# DevOps Project Documentation

# 1. Setup Instructions

## Prerequisites and Dependencies

Tools Required: Terraform ≥1.8.0, AWS CLI ≥2.15, kubectl ≥1.30, Helm ≥3.14, Jenkins (LTS), Docker ≥25.0, Trivy, Checkov, TruffleHog.

kubectl version

Client Version: v1.31.13

Kustomize Version: v5.4.2

Server Version: v1.31.13

AWS Resources: IAM user/role, S3 bucket for backend, EC2 Key Pair, 2× Rocky Linux 9.3 nodes.

## Step-by-Step Deployment Guide

Infrastructure Setup (Terraform):

terraform init  
terraform validate  
terraform plan -out plan.out  
terraform apply plan.out

Creates VPC, EKS Cluster, IAM roles, and S3 backend.

## Jenkins Setup

Install plugins: Pipeline, Docker, Kubernetes, Git, Trivy, Checkov, TruffleHog.

Configure AWS, Docker, kubeconfig credentials, and connect Jenkins to Docker Private Registry for secure image push/pull.

## Pipeline Stages

Stages: Checkout, Security Scans, Unit Tests, Build & Push, Deploy, Notifications.

## Troubleshooting Common Issues

|  |  |  |
| --- | --- | --- |
| Issue | Possible Cause | Resolution |
| Terraform backend lock error | Concurrent apply | Run terraform force-unlock <ID> |
| Helm deployment fails | Wrong context | Verify kubeconfig and namespace |
| Trivy/Checkov errors | Outdated plugin | Upgrade or reinstall plugin |

# 2. Architecture Decisions

## Cloud vs Local

AWS EKS chosen for scalability, IAM integration, and real-world reliability.

## Technology Stack Justification

|  |  |  |
| --- | --- | --- |
| Component | Technology | Reason |
| IaC | Terraform | Declarative, reusable, supports AWS |
| CI/CD | Jenkins | Flexible, plugin-rich |
| Security Scanning | Trivy, Checkov, TruffleHog | Comprehensive scanning coverage |
| Container Registry | Docker Private Registry | Ensures image security and access control |
| Deployment | Helm + Kustomize | Simplifies environment management |

## Security Implementation Approach

IAM least privilege, Trivy scans, Checkov compliance, TLS ingress, private subnets, Docker registry authentication.

## Scalability Considerations

Auto-scaling, reusable Terraform modules, CI/CD branch triggers, stateless app design.

# 3. Lessons Learned

|  |  |
| --- | --- |
| Challenge | Solution |
| Jenkins agent Docker build errors | Used dedicated EC2 with Docker-in-Docker |
| Checkov false positives | Added justified exceptions |
| Kustomize overlays mismatch | Linked Jenkins ${BUILD\_NUMBER} variable |

## Improvements for Future

Integrate SonarQube, Argo CD, Grafana/Prometheus, and add canary deployments.

## Alternative Approaches Considered

Evaluated GitLab CI/CD, Minikube, and Terraform Cloud—preferred full Jenkins control.

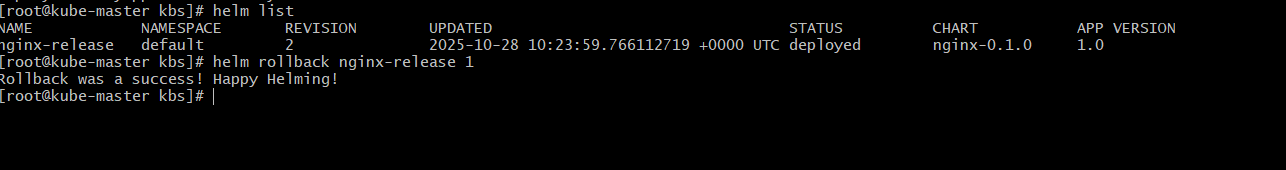
# 4. Demo Instructions

Testing Application: Get LoadBalancer URL and open in browser.

kubectl get svc -n app-namespace

Trigger CI/CD Pipeline: Push code to main branch, Jenkins auto-builds and deploys.

Rollback Instructions:



helm rollback myapp <previous\_revision> -n app-namespace  
git revert <commit-id>  
git push origin main