Jenkins Pipeline Script

```groovy
pipeline {

  agent {

    label 'dev'

  }

  tools {

    maven 'mymaven'

    jdk 'jdk17'

    nodejs 'node16'

  }

  stages {

    stage('clean work space') {

      steps {

        cleanWs()

      }

    }


    stage('Checkout Code') {

      steps {

        git 'https://github.com/PrathimaVyas/dockerwebapp.git'

      }

    }


    stage('code quality Analysis') {

      steps {
```

```
            withSonarQubeEnv('mysonar') {

                sh 'mvn clean verify sonar:sonar -Dsonar.projectKey=zomato'

            }

        }

    }


    stage('Quality Gates') {

        steps {

            script {

                waitForQualityGate abortPipeline: false, credentialsId: 'sonar'

            }

        }

    }


    stage('Build the Code') {

        steps {

            sh 'mvn clean package'

            sh 'mkdir -p Docker-app/target && cp -r target/* Docker-app/target/'

        }

    }


    stage('Deploy war file to Nexus') {

        steps {

            sh 'ls -l'

            sh 'pwd'

                nexusArtifactUploader artifacts: [[artifactId: 'vprofile', classifier: '', file: 'target/vprofilev2.war', type: 'war']],
```
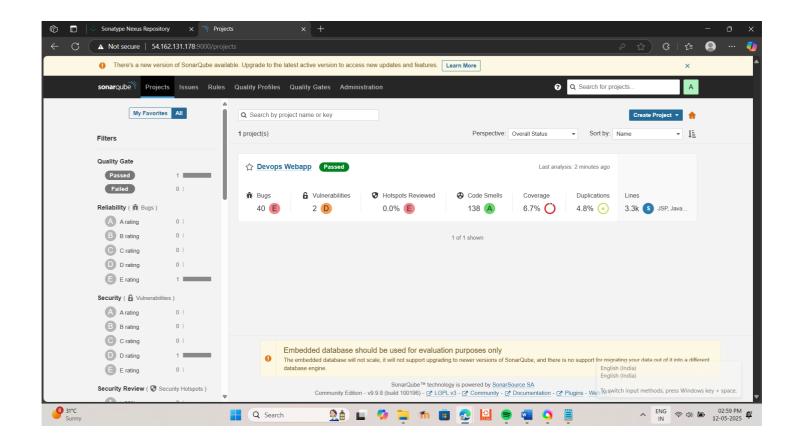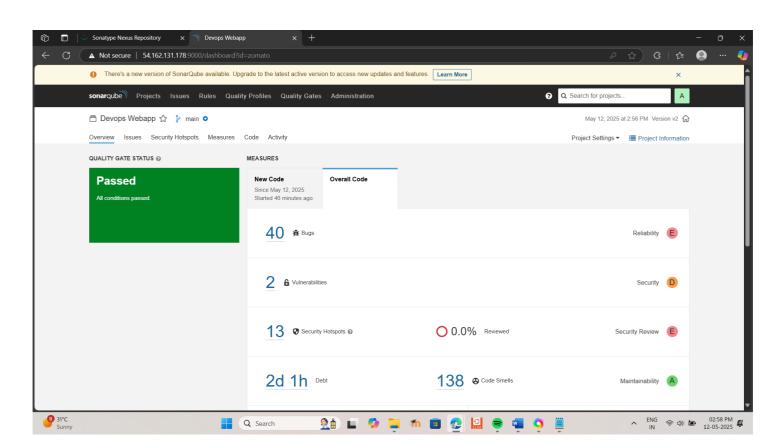
```
credentialsId: 'nexus', groupId: 'com.visualpathit', nexusUrl: '34.227.87.113:8081', nexusVersion: 'nexus3', protocol:
'http', repository: 'DockerProject', version: 'v2'

        }

    }


    stage('create docker image') {

        steps {

            sh 'docker build -t prathima77/docker-app:v1 Docker-app'

            sh 'docker build -t prathima77/docker-db:v1 Docker-db'

        }

    }


    stage('scan Docker image') {

        steps {

            echo 'Scanning Docker images using Trivy...'

            sh 'trivy image prathima77/docker-app:v1'

            sh 'trivy image prathima77/docker-db:v1'

        }

    }


    stage('Push to Docker Hub') {

        steps {

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

                sh 'docker push prathima77/docker-app:v1'

                sh 'docker push prathima77/docker-db:v1'

            }
```
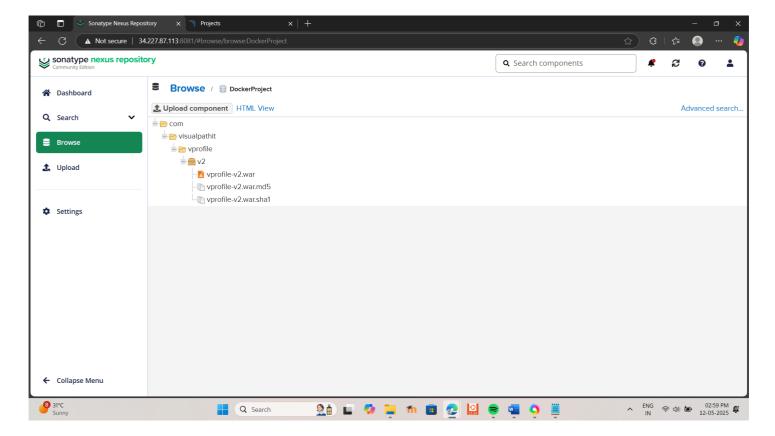
```
        }

    }


    stage('Deploy Docker Stack') {

        steps {

            sh 'docker stack deploy newstack --compose-file=compose.yml'

        }

    }

}

post {

    always {

        slackSend channel: 'all-docker-project',

                message: "*${currentBuild.currentResult}:* Job ${env.JOB_NAME}

Build ${env.BUILD_NUMBER}

More info at: ${env.BUILD_URL}"

    }

}

}
```
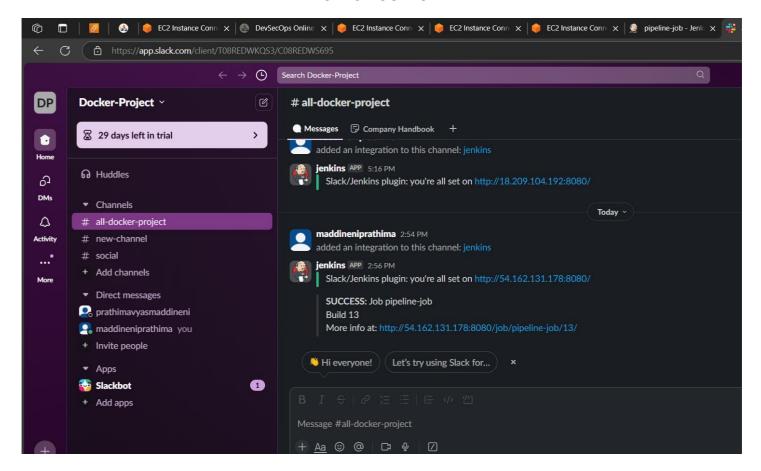
SONARQUBE OUTPUT

NEXUS OUTPUT

SLACK OUTPUT

PIPELINE STAGE VIEW
(WE CAN SEE ALL STAGES OUTPUTS IN LOGS HERE)