

**< CI – JENKINS PIPELINE AS A CODE >**  
**IMPLEMENTATION PLAN**

VERSION 1.0 | 06/06/2022 – 06/06/2022

## Purpose

This section, describe the Continuous Integration Using Jenkins, Nexus, Sonarqube & Slack.

## Project Scenario

- Agile SDLC.
- Developer makes regular code changes.
- These commits need to be regularly Build and Tested.
- Build and Release Team will do this job.
- Or Developer job to merge and integrate code.

## Problem Statement

- In Agile SDLC, there will be frequent code changes.
- Not so frequently code will be tested, which accumulates bugs and error in the code.
- Developer need to rework to fix these bugs and errors.
- Manual Build & release process.
- Inter Team Dependencies.

## Solution

- Build & test for every commit.
- Automated process.
- Notify for every build status.
- Fix code if bugs or error found instantly rather than waiting.

## Tools

- Jenkins – Continuous Integration Server
- GIT – Version Control System
- Maven – Build Tool
- CheckStyle – Code Analysis Tool
- Slack – Notification
- Nexus – Artifact/Software Repository
- Sonarqube – Code Analysis Server
- AWS – EC2 Resource

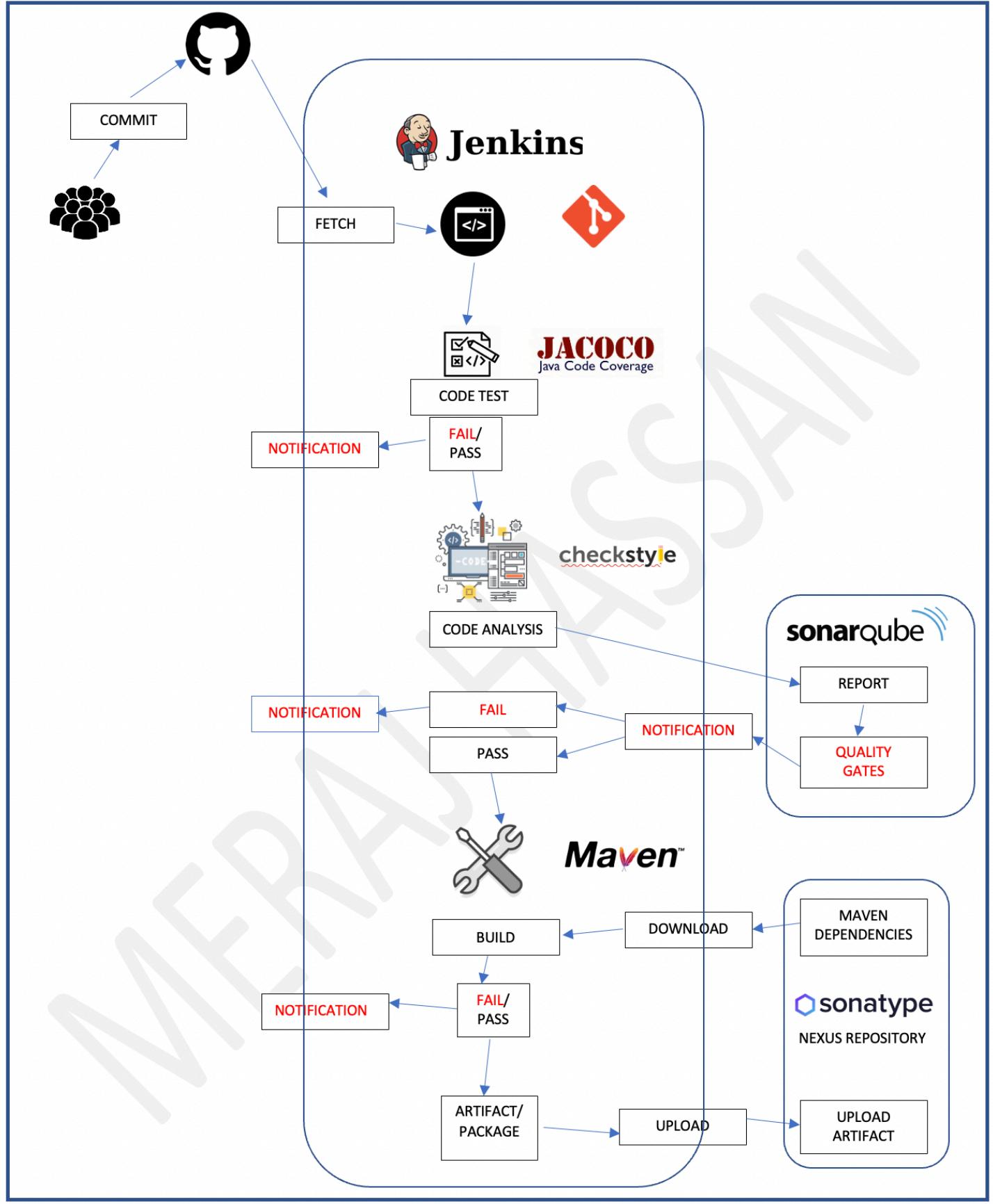
## Objective

- Fault Isolation.
- Short MTTR
- Fast turnaround on feature changes
- Less disruptive

## Architecture of AWS Services

- EC2 Instances

## Architecture Continuous Integration Pipeline



## **Flow of Execution**

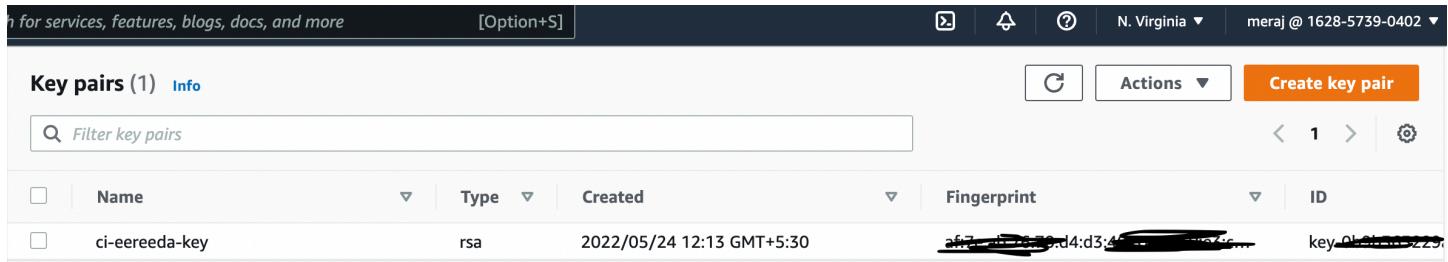
- *Login to AWS Account*
- *Create Login Key*
- *Create Security Group*
  - *Jenkins*
  - *Nexus*
  - *Sonar*
- *Create EC2 instance with user data*
  - *Jenkins*
  - *Sonarqube*
  - *Nexus*
- *Jenkins post installation*
- *Nexus repository setup*
  - *3 Repos*
- *Sonarqube Post Installation*
- *Jenkins Steps – Pipeline as a Code*
- *Clean-up*

## **Prerequisite**

1. *AWS Account*
2. *GITHUB Account*
3. *IntelliJ*

## 1. Security Group & Keypairs

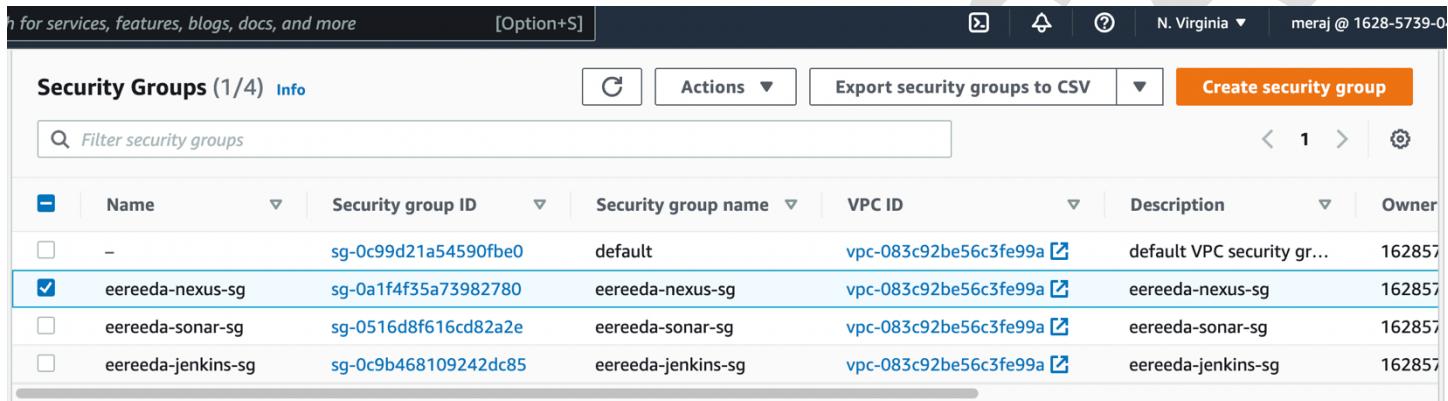
### a. KeyPairs



A screenshot of the AWS Lambda console showing the 'Key pairs' list. The table has columns: Name, Type, Created, Fingerprint, and ID. One row is visible: 'ci-eereeda-key' (rsa, 2022/05/24 12:13 GMT+5:30, fingerprint, key\_ID).

	Name	Type	Created	Fingerprint	ID
<input type="checkbox"/>	ci-eereeda-key	rsa	2022/05/24 12:13 GMT+5:30	[REDACTED]	key_ID[REDACTED]

### b. Security Group



A screenshot of the AWS Lambda console showing the 'Security Groups' list. The table has columns: Name, Security group ID, Security group name, VPC ID, Description, and Owner. One row is checked: 'eereeda-nexus-sg' (sg-0a1f4f35a73982780, default, vpc-083c92be56c3fe99a, eereeda-nexus-sg, 162857).

	Name	Security group ID	Security group name	VPC ID	Description	Owner
<input type="checkbox"/>	-	sg-0c99d21a54590fbe0	default	vpc-083c92be56c3fe99a	default VPC security gr...	162857
<input checked="" type="checkbox"/>	eereeda-nexus-sg	sg-0a1f4f35a73982780	eereeda-nexus-sg	vpc-083c92be56c3fe99a	eereeda-nexus-sg	162857
<input type="checkbox"/>	eereeda-sonar-sg	sg-0516d8f616cd82a2e	eereeda-sonar-sg	vpc-083c92be56c3fe99a	eereeda-sonar-sg	162857
<input type="checkbox"/>	eereeda-jenkins-sg	sg-0c9b468109242dc85	eereeda-jenkins-sg	vpc-083c92be56c3fe99a	eereeda-jenkins-sg	162857

### User Data Script

```
$ git clone https://github.com/merajafnan/DevOps_Projects.git  
$ git checkout CI-Jenkins  
$ cd userdata  
$ ls
```



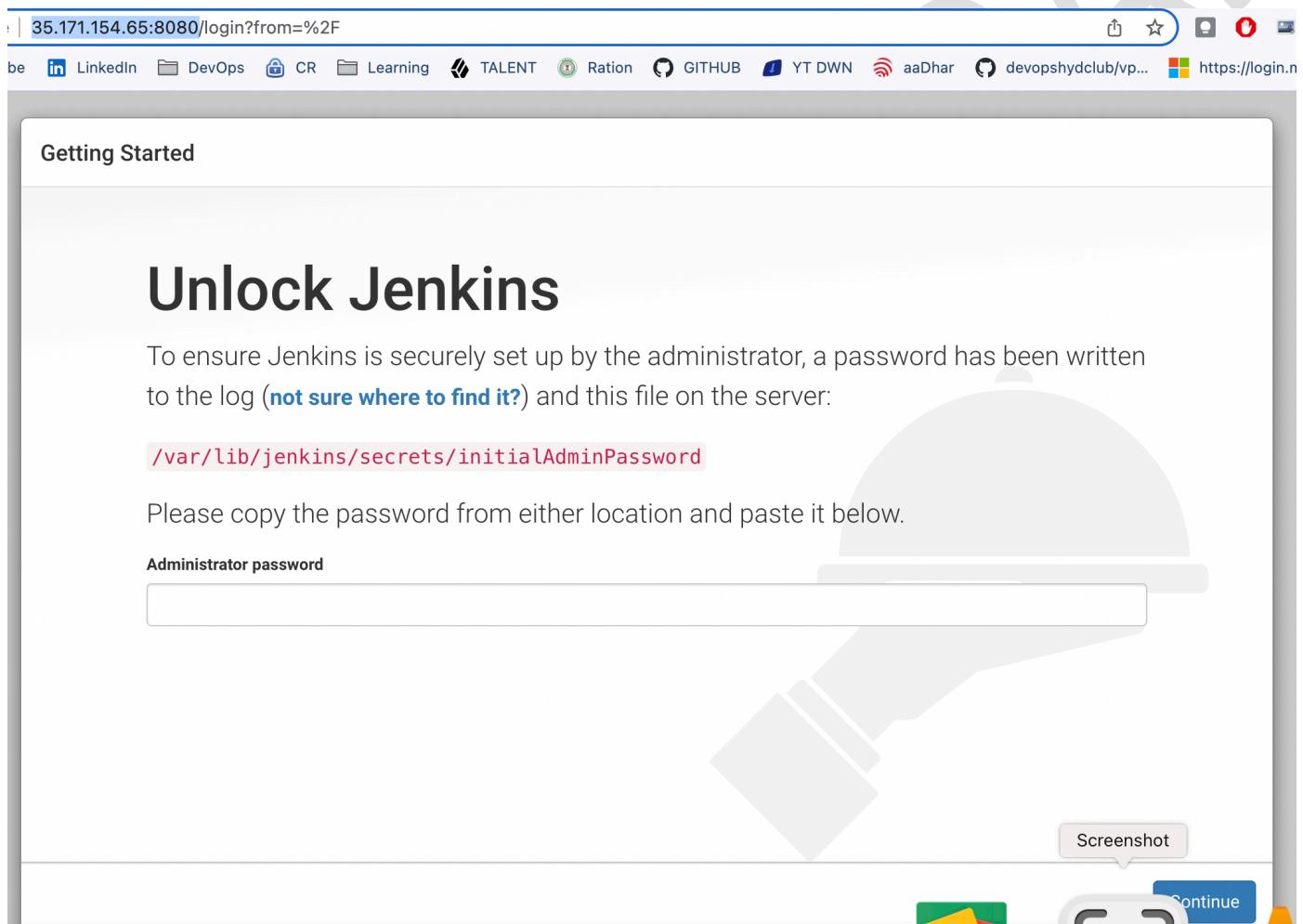
A screenshot of a terminal window titled 'DevOps\_Projects -- zsh -- 80x24'. The command 'ls userdata' is run, showing files: jenkins-setup.sh, nexus-setup.sh, sonar-analysis-properties, and sonar-setup.sh.

```
merajhassan@MERAJs-MacBook-Air DevOps_Projects % ls userdata  
jenkins-setup.sh  
nexus-setup.sh  
sonar-analysis-properties  
sonar-setup.sh
```

## 2. Setup Jenkins, Nexus & SonarQube Server

Instances (1/3) <a href="#">Info</a>							
	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
<input type="checkbox"/>	Nexus-Server	i-01982dc566661f662	<span>Running</span>  	t2.medium	<span>Initializing</span> 	No alarms 	us-east-1d
<input checked="" type="checkbox"/>	Jenkins-Server	i-0dde0e3c038db66f3	<span>Running</span>  	t2.small	<span>2/2 checks passed</span> 	No alarms 	us-east-1d
<input type="checkbox"/>	SonarQube-Se...	i-0f350bfa8ac3a3267	<span>Running</span>  	t2.medium	<span>Initializing</span> 	No alarms 	us-east-1d

Check Jenkins is working - <http://35.171.154.65:8080/>



The screenshot shows the Jenkins 'Unlock Jenkins' setup page. At the top, there's a header with a search bar and various links. Below it, a large heading says 'Getting Started' and 'Unlock Jenkins'. A text block explains that a password has been written to the log and on the server. It provides the path '/var/lib/jenkins/secrets/initialAdminPassword' in red. Below this, there's a text input field labeled 'Administrator password' with a placeholder 'Enter password here'. On the right, there are 'Screenshot' and 'Continue' buttons.

```
$ ssh -i "ci-eereeda-key.pem" ubuntu@ec2-35-171-154-65.compute-1.amazonaws.com
$ sudo -i
# cat /var/lib/jenkins/secrets/initialAdminPassword
```

The screenshot shows the Jenkins dashboard at <http://35.171.154.65:8080>. The left sidebar contains links for 'New Item', 'People', 'Build History', 'Manage Jenkins', 'My Views', and 'New View'. The main content area features a 'Welcome to Jenkins!' heading and a 'Start building your software project' section with a 'Create a job' button.

Check Nexus is working - <http://54.165.249.37:8081/>

The screenshot shows the Sonatype Nexus Repository Manager at <http://54.165.249.37:8081>. The left sidebar has 'Welcome', 'Search', and 'Browse' options. The main content includes a 'Welcome' message, a 'Read More...' button, a 'Get Started' section, a 'Configuration' section, and a 'Repository Formats' section. A 'Sign In' dialog box is overlaid on the page, containing instructions about the admin password location and two input fields for 'Username' and 'Password'.

```
$ ssh -i "ci-eereeda-key.pem" centos@ec2-54-165-249-37.compute-1.amazonaws.com
$ sudo -i
$ cat /opt/nexus/sonatype-work/nexus3/admin.password
```

The screenshot shows the Sonatype Nexus Repository Manager at <http://54.165.249.37:8081>. The left sidebar includes 'Upload' in addition to 'Welcome', 'Search', and 'Browse'. The main content features a warning message about the log4j vulnerability and its mitigation, followed by a 'Enable Capability' button.

## Settings > Repositories > Create Repository

Maven hosted – Upload our artifact

Maven hosted – Maven Build tool will download dependencies from here  
[<https://repo1.maven.org/maven2/>]

Maven group – Group both repository together

The screenshot shows the Artifactory Manager interface with the 'Repositories' screen selected. The top navigation bar includes 'Manager', a cube icon, a gear icon (highlighted in green), a search bar, and user information for 'admin'. Below the header, there's a 'Create repository' button and a 'Filter' dropdown. The main area lists three repositories:

Name ↑	Type	Format	Status	URL	Health check	IQ Policy Vi...
eereeda-central	proxy	maven2	Online - Ready to Connect	<button>copy</button>	Analyze	
eereeda-maven-group	group	maven2	Online	<button>copy</button>		
eereeda-release	hosted	maven2	Online	<button>copy</button>		

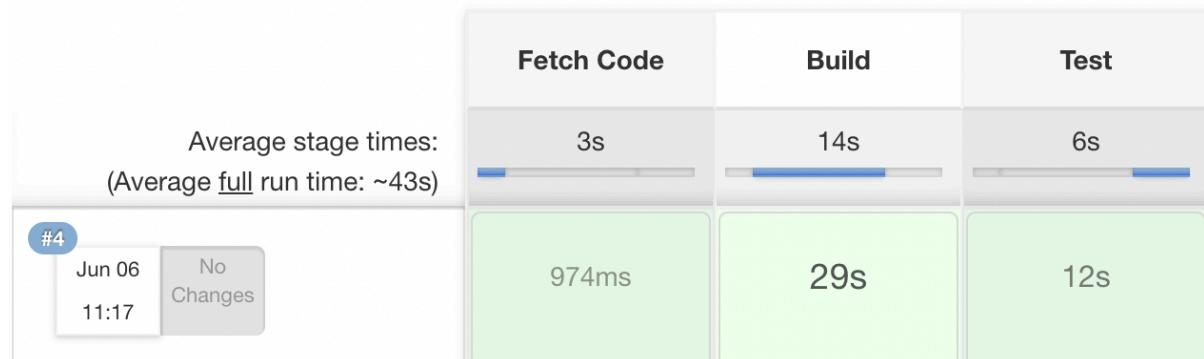
### 3. Pipeline as a Code:

Manage Jenkins > Plugins > Available > Pipeline Utility Steps & Pipeline Maven Integration Plugin > Install without restart

New Item > Pipeline > sample-paas

Sample Pipeline Code:

## Stage View



```
Get Started Jenkinsfile ×  
Jenkinsfile  
1 pipeline {  
2     agent any  
3     stages [  
4         stage('Fetch Code') {  
5             steps {  
6                 git branch: 'paac', url: 'https://github.com/merajafnan/DevOps_Projects.git'  
7             }  
8         }  
9         stage('Build') {  
10            steps {  
11                sh 'mvn install'  
12            }  
13        }  
14         stage('Test') {  
15            steps {  
16                sh 'mvn test'  
17            }  
18        }  
19    }  
20 }
```

## Source Code for Jenkinsfile:

[https://github.com/merajafnan/DevOps\\_Projects/blob/42013bd4427e239b3638f349b06e01caacb1d6fe/Jenkinsfile](https://github.com/merajafnan/DevOps_Projects/blob/42013bd4427e239b3638f349b06e01caacb1d6fe/Jenkinsfile)

## Stage View

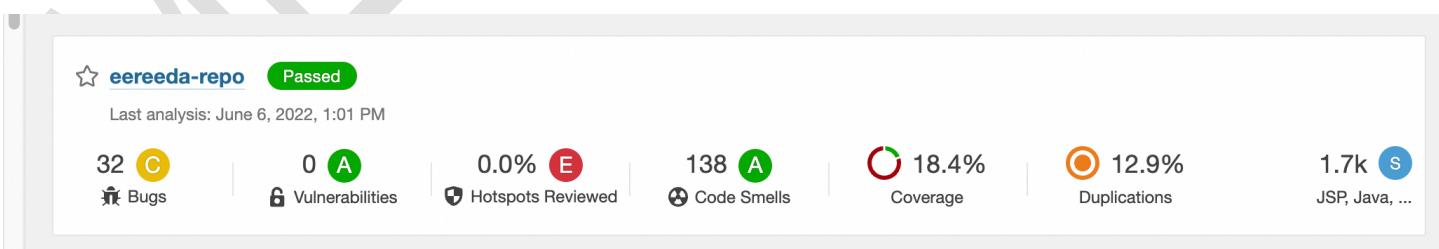
	Fetch Code	Build	Unit Test	Integration Test	Code Analysis with CHECKSTYLE	CODE ANALYSIS with SONARQUBE	Publish to Nexus Repository Manager
Average stage times: (Average full run time: ~1min 11s)							
#5 Jun 06 13:11 No Changes	1s	12s	12s	16s	8s	17s (paused for 2s)	3s
#4 Jun 06 13:00 No Changes	308ms	12s	12s	16s	8s	16s (paused for 2s)	3s
#3 Jun 06 12:48 No Changes	298ms	13s	12s	16s	8s	15s (paused for 1s)	3s
#2 Jun 06 12:45 No Changes	431ms	12s	12s	16s	8s	17s (paused for 9min 59s) aborted	3s
	473ms	12s	12s	16s	8s	16s (paused for 2s)	3s

## Webhook:

The screenshot shows the SonarQube Project Settings page for the 'eereeda-repo' project. At the top, there's a navigation bar with 'Overview', 'Issues', 'Security Hotspots', 'Measures', 'Code', and 'Activity'. On the right, there are buttons for 'Project Settings' (selected), 'Project information', and a 'Create' button. A message at the top right indicates 'Last analysis had 2 warnings' on June 6, 2022, at 1:01 PM. Below the message, the 'Webhooks' section is visible, which explains what webhooks are and how they can be used to notify external services. A table lists the current webhook configuration:

Name	URL	Secret?	Last delivery
eereeda-jenkins-paac	http://172.31.84.74:8080/sonarqube-webhook/	Yes	June 6, 2022, 1:02 PM

## Sonarqube Result:



#### 4. References

The following table summarizes the documents referenced in this plan.

DOCUMENT NAME	INSTRUCTOR	LOCATION
DevOps Beginners to Advanced	Imran Teli	<a href="https://www.udemy.com/course/decodingdevops/learn/lecture/28273912?start=0#overview">https://www.udemy.com/course/decodingdevops/learn/lecture/28273912?start=0#overview</a>