< 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*

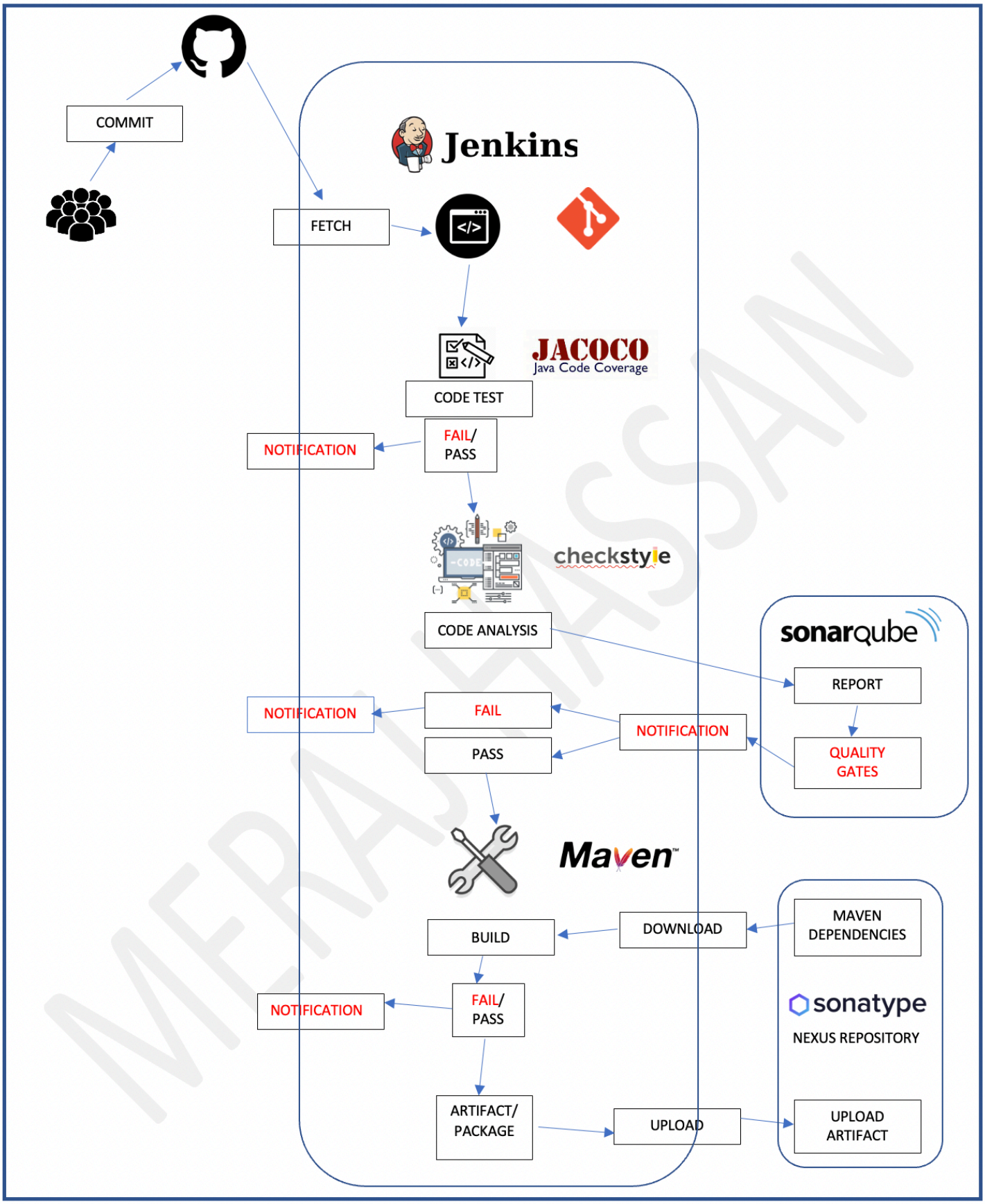
Objective

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

Architecture of AWS Services

- *EC2 Instances*

Architecture Continuous Integration Pipeline



Flow of Execution

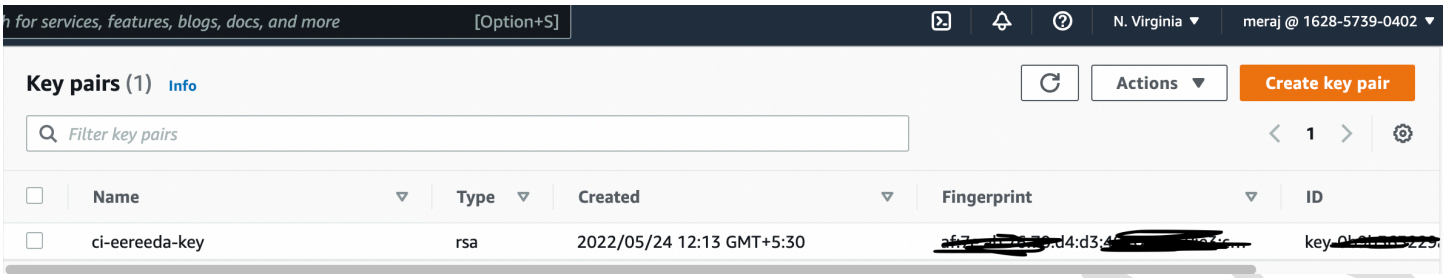
- *Login to AWS Account*
- *Create Login Key*
- *Create Security Group*
    - *Jenkins*
    - *Nexus*
    - *Sonar*
- *Create EC2 instance with user data*
    - *Jenkins*
    - *Sonarqube*
    - *Nexus*
- *Jenkins post installation*
- *Nexus repository setup*
    - *3 Repos*
- *Sonarqube Post Installation*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

Prerequisite

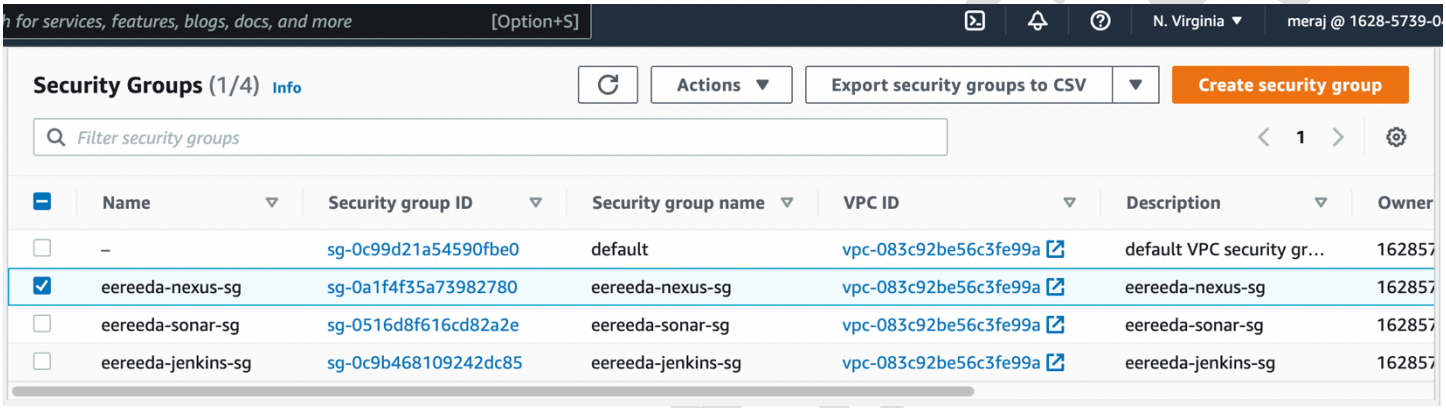
1. *AWS Account*
2. *GITHUB Account*
3. *Intellij*

1. Security Group & Keypairs

a. KeyPairs



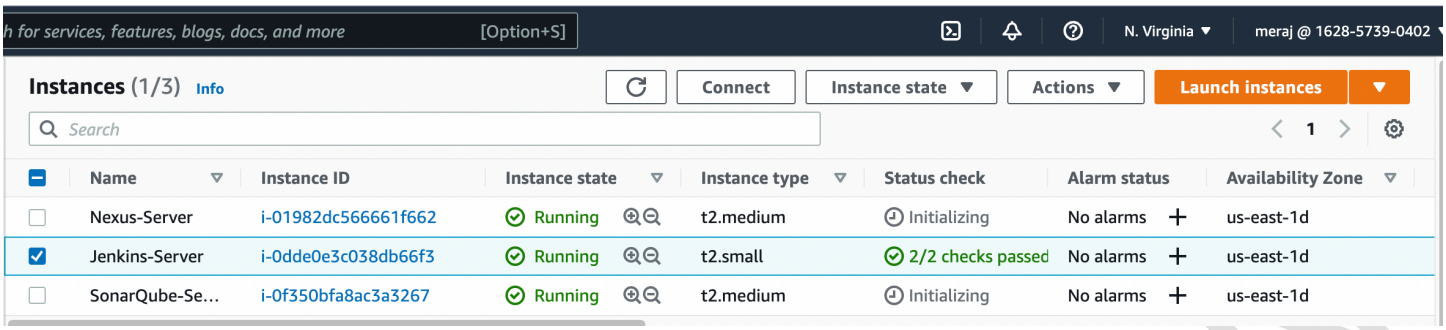
b. Security Group



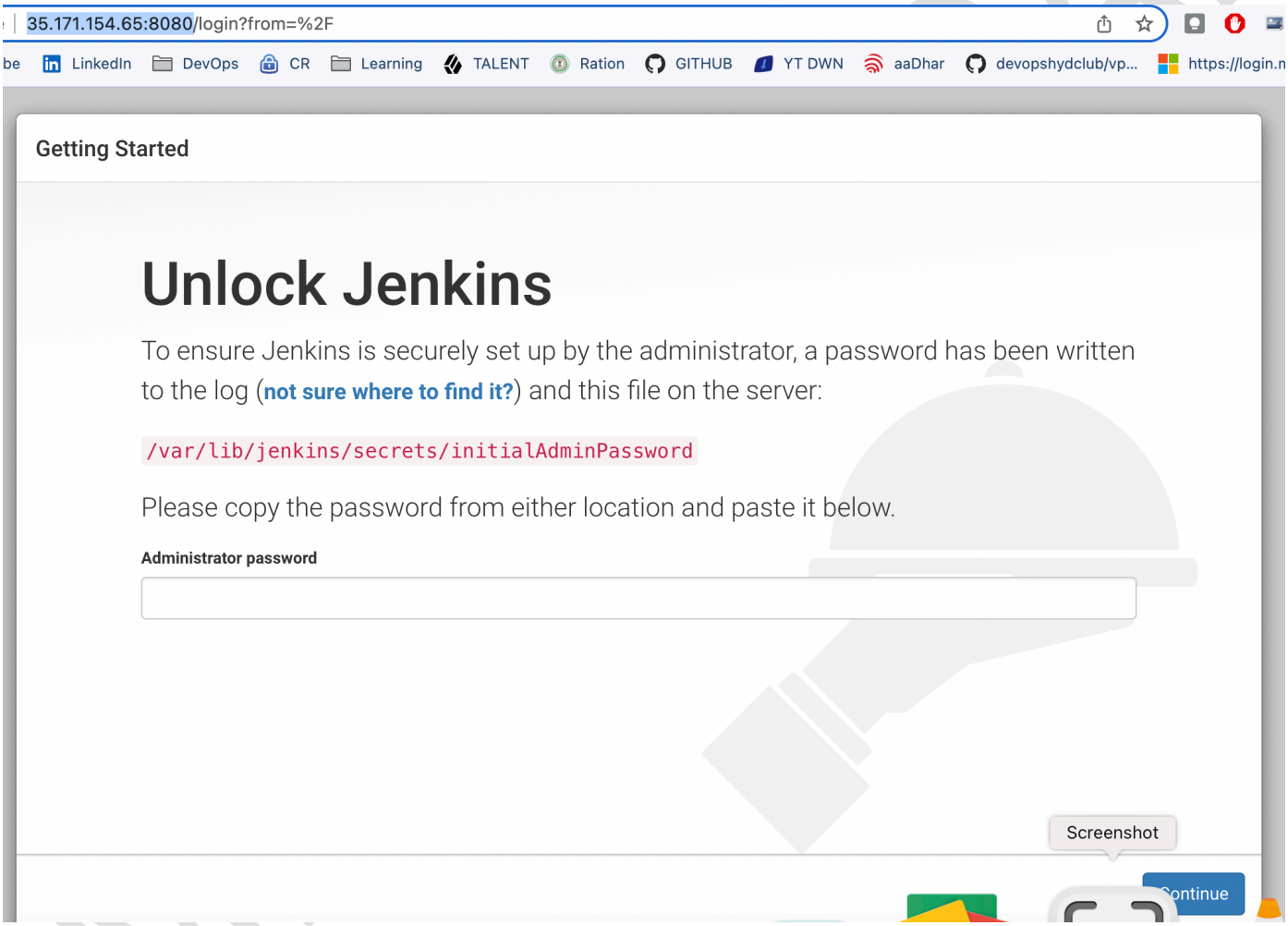
User Data Script

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

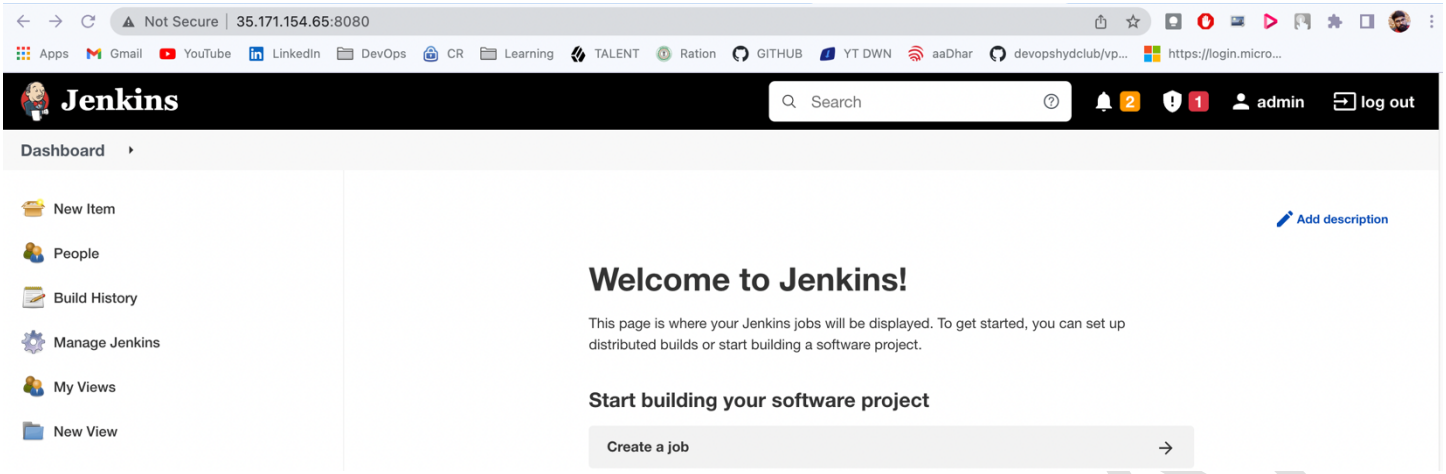
2. Setup Jenkins, Nexus & SonarQube Server



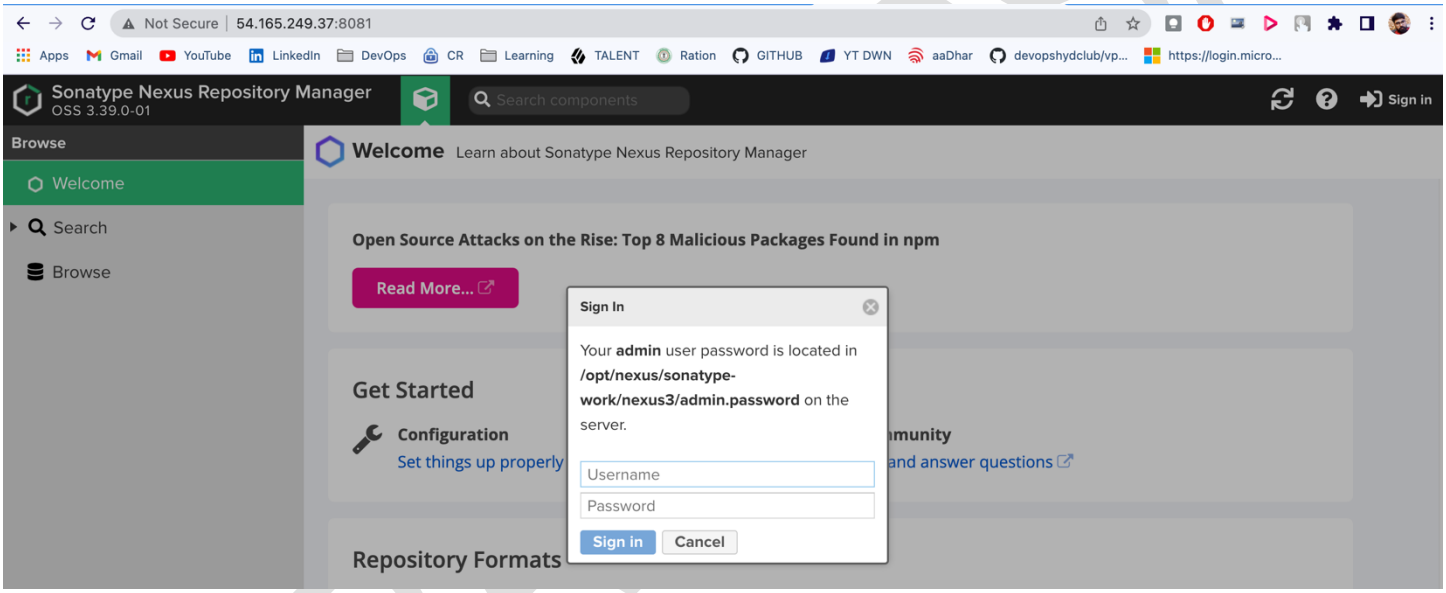
Check Jenkins is working - http://35.171.154.65:8080/



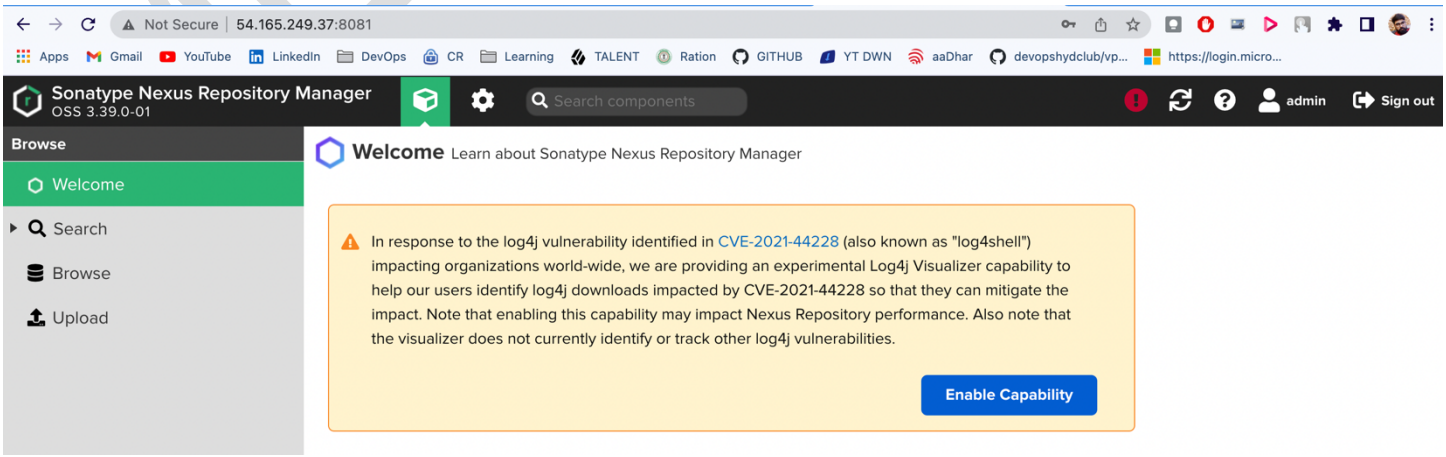
$ ssh -i "ci-eereeda-key.pem" ubuntu@ec2-35-171-154-65.compute-1.amazonaws.com
$ sudo -i
# cat /var/lib/jenkins/secrets/initialAdminPassword



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

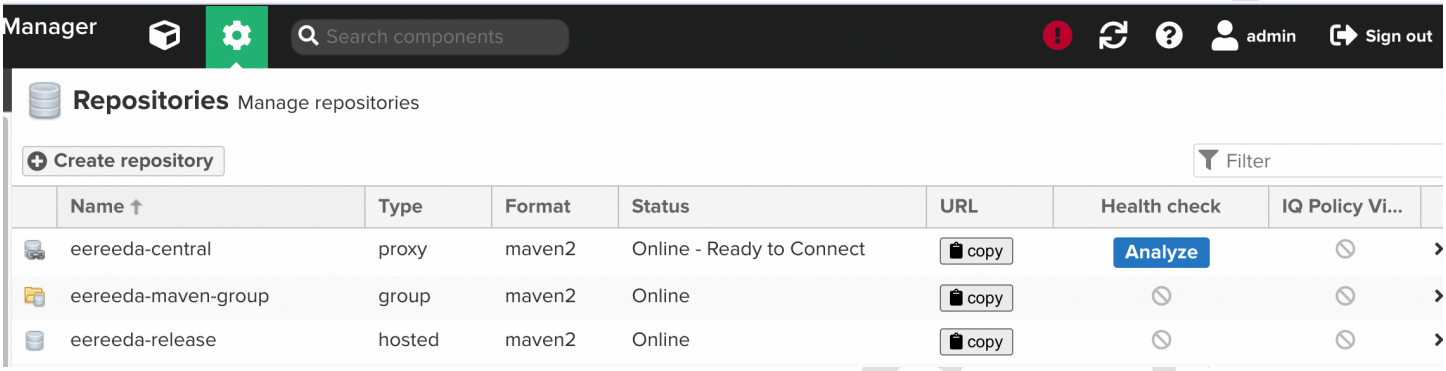


$ ssh -i "ci-eereeda-key.pem" centos@ec2-54-165-249-37.compute-1.amazonaws.com
$ sudo -i
$ cat /opt/nexus/sonatype-work/nexus3/admin.password



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

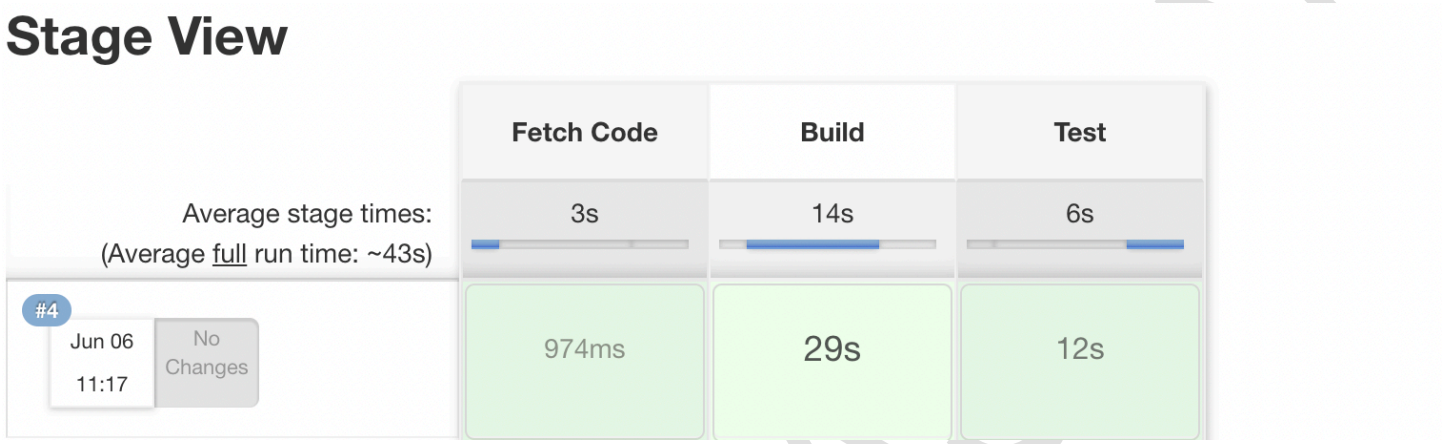


3. **Pipeline as a Code:**

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

New Item > Pipeline > sample-paas

Sample Pipeline Code:



Get Started | Jenkinsfile

Jenkinsfile

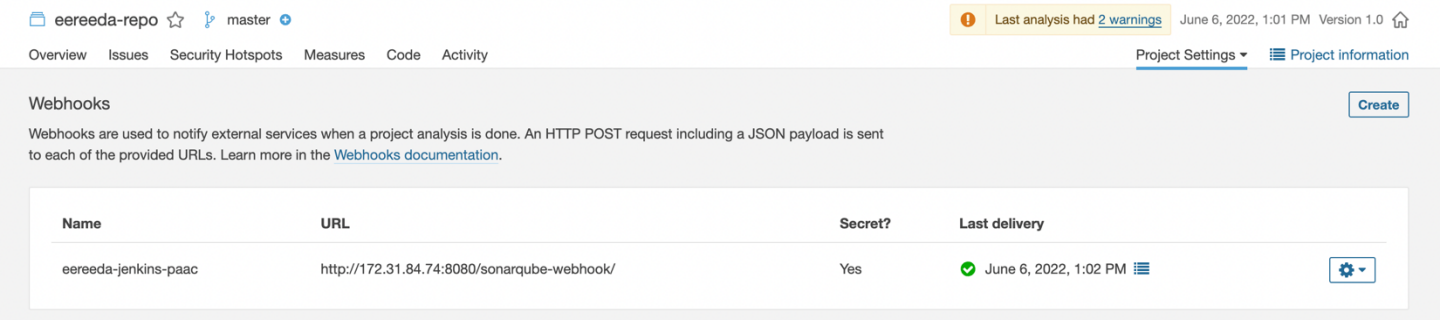
```
pipeline {
    agent any
    stages {
        stage('Fetch Code') {
            steps {
                git branch: 'paac', url: 'https://github.com/merajafnan/DevOps_Projects.git'
            }
        }
        stage('Build') {
            steps {
                sh 'mvn install'
            }
        }
        stage('Test') {
            steps {
                sh 'mvn test'
            }
        }
    }
}
```

Source Code for Jenkinsfile:
https://github.com/merajafnan/DevOps_Projects/blob/42013bd4427e239b3638f349b06e01caacb1d6fe/Jenkinsfile

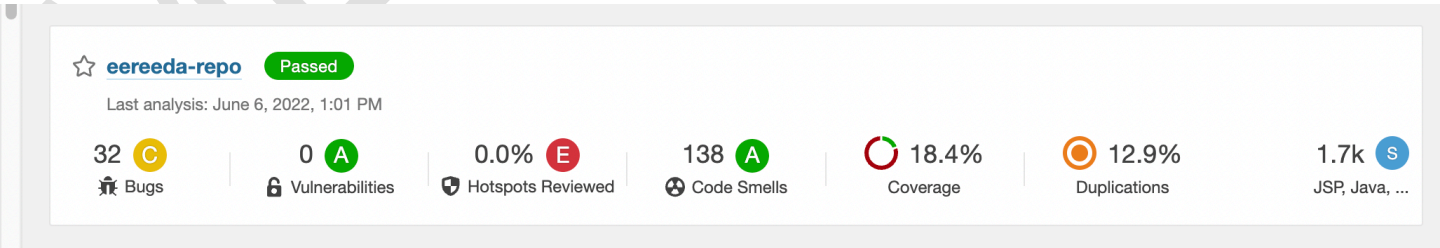
Stage View

	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)	1s	12s	12s	16s	8s	17s	3s
#5 Jun 06 13:11 No Changes	308ms	12s	12s	16s	8s	16s (paused for 2s)	3s
#4 Jun 06 13:00 No Changes	298ms	13s	12s	16s	8s	15s (paused for 1s)	
#3 Jun 06 12:48 No Changes	431ms	12s	12s	16s	8s	17s (paused for 9min 59s) aborted	
#2 Jun 06 12:45 No Changes	473ms	12s	12s	16s	8s	16s (paused for 2s)	

Webhook:



Sonarqube Result:



4. References

The following table summarizes the documents referenced in this plan.

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