

# Spring Boot Web Application – Automated CI/CD Pipeline

This project contains a Spring Boot (Maven) web application with a fully automated CI/CD pipeline using Jenkins, Docker, SonarQube, and Trivy. The deployment is done on an AWS EC2 instance running Docker.

## Pipeline Stage view

### Stage View

	Clone Repository	Maven Build	SonarQube Analysis	Docker Build	Trivy Security Scan	Push to Docker Hub	Deploy on Server	Declarative: Post Actions
Average stage times: (full run time: ~52s)	640ms	7s	10s	3s	6s	17s	3s	1s
#22 12:23 No Changes	616ms	7s	9s	3s	6s	17s	3s	4s

## Project Overview

This project demonstrates a DevOps workflow for automating the deployment of a Spring Boot application using:

- ✓ Jenkins – CI/CD automation
- ✓ Maven – Build management
- ✓ SonarQube – Code quality analysis
- ✓ Docker – Containerization
- ✓ Trivy – Security vulnerability scanning
- ✓ AWS EC2 – Deployment server

## CI/CD Pipeline Workflow

- ❖ Git Checkout – Pulls the latest code from GitHub.
- ❖ Maven Build – Compiles and packages the application.
- ❖ SonarQube Analysis – Scans for bugs, security issues, and code smells.
- ❖ Docker Build – Creates a containerized application.
- ❖ Trivy Scan – Checks for security vulnerabilities.
- ❖ Push to Docker Registry – Uploads the image to Docker Hub / AWS ECR.
- ❖ Deploy to AWS EC2 – Pulls the latest image, stops the old container, and runs the new one.
- ❖ Email Notifications – Sends success/failure alerts.

## **Setup & Installation**

### **Prerequisites**

Ensure you have the following installed:

- JDK 17+
- Maven
- Docker
- Jenkins (with required plugins)
- SonarQube Server
- Trivy Scanner

### **Jenkins Pipeline Script**

```
pipeline {
    agent any

    environment {
        MAVEN_HOME = "/opt/maven"
        PATH = "$PATH:$MAVEN_HOME/bin"
        SONARQUBE_URL = 'http://<ip-address or URL>:9000'
        SONARQUBE_CREDENTIALS = '<sonarqube-secret-id>'
        DOCKER_IMAGE = 'suryaprakash6027/devops'
        TRIVY_EXIT_CODE = '0'
    }

    stages {
        stage('Clone Repository') {
            steps {
                git branch: 'main', credentialsId:
<credentials-id>', url: 'https://github.com/suryaprakash-
r/blog-app.git'
            }
        }
    }
}
```

```

    }

    stage('Maven Build') {
        steps {
            sh 'mvn clean package -DskipTests'
        }
    }

    stage('SonarQube Analysis') {
        steps {
            withSonarQubeEnv('SonarQube') {
                sh 'mvn sonar:sonar'
            }
        }
    }

    stage('Docker Build') {
        steps {
            sh 'mv target/*.war target/blog.war'
            sh "docker build -t ${DOCKER_IMAGE}:latest ."
        }
    }

    stage('Trivy Security Scan') {
        steps {
            script {
                def result = sh(script: 'trivy image --
exit-code 1 --severity CRITICAL ${DOCKER_IMAGE}:latest | tee
trivy-report.txt', returnStatus: true)

                archiveArtifacts artifacts: 'trivy-
report.txt', fingerprint: true

                if (result != 0) {

```

```

                                error("Security scan failed! Critical
vulnerabilities found.")
                                }
                                }
                                }
                                }

stage('Push to Docker Hub') {
    steps {
        sh 'docker push ${DOCKER_IMAGE}:latest'
    }
}

//                                Build URL: ${env.BUILD_URL}
stage('Deploy on Server') {
    steps {
        sh 'docker pull ${DOCKER_IMAGE}:latest'
        sh 'docker stop blog-app || true'
        sh 'docker rm blog-app || true'
        sh 'docker run -d --name blog-app -p 8090:8080
--restart unless-stopped ${DOCKER_IMAGE}:latest'
    }
}

}

post {
    always {
        echo "Sending email notification"
    }
    success {
        script {
            echo "Pipeline successful, sending success
email"

```

```

        emailext (
            subject: "✅ SUCCESS: Jenkins Pipeline -
${env.JOB_NAME}",
            body: ""

                🎉 Jenkins Pipeline Execution
Successful! 🎉

                ✅ The Blog Application Docker
container has been successfully deployed! 🚀

            "",
            to: "your-mail@gmail.com"
        )
    }
}

failure {
    script {
        echo "Pipeline failed, sending failure email"
        emailext (
            subject: "❌ FAILURE: Jenkins Pipeline -
${env.JOB_NAME}",
            body: ""

                ⚠️ Jenkins Pipeline Execution Failed!

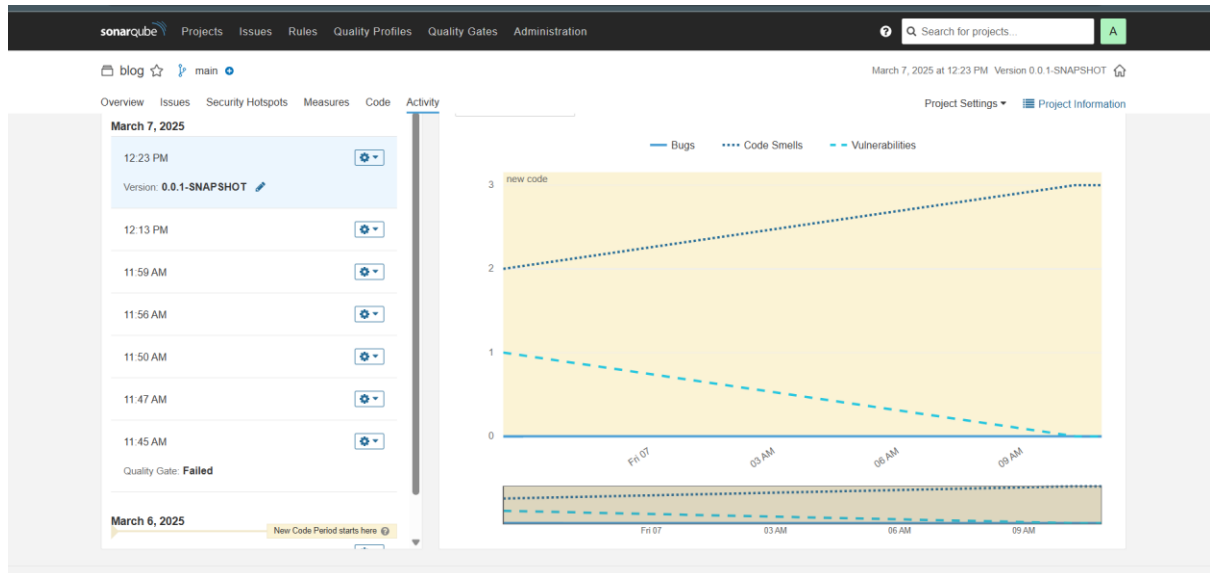
                ! Please check Jenkins logs for
details.

            "",
            to: "your-mail@gmail.com"
        )
    }
}
}
}
}

```

## Sample Screenshots

### SonarQube Analysis



### Mail alerts

