

## Acme Corp – DevOps Onboarding Guide

Welcome to Acme Corp's DevOps Platform!

This guide is intended to onboard new engineers and provide a high-level overview of our CI/CD infrastructure, tools, and operational best practices.

### Section 1: Repository Setup

- Clone the service template repository from the internal GitHub Enterprise.
- Create a new branch from `main` and follow the naming convention: `feature/<ticket-id>-<short-description>`.
- Ensure all commits are GPG signed and push changes to trigger automated checks.

### Section 2: CI/CD Pipelines

- All services use GitHub Actions for continuous integration.
- Workflows include code linting, unit tests, SAST (via SonarQube), and Docker builds.
- On pull request approval, workflows automatically deploy to a staging environment via ArgoCD.

### Section 3: Infrastructure and Deployment

- Infrastructure is defined using Terraform and provisioned on AWS.
- Secrets are managed using HashiCorp Vault and injected during deployment.
- Kubernetes manifests are Helm-based and reside in the `infra/` folder of each repo.

### Section 4: Monitoring and Alerts

- Metrics: Prometheus scrapes application-level and infrastructure metrics.
- Logging: Loki is used for centralized log aggregation.
- Traces: OpenTelemetry is enabled in all microservices.
- Alerts are routed via Grafana and escalated using PagerDuty.

### Section 5: Security Guidelines

- All container images must be scanned with Trivy before deployment.
- Dependencies should be regularly checked using Dependabot.
- Access to production requires MFA and approval via ServiceNow workflow.

Thank you for being part of Acme Corp's Engineering Team!

For more help, visit the internal wiki or contact #devops-help on Slack.