pipeline {

agent any

tools {

jdk 'jdk20'

nodejs 'nodeJS'

}

environment {

SCANNER\_HOME = tool 'sonar-scanner'

GIT\_REPO\_NAME = "TetrisGameProject-K8smanifest"

GIT\_USER\_NAME = "Sravyatirumala"

}

stages {

stage('Clean Workspace') {

steps {

cleanWs()

}

}

stage('Checkout from Git') {

steps {

git branch: 'main', url: 'https://github.com/Sravyatirumala/TetrisGameProject-V1.git'

}

}

stage("Sonarqube Analysis") {

steps {

withSonarQubeEnv('sonar-server') {

sh '''$SCANNER\_HOME/bin/sonar-scanner -Dsonar.projectName=TetrisVersion1.0 \

-Dsonar.projectKey=TetrisVersion1.0'''

}

}

}

stage("Quality Gate") {

steps {

script {

waitForQualityGate abortPipeline: false, credentialsId: 'Sonar-token'

}

}

}

stage('Install Dependencies') {

steps {

sh "npm install"

}

}

// stage('OWASP FS SCAN') {

// steps {

// dependencyCheck additionalArguments: '--scan ./ --disableYarnAudit --disableNodeAudit', odcInstallation: 'DP-Check'

// dependencyCheckPublisher pattern: '\*\*/dependency-check-report.xml'

// }

// }

stage('TRIVY FS SCAN') {

steps {

sh "trivy fs . > trivyfs.txt"

}

}

stage("Docker Build & Push") {

steps {

script {

withDockerRegistry(credentialsId: 'Docker-creds', toolName: 'docker') {

sh "docker build -t tetris ."

sh "docker tag tetris sravyatirumala/tetris:latest"

sh "docker push sravyatirumala/tetris:latest"

}

}

}

}

stage("TRIVY Image Scan") {

steps {

sh "trivy image sravyatirumala/tetris:latest > trivyimage.txt"

}

}

stage("Deploy to Container") {

steps {

sh 'docker run -d --name Tetrisgame -p 3000:3000 sravyatirumala/tetris:latest'

}

}

stage('GitManifest') {

steps {

git branch: 'main', url: 'https://github.com/Sravyatirumala/TetrisGameProject-K8smanifest.git'

}

}

stage('Update Deployment File') {

steps {

script {

withCredentials([string(credentialsId: 'github-token', variable: 'github\_token')]) {

def NEW\_IMAGE\_NAME = "sravyatirumala/tetris:latest"

// Set Git author identity

sh "git config user.email 'tvsravya95@gmail.com'"

sh "git config user.name 'sravyatirumala'"

// Update image in deployment.yml and commit the change

sh "sed -i 's|image: .\*|image: ${NEW\_IMAGE\_NAME}|' deployment.yml"

sh "git add deployment.yml"

sh "git commit -m 'Update deployment image to ${NEW\_IMAGE\_NAME}'"

sh "git push https://${github\_token}@github.com/${GIT\_USER\_NAME}/${GIT\_REPO\_NAME} HEAD:main"

}

}

}

}

}

}

………………………………………………………

………………………………………………………

Install argocd and patch to load balancer.

kubectl create ns argocd

kubectl apply -n argocd -f https://raw.githubusercontent.com/argoproj/argo-cd/v2.4.7/manifests/install.yaml

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

This will create a load balancer.

………………………………………………………

………………………………………………………

sudo apt install jq -y

export ARGOCD\_SERVER=`kubectl get svc argocd-server -n argocd -o json | jq --raw-output '.status.loadBalancer.ingress[0].hostname'`

echo $ARGOCD\_SERVER. ( This will give load balancer name for argocd)

Login argocd:

export ARGO\_PWD=`kubectl -n argocd get secret argocd-initial-admin-secret -o jsonpath="{.data.password}" | base64 -d`

Password:

echo $ARGO\_PWD

In argocd give path as [ ./ ] as there is no folder type in file.

https://github.com/Sravyatirumala/TetrisGameProject-V2.git