```
pipeline{
    agent any
    tools {
        jdk 'jdk17'
        maven 'maven3'
    }
    environment {
        SCANNER_HOME=tool 'sonar-scanner'
    }
    stages{
        stage ('clean Workspace'){
            steps{
                cleanWs()
            }
        }
        stage ('checkout scm') {
            steps {
                git 'https://github.com/Aj7Ay/jpetstore-6.git'
            }
        }
        stage ('maven compile') {
            steps {
                sh 'mvn clean compile'
            }
        }
        stage ('maven Test') {
            steps {
                sh 'mvn test'
            }
        }
        stage("Sonarqube Analysis "){
            steps{
                withSonarQubeEnv('sonar-server') {
                    sh ''' $SCANNER_HOME/bin/sonar-scanner
-Dsonar.projectName=Petshop \
                    -Dsonar.java.binaries=. \
                    -Dsonar.projectKey=Petshop '''
                }
            }
        }
        stage("quality gate"){
            steps {
                script {
                  waitForQualityGate abortPipeline: false, credentialsId:
'Sonar-token'
                }
           }
        }
        stage ('Build war file'){
            steps{
```

```
                sh 'mvn clean install -DskipTests=true'
            }
        }
        stage("OWASP Dependency Check"){
            steps{
                dependencyCheck additionalArguments: '--scan ./ --format XML ',
odcInstallation: 'DP-Check'
                dependencyCheckPublisher pattern: '**/dependency-check-report.xml'
            }
        }
        stage ('Build and push to docker hub'){
            steps{
                script{
                    withDockerRegistry(credentialsId: 'docker', toolName: 'docker')
{
                        sh "docker build -t petshop ."
                        sh "docker tag petshop sevenajay/petshop:latest"
                        sh "docker push sevenajay/petshop:latest"
                   }
                }
            }
        }
        stage("TRIVY"){
            steps{
                sh "trivy image sevenajay/petshop:latest > trivy.txt"
            }
        }
        stage ('Deploy to container'){
            steps{
                sh 'docker run -d --name pet1 -p 8080:8080 sevenajay/petshop:latest'
            }
        }
        stage('K8s'){
            steps{
                script{
                    withKubeConfig(caCertificate: '', clusterName: '', contextName:
'', credentialsId: 'k8s', namespace: '', restrictKubeConfigAccess: false, serverUrl:
'') {
                        sh 'kubectl apply -f deployment.yaml'
                    }
                }
            }
        }
    }
    post {
     always {
        emailext attachLog: true,
            subject: "'${currentBuild.result}'",
            body: "Project: ${env.JOB_NAME}<br/>" +
                "Build Number: ${env.BUILD_NUMBER}<br/>" +
```

```
                "URL: ${env.BUILD_URL}<br/>",
            to: 'postbox.aj99@gmail.com',
            attachmentsPattern: 'trivy.txt'
        }
    }
}
```