JENKINS PIPELINE:  
pipeline {

agent any

tools {

jdk 'jdk-17'

nodejs 'node16'

}

environment {

SCANNER\_HOME = tool 'sonar-scanner'

}

stages {

stage('CW') {

steps {

cleanWs()

}

}

stage('Git CheckOut') {

steps {

git branch: 'main', url: 'https://github.com/Aj7Ay/Amazon-FE.git'

}

}

stage('Sonar Analysis') {

steps {

script {

withSonarQubeEnv('sonar-server') {

sh '''

$SCANNER\_HOME/bin/sonar-scanner -Dsonar.projectName=Amazon-FE \

-Dsonar.projectKey=Amazon-FE

'''

}

}

}

}

stage('Quality Gate') {

steps {

script {

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

}

}

}

stage('Install NPM') {

steps {

sh 'npm install'

}

}

stage('OWASP') {

steps {

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

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

}

}

stage('Trivy Filesystem Scan') {

steps {

sh "trivy fs . > trivyfs.txt"

}

}

stage('Docker Build & Push') {

steps {

script {

withDockerRegistry(credentialsId: 'docker', toolName: 'docker') {

sh '''

docker build -t amazon .

docker tag amazon balasuriya/amazon:latest

docker push balasuriya/amazon:latest

'''

}

}

}

}

stage('Trivy Image Scan') {

steps {

sh "trivy image balasuriya/amazon:latest > trivyimage.txt"

}

}

stage('Deploy to Container') {

steps {

sh 'docker run -d --name amazon -p 3000:3000 balasuriya/amazon:latest'

}

}

}

}