Automation

Jan 2024 – Present

Verizon billing platform · production incident management, reliability automation, and deployment operations

- Prevented ~\$200K revenue loss impacting 24,700+ customers by diagnosing and resolving a critical billing defect using production log analysis and SQL debugging during a live deployment
- Reduced MTTR by 30% across 100+ production incidents by implementing structured log tracing and root cause analysis workflows aligned with SLA targets
- Reduced release validation time by 85% (3 days to 4 hours) across 10+ deployments by automating 157 test cases using a Python-Selenium framework
- Maintained zero-downtime production deployments across multiple release cycles by executing controlled rollout strategies and validating change requests
- Resolved multiple production incidents by handling customer-facing tickets and collaborating across application, database, and infrastructure teams to restore service stability
- Ranked top 3 in corporate technical training cohort, onboarding to live Verizon production support within 2 months which is 4 months ahead of the standard timeline

Freelance

Remote

Technical Mentor

Aug 2022 – Dec 2023

- Directed technical mentorship for 5+ engineers and university students (including UIUC and University of Windsor), guiding them through complex, hands-on AWS infrastructure builds and academic capstone deployments
- Supported development of 10+ academic projects by teaching Python scripting, Linux administration, and web development fundamentals

SkillVertex

Bangalore, KA

Cloud Intern

Sep 2022 – Nov 2022

- Engineered an event-driven log monitoring pipeline using AWS Lambda and CloudWatch, automating anomaly detection to reduce alert latency by 45% compared to manual review
- Solidified foundational cloud skills through hands-on implementation of IAM least-privilege principles, automated quality gates, and containerized deployments

CERTIFICATIONS

- **GitHub Copilot** | Microsoft | Mar 2026
- **AWS Certified Solutions Architect - Associate** | AWS | In Progress | Q3 2026

EDUCATION

Narasaraopeta Engineering College

Guntur, AP

Bachelor of Technology, Computer Science & Engineering

MAY 2023

- **CGPA:** 8.54
- **Best Paper Award:** International Conference on AI & IT (Mar 2023)