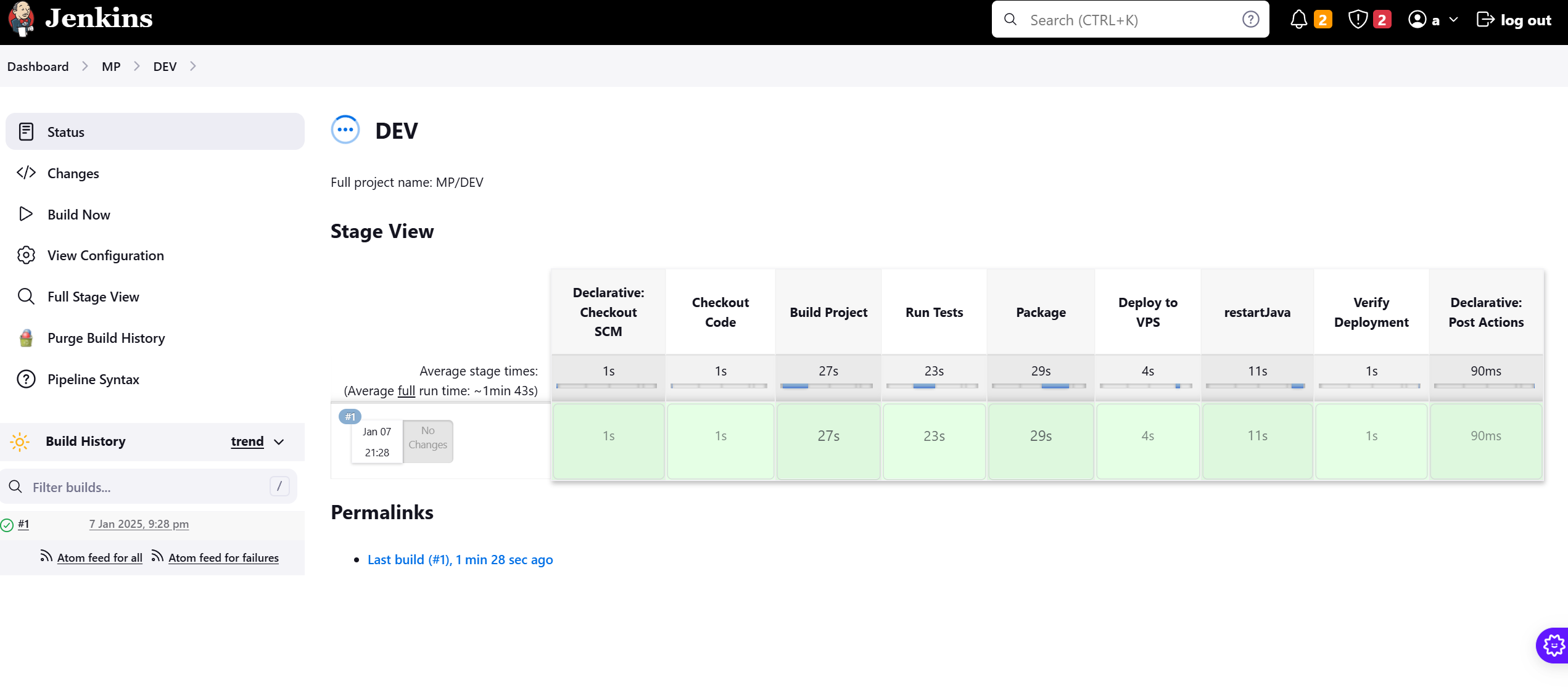
图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成



图形用户界面, 应用程序, 电子邮件

描述已自动生成

图形用户界面, 文本, 应用程序

描述已自动生成

|  |
| --- |
|  |

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

图形用户界面, 文本, 应用程序, 电子邮件, Teams

描述已自动生成

图形用户界面, 文本, 应用程序

描述已自动生成

Jenkinsfile

|  |
| --- |
| pipeline {  agent any  *// tools { // maven 'maven3.9.9' // }*  environment {  *// Define environment variables // JAVA\_HOME = '/opt/java/openjdk' // Adjust if necessary  JAVA\_HOME* = '/opt/jdk' *// Adjust if necessary  REMOTE\_SERVER* = 'root@5.104.80.4' *// SSH user and VPS IP address  DEPLOY\_DIR* = '/opt/jar' *// Directory on your VPS to deploy to  JAR\_NAME* = 'demo-SpringBoot-SwaggerV3' *// Directory on your VPS to deploy to  CURRENT\_TIME* = "${new Date().format('yyyy-MM-dd HH:mm:ss')}"  CURRENT\_TIMVvVV = "${new Date().format('yyyy-MM-dd-HH-mm-ss')}"  *ENVENV*='dev1'  }   stages {  stage('Checkout Code') {  steps {  *// Clone the Git repository*  git 'https://github.com/dsl1888888/demo-SpringBoot-SwaggerV3.git'  }  }   stage('Build Project') {  steps {  script {  *// Run Maven to build the Spring Boot project*  sh '''#!/bin/bash  mvn clean install -DskipTests  '''  }  }  }   stage('Run Tests') {  steps {  script {  *// Optionally, run tests using Maven*  sh 'mvn test'  }  }  }   stage('Package') {  steps {  script {  *// Package the application (Spring Boot app as .jar or .war)*  sh 'mvn package'  }  }  }   stage('Deploy to VPS') {  steps {  script {  *// Copy the packaged application to the VPS using SCP*  sh '''  #!/bin/bash  scp -P 22 target/${JAR\_NAME}.jar ${REMOTE\_SERVER}:${DEPLOY\_DIR}/${JAR\_NAME}-${ENVENV}-${CURRENT\_TIMVvVV}-${BUILD\_NUMBER}.jar  '''  }  }  }   stage('restartJava') {  steps {  script {  sh '''  ssh -p 22 root@5.104.80.4 << EOF || true  java -version  pkill -f 'demo-SpringBoot-SwaggerV3-${ENVENV}'  nohup java -jar ${DEPLOY\_DIR}/${JAR\_NAME}-${ENVENV}-${CURRENT\_TIMVvVV}-${BUILD\_NUMBER}.jar --server.port=17001 > /dev/null 2>&1 &  sleep 10  ps aux | grep demo-SpringBoot-SwaggerV3.jar  EOF  '''  }  }  }   stage('Verify Deployment') {  steps {  script {  *// Optionally, verify if the app is running by checking the health endpoint (or any endpoint)*  echo "Verifying deployment..."  sh '''  curl -f http://5.104.80.4:17001/a2 || exit 1  '''  }  }  }       }   post {  success {  echo 'Deployment Successful!'  }  failure {  echo 'Build or Deployment Failed.'  }  } } |

Jenkinsfile1

|  |
| --- |
| pipeline {  agent any  *// tools { // maven 'maven3.9.9' // }*  environment {  *// Define environment variables // JAVA\_HOME = '/opt/java/openjdk' // Adjust if necessary  JAVA\_HOME* = '/opt/jdk' *// Adjust if necessary  REMOTE\_SERVER* = 'root@5.104.80.4' *// SSH user and VPS IP address  DEPLOY\_DIR* = '/opt/jar' *// Directory on your VPS to deploy to  JAR\_NAME* = 'demo-SpringBoot-SwaggerV3' *// Directory on your VPS to deploy to  CURRENT\_TIME* = "${new Date().format('yyyy-MM-dd HH:mm:ss')}"  CURRENT\_TIMVvVV = "${new Date().format('yyyy-MM-dd-HH-mm-ss')}"  *ENVENV*='TEST'  }   stages {  stage('Checkout Code') {  steps {  *// Clone the Git repository*  git 'https://github.com/dsl1888888/demo-SpringBoot-SwaggerV3.git'  }  }   stage('Build Project') {  steps {  script {  *// Run Maven to build the Spring Boot project*  sh '''#!/bin/bash  mvn clean install -DskipTests  '''  }  }  }   stage('Run Tests') {  steps {  script {  *// Optionally, run tests using Maven*  sh 'mvn test'  }  }  }   stage('Package') {  steps {  script {  *// Package the application (Spring Boot app as .jar or .war)*  sh 'mvn package'  }  }  }   stage('Deploy to VPS') {  steps {  script {  *// Copy the packaged application to the VPS using SCP*  sh '''  #!/bin/bash  scp -P 22 target/${JAR\_NAME}.jar ${REMOTE\_SERVER}:${DEPLOY\_DIR}/${JAR\_NAME}-${ENVENV}-${CURRENT\_TIMVvVV}-${BUILD\_NUMBER}.jar  '''  }  }  }   stage('restartJava') {  steps {  script {  sh '''  ssh -p 22 root@5.104.80.4 << EOF || true  java -version  pkill -f 'demo-SpringBoot-SwaggerV3-${ENVENV}'  nohup java -jar ${DEPLOY\_DIR}/${JAR\_NAME}-${ENVENV}-${CURRENT\_TIMVvVV}-${BUILD\_NUMBER}.jar --server.port=17002 > /dev/null 2>&1 &  sleep 10  ps aux | grep demo-SpringBoot-SwaggerV3.jar  EOF  '''  }  }  }   stage('Verify Deployment') {  steps {  script {  *// Optionally, verify if the app is running by checking the health endpoint (or any endpoint)*  echo "Verifying deployment..."  sh '''  curl -f http://5.104.80.4:17002/a2 || exit 1  '''  }  }  }       }   post {  success {  echo 'Deployment Successful!'  }  failure {  echo 'Build or Deployment Failed.'  }  } } |

Jenkins example on docker

|  |
| --- |
| pipeline {  agent any  tools {  maven 'maven3.9.9' // Updated to the configured Maven version  git 'Default' // Updated to the configured Git version  }  environment {  // Define environment variables  JAVA\_HOME = '/opt/java/openjdk' // Adjust if necessary  REMOTE\_SERVER = 'root@5.104.80.4' // SSH user and VPS IP address  DEPLOY\_DIR = '/opt/jar' // Directory on your VPS to deploy to  JAR\_NAME = 'demo-SpringBoot-SwaggerV3' // Directory on your VPS to deploy to  CURRENT\_TIME = "${new Date().format('yyyy-MM-dd HH:mm:ss')}"  CURRENT\_TIMVvVV = "${new Date().format('yyyy-MM-dd-HH-mm-ss')}"  ENVENV='DEV-DOCKER'  }  stages {  stage('Checkout Code') {  steps {  // Clone the Git repository  git 'https://github.com/dsl1888888/demo-SpringBoot-SwaggerV3.git'  }  }  stage('Build Project') {  steps {  script {  // Run Maven to build the Spring Boot project  sh '''#!/bin/bash  mvn clean install -DskipTests  '''  }  }  }  stage('Run Tests') {  steps {  script {  // Optionally, run tests using Maven  sh 'mvn test'  }  }  }  stage('Package') {  steps {  script {  // Package the application (Spring Boot app as .jar or .war)  sh 'mvn package'  }  }  }  stage('Deploy to VPS') {  steps {  script {  // Copy the packaged application to the VPS using SCP  sh '''  #!/bin/bash  scp -P 22 target/${JAR\_NAME}.jar ${REMOTE\_SERVER}:${DEPLOY\_DIR}/${JAR\_NAME}-${ENVENV}-${CURRENT\_TIMVvVV}-${BUILD\_NUMBER}.jar  '''  }  }  }  stage('restartJava') {  steps {  script {  sh '''  ssh -p 22 root@5.104.80.4 << EOF || true  java -version  pkill -f 'demo-SpringBoot-SwaggerV3-${ENVENV}'  nohup java -jar ${DEPLOY\_DIR}/${JAR\_NAME}-${ENVENV}-${CURRENT\_TIMVvVV}-${BUILD\_NUMBER}.jar --server.port=17003 > /dev/null 2>&1 &  sleep 10  ps aux | grep demo-SpringBoot-SwaggerV3.jar  EOF  '''  }  }  }  stage('Verify Deployment') {  steps {  script {  // Optionally, verify if the app is running by checking the health endpoint (or any endpoint)  echo "Verifying deployment..."  sh '''  sleep 10  curl -f http://5.104.80.4:17003/a2 || exit 1  '''  }  }  }  }  post {  always {  // Archive the build artifacts  archiveArtifacts artifacts: '\*\*/target/\*.jar', allowEmptyArchive: true  }  success {  echo 'Deployment Successful!'  }  failure {  echo 'Build or Deployment Failed.'  }  }  } |