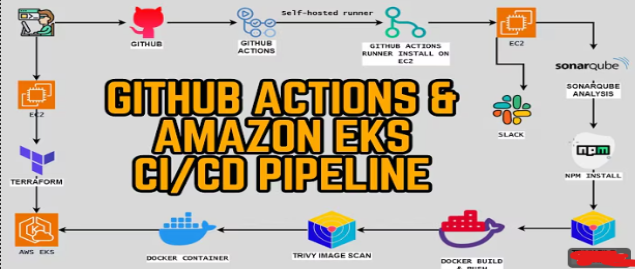
# GitHub Actions & Amazon EKS CI/CD Pipeline

**1. Project Overview**

This project implements a complete CI/CD pipeline using GitHub Actions and Amazon EKS. It includes Docker builds, image scans, Terraform deployment, SonarQube analysis, and Slack notifications.

**2. Architecture Diagram**

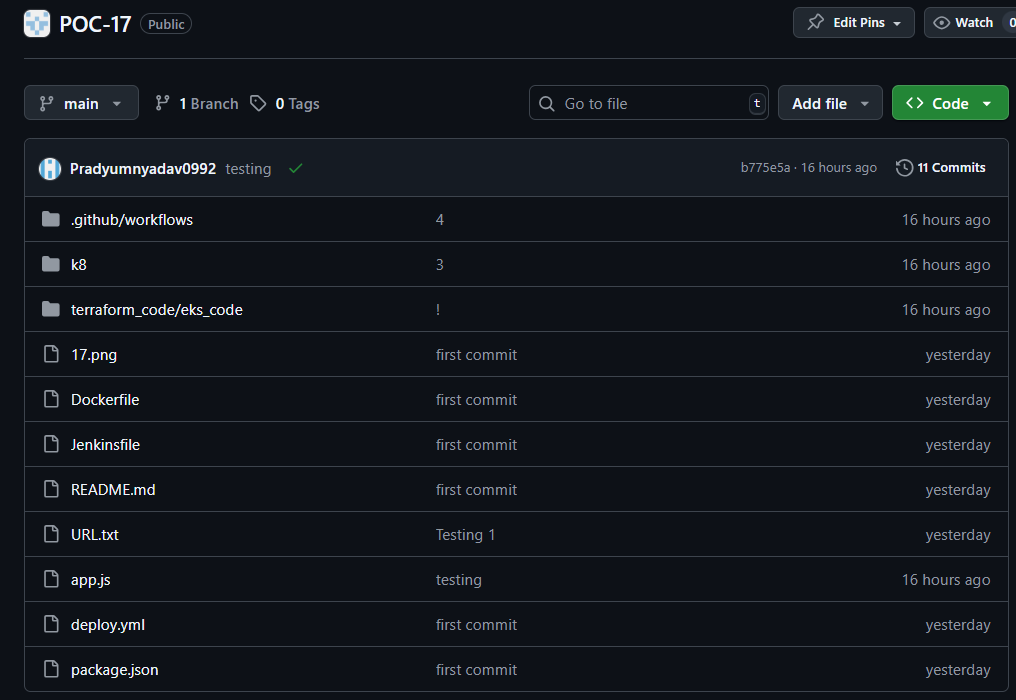


**3. Prerequisites**

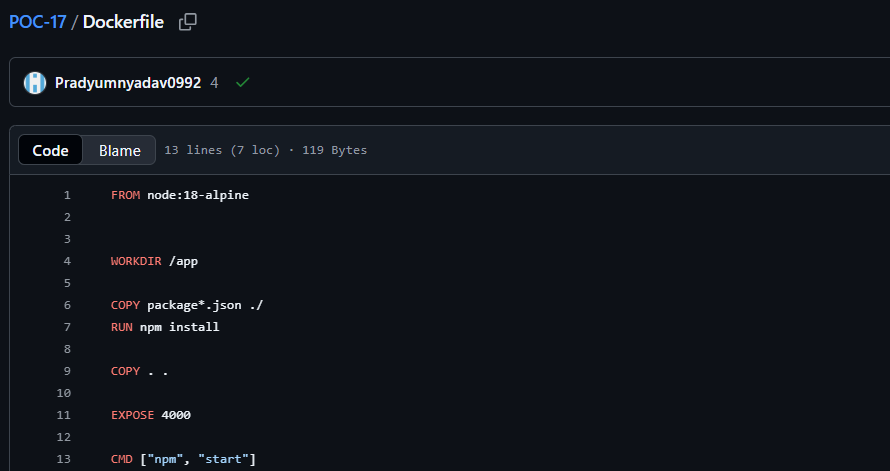
1. AWS account with necessary permissions.
2. EC2 instance for self-hosted GitHub Actions runner.
3. Terraform installed locally or on EC2.
4. Docker installed on EC2 and EKS nodes.
5. GitHub repository with source code.
6. SonarQube server accessible.

**4. Detailed Steps**

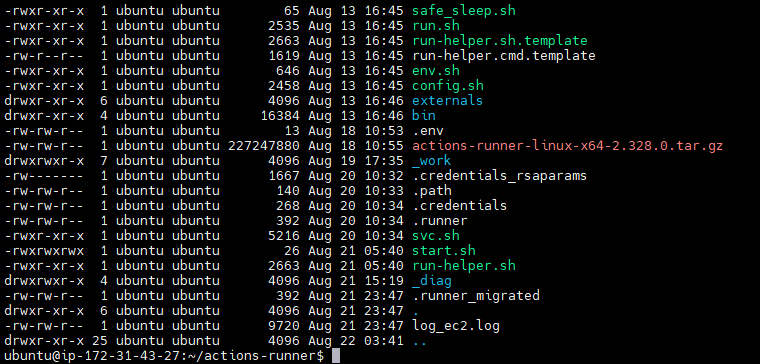
**Step 1: GitHub Repository Setup**



Create a docker file

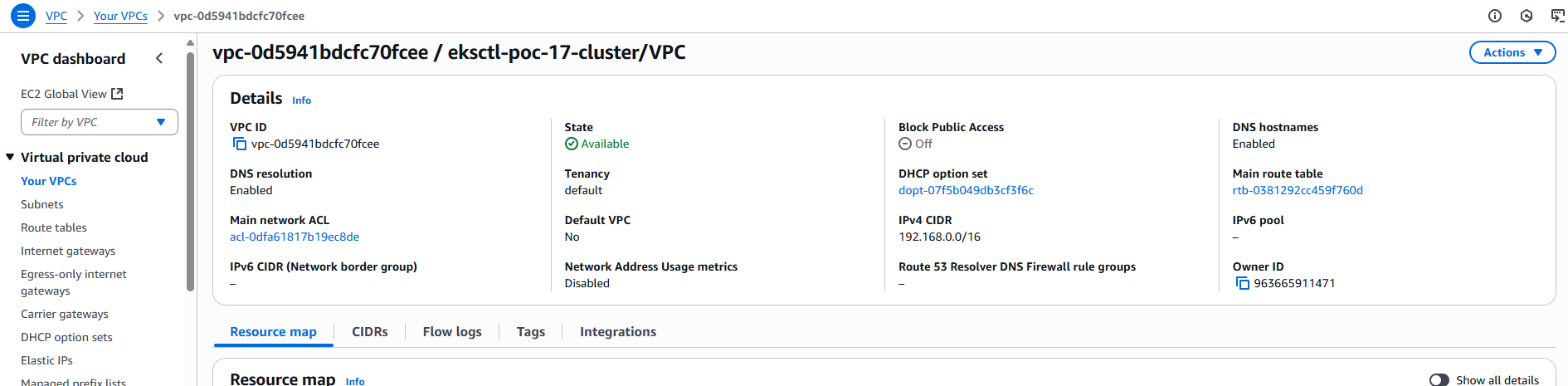


**Step 2: Setup Self-Hosted GitHub Runner**

1. Launch an EC2 instance.
2. Install GitHub Actions runner on EC2.
3. Configure it to connect to your repository.
4. Verify the runner is online in GitHub Actions.
5. 

**Step 3: Terraform Deployment to AWS EKS**

1. Create Terraform scripts for VPC, EKS, and necessary IAM roles.
2. Run terraform init and terraform apply.
3. Verify EKS cluster and nodes are created.

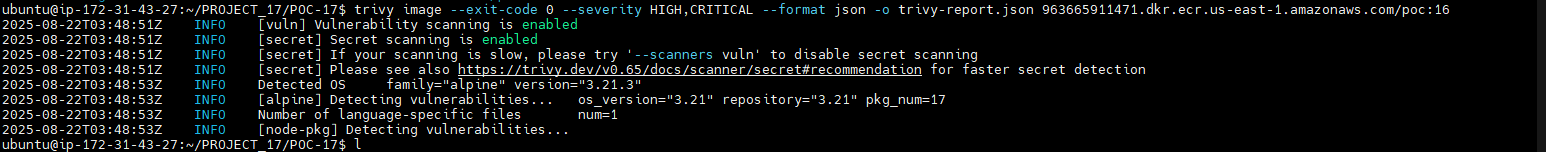


**Step 4: Docker Build**

1. Build Docker image from your application code:
2. docker build -t <image\_name>:<tag> .
3. Verify the image exists locally.

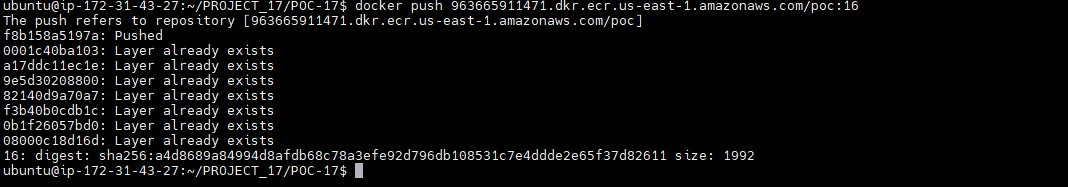
**Step 5: Trivy Image Scan**

1. Install Trivy for container vulnerability scanning.
2. Scan your Docker image for vulnerabilities:
3. trivy image <image\_name>:<tag>

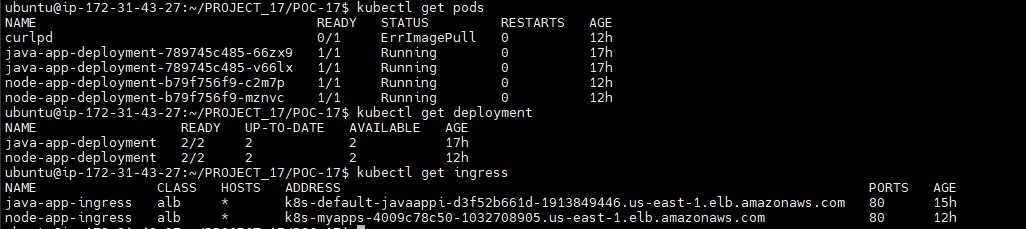


**Step 6: Push Docker Image to Registry**

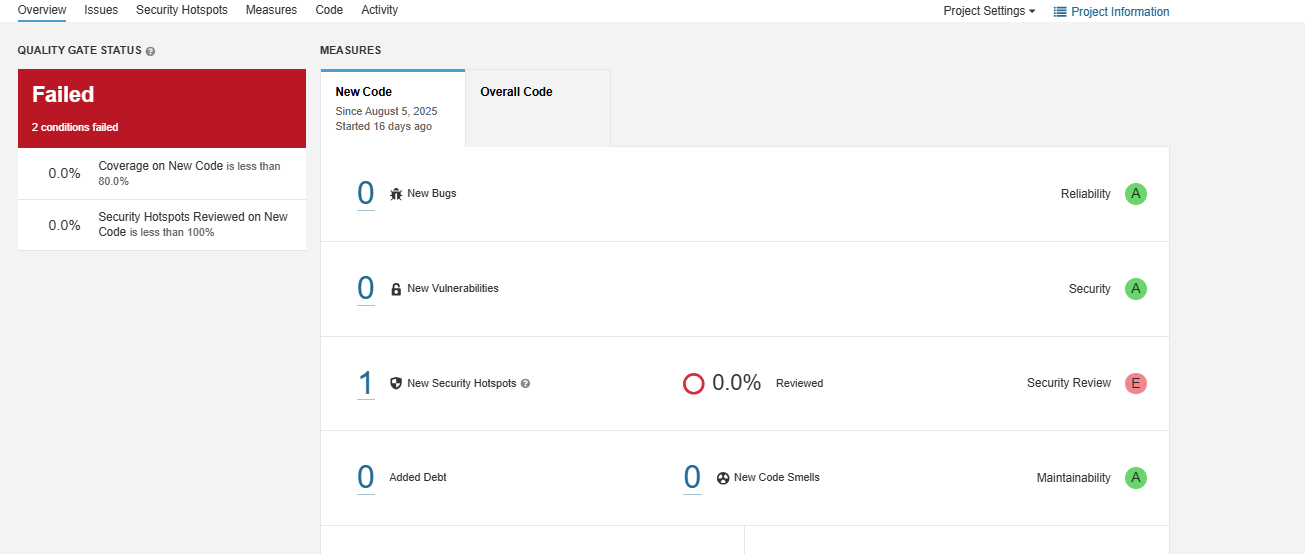
1. Tag and push the Docker image to Docker Hub or AWS ECR:
2. docker tag <image\_name>:<tag> <registry\_url>/<image\_name>:<tag>
3. docker push <registry\_url>/<image\_name>:<tag>
4. Verify the image is available in the registry.



**Step 7: Deploy to EKS**

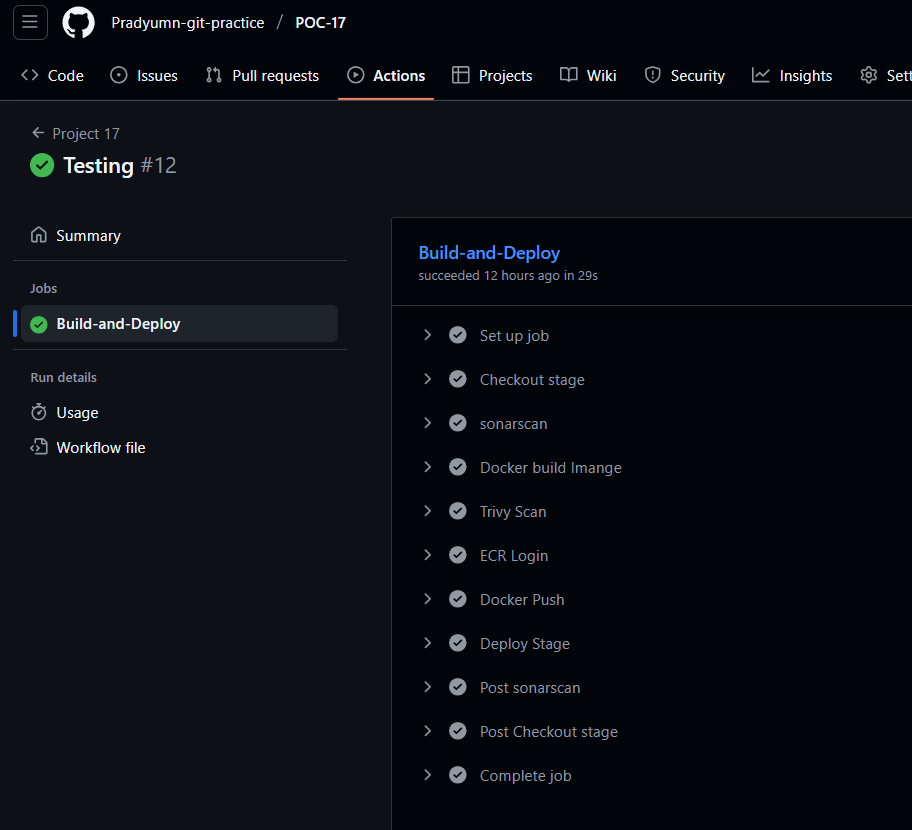
1. Create Kubernetes manifests for Deployment and Service.
2. Verify pods are running and services are accessible.

**Step 8: Sonar Qube**

****

**Step 9: CI/CD Workflow Execution**

1. Commit and push code to GitHub.
2. Verify workflow triggers automatically.
3. Check logs for each stage: Docker build, scan, SonarQube, and deployment.



1. **Outcome**

****

* Fully automated CI/CD pipeline.
* Application successfully deployed on Amazon EKS.
* Vulnerabilities checked using Trivy.
* Code quality ensured with SonarQube.
* Notifications delivered to Slack.