**Jenkins Dashboard**A screenshot of a computer

AI-generated content may be incorrect.

**Integration pipeline script***pipeline {*

*agent any*

*parameters {*

*string(name: 'ECR\_REPO\_NAME', defaultValue: 'amazon-prime', description: 'Enter repository name')*

*string(name: 'AWS\_ACCOUNT\_ID', defaultValue: '123456789012', description: 'Enter AWS Account ID') // Added missing quote*

*}*

*tools {*

*jdk 'JDK'*

*nodejs 'NodeJS'*

*}*

*environment {*

*SCANNER\_HOME = tool 'SonarQube Scanner'*

*}*

*stages {*

*stage('1. Git Checkout') {*

*steps {*

*git branch: 'main', url: 'https://github.com/pandacloud1/DevopsProject2.git'*

*}*

*}*

*stage('2. SonarQube Analysis') {*

*steps {*

*withSonarQubeEnv ('sonar-server') {*

*sh """*

*$SCANNER\_HOME/bin/sonar-scanner \*

*-Dsonar.projectName=amazon-prime \*

*-Dsonar.projectKey=amazon-prime*

*"""*

*}*

*}*

*}*

*stage('3. Quality Gate') {*

*steps {*

*waitForQualityGate abortPipeline: false,*

*credentialsId: 'sonar-token'*

*}*

*}*

*stage('4. Install npm') {*

*steps {*

*sh "npm install"*

*}*

*}*

*stage('5. Trivy Scan') {*

*steps {*

*sh "trivy fs . > trivy.txt"*

*}*

*}*

*stage('6. Build Docker Image') {*

*steps {*

*sh "docker build -t ${params.ECR\_REPO\_NAME} ."*

*}*

*}*

*stage('7. Create ECR repo') {*

*steps {*

*withCredentials([string(credentialsId: 'access-key', variable: 'AWS\_ACCESS\_KEY'),*

*string(credentialsId: 'secret-key', variable: 'AWS\_SECRET\_KEY')]) {*

*sh """*

*aws configure set aws\_access\_key\_id $AWS\_ACCESS\_KEY*

*aws configure set aws\_secret\_access\_key $AWS\_SECRET\_KEY*

*aws ecr describe-repositories --repository-names ${params.ECR\_REPO\_NAME} --region us-east-1 || \*

*aws ecr create-repository --repository-name ${params.ECR\_REPO\_NAME} --region us-east-1*

*"""*

*}*

*}*

*}*

*stage('8. Login to ECR & tag image') {*

*steps {*

*withCredentials([string(credentialsId: 'access-key', variable: 'AWS\_ACCESS\_KEY'),*

*string(credentialsId: 'secret-key', variable: 'AWS\_SECRET\_KEY')]) {*

*sh """*

*aws ecr get-login-password --region us-east-1 | docker login --username AWS --password-stdin ${params.AWS\_ACCOUNT\_ID}.dkr.ecr.us-east-1.amazonaws.com*

*docker tag ${params.ECR\_REPO\_NAME} ${params.AWS\_ACCOUNT\_ID}.dkr.ecr.us-east-1.amazonaws.com/${params.ECR\_REPO\_NAME}:${BUILD\_NUMBER}*

*docker tag ${params.ECR\_REPO\_NAME} ${params.AWS\_ACCOUNT\_ID}.dkr.ecr.us-east-1.amazonaws.com/${params.ECR\_REPO\_NAME}:latest*

*"""*

*}*

*}*

*}*

*stage('9. Push image to ECR') {*

*steps {*

*withCredentials([string(credentialsId: 'access-key', variable: 'AWS\_ACCESS\_KEY'),*

*string(credentialsId: 'secret-key', variable: 'AWS\_SECRET\_KEY')]) {*

*sh """*

*docker push ${params.AWS\_ACCOUNT\_ID}.dkr.ecr.us-east-1.amazonaws.com/${params.ECR\_REPO\_NAME}:${BUILD\_NUMBER}*

*docker push ${params.AWS\_ACCOUNT\_ID}.dkr.ecr.us-east-1.amazonaws.com/${params.ECR\_REPO\_NAME}:latest*

*"""*

*}*

*}*

*}*

*stage('10. Cleanup Images') {*

*steps {*

*sh """*

*docker rmi ${params.AWS\_ACCOUNT\_ID}.dkr.ecr.us-east-1.amazonaws.com/${params.ECR\_REPO\_NAME}:${BUILD\_NUMBER}*

*docker rmi ${params.AWS\_ACCOUNT\_ID}.dkr.ecr.us-east-1.amazonaws.com/${params.ECR\_REPO\_NAME}:latest*

*docker images*

*"""*

*}*

*}*

*}*

*}*

**Integration pipeline**  
A screenshot of a computer

AI-generated content may be incorrect.

**Deployment pipeline script**  
  
*pipeline {*

*agent any*

*environment {*

*KUBECTL = '/usr/local/bin/kubectl'*

*}*

*parameters {*

*string(name: 'CLUSTER\_NAME', defaultValue: 'amazon-prime-cluster', description: 'Enter your EKS cluster name')*

*}*

*stages {*

*stage("Login to EKS") {*

*steps {*

*script {*

*withCredentials([string(credentialsId: 'access-key', variable: 'AWS\_ACCESS\_KEY'),*

*string(credentialsId: 'secret-key', variable: 'AWS\_SECRET\_KEY')]) {*

*sh "aws eks --region us-east-1 update-kubeconfig --name ${params.CLUSTER\_NAME}"*

*}*

*}*

*}*

*}*

*stage("Configure Prometheus & Grafana") {*

*steps {*

*script {*

*sh """*

*helm repo add stable https://charts.helm.sh/stable || true*

*helm repo add prometheus-community https://prometheus-community.github.io/helm-charts || true*

*# Check if namespace 'prometheus' exists*

*if kubectl get namespace prometheus > /dev/null 2>&1; then*

*# If namespace exists, upgrade the Helm release*

*helm upgrade stable prometheus-community/kube-prometheus-stack -n prometheus*

*else*

*# If namespace does not exist, create it and install Helm release*

*kubectl create namespace prometheus*

*helm install stable prometheus-community/kube-prometheus-stack -n prometheus*

*fi*

*kubectl patch svc stable-kube-prometheus-sta-prometheus -n prometheus -p '{"spec": {"type": "LoadBalancer"}}'*

*kubectl patch svc stable-grafana -n prometheus -p '{"spec": {"type": "LoadBalancer"}}'*

*"""*

*}*

*}*

*}*

*stage("Configure ArgoCD") {*

*steps {*

*script {*

*sh """*

*# Install ArgoCD*

*kubectl create namespace argocd || true*

*kubectl apply -n argocd -f https://raw.githubusercontent.com/argoproj/argo-cd/stable/manifests/install.yaml*

*kubectl patch svc argocd-server -n argocd -p '{"spec": {"type": "LoadBalancer"}}'*

*"""*

*}*

*}*

*}*

*}*

*}*  
**Jenkins Deployment pipeline**  
A screenshot of a computer

AI-generated content may be incorrect.

**AWS EC2 instances**  
A screenshot of a computer

AI-generated content may be incorrect.

**Repositories**  
A screenshot of a computer

AI-generated content may be incorrect.  
  
  
  
**Amazon Elastic Container Registry (ECR)**

A screenshot of a computer

AI-generated content may be incorrect.

**SonarQube**  
A screenshot of a computer

AI-generated content may be incorrect.

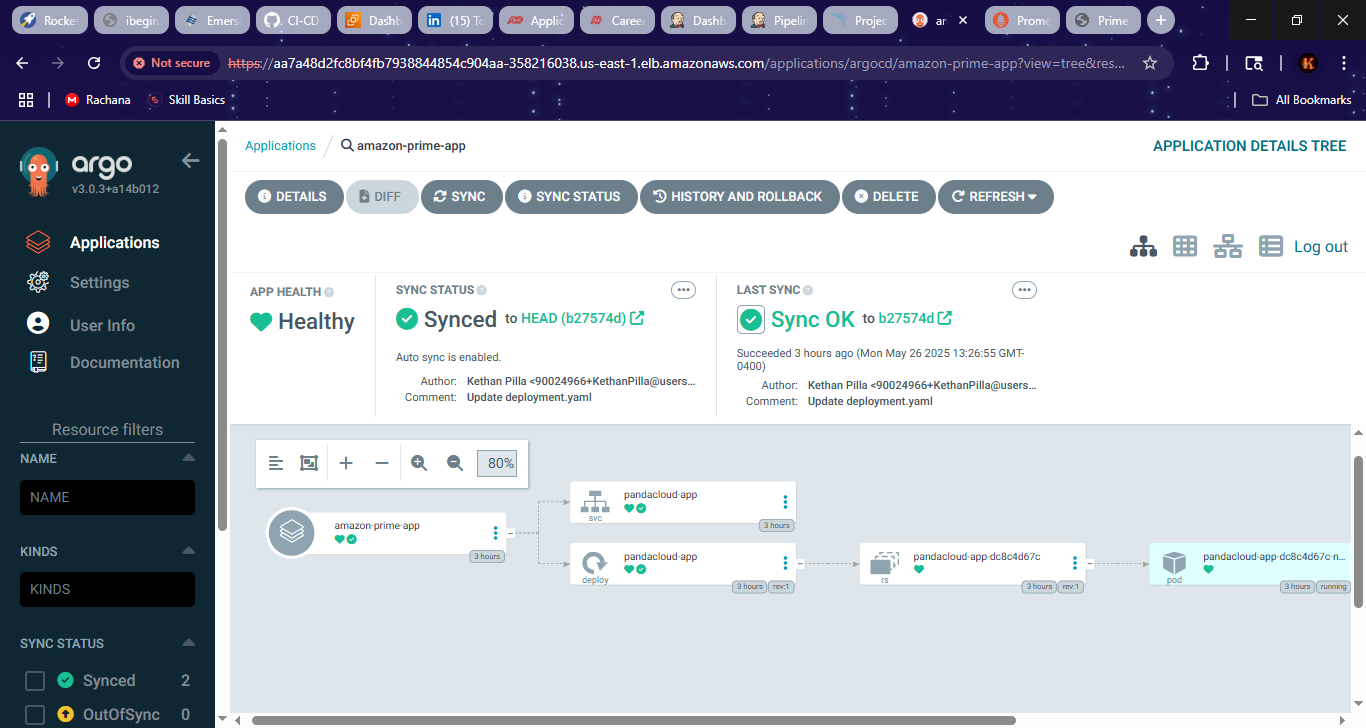
**Terraform**

A screenshot of a computer program

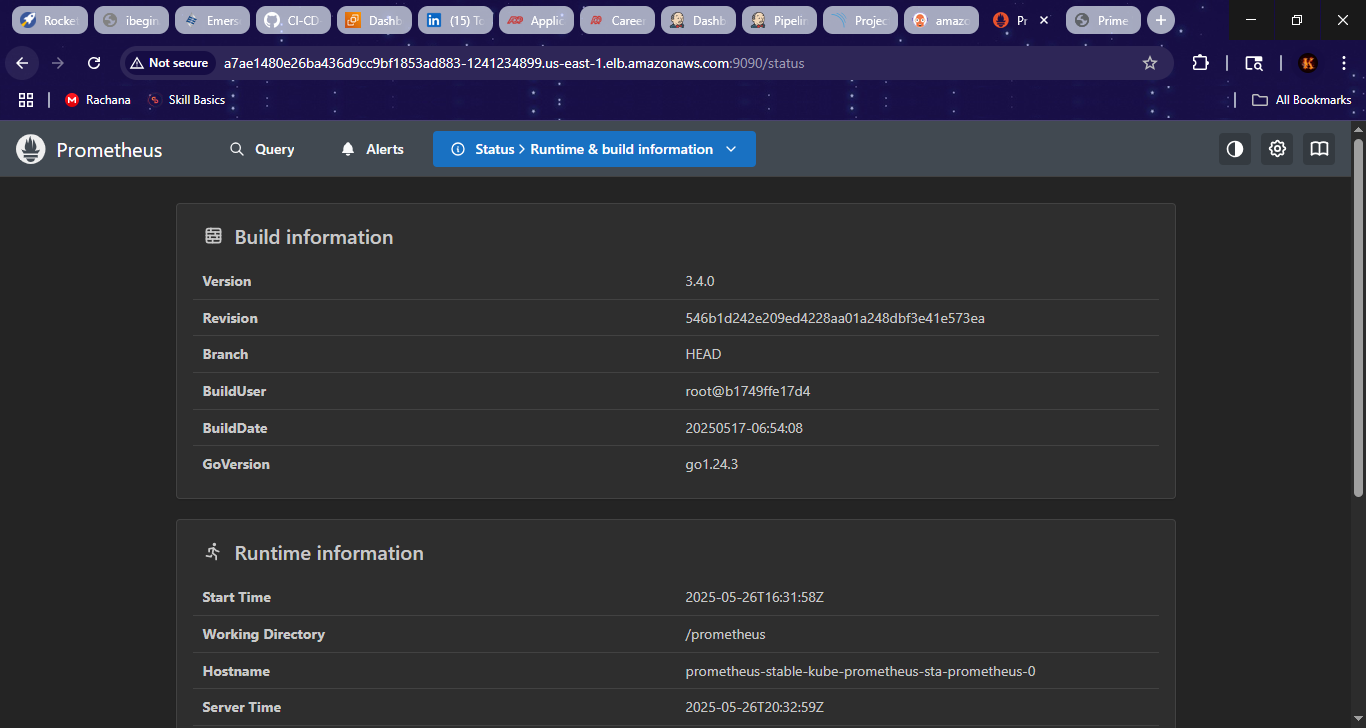
AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

**Argo application**

**Prometheus**



**Grafana dashboard**

A screenshot of a computer

AI-generated content may be incorrect.

**Deployed app**A screenshot of a computer

AI-generated content may be incorrect.