SonarQube Code Analysis Project:

SonarQube Code Analysis Project:

-Building a SonarQube code Analysis project for a client.

What is SonarQube?

-SonarQube is an automatic code review tool to detect bugs, vulnerabilities, and code smells. SonarQube can be integrated with existing workflow to enable continuous code inspection across the project branches and pull requests. More importantly, SonarQube empowers all developers to write cleaner and safer code across the board.

# Problem Statement:

-A manager at an IT solution development company wants to add one more extra person to the team to help with setting up a code review process with SonarQube for their project.

# IT solution development company's project overview:

-The company is creating a geolocation application for medical doctors and patients. When doctors log in, they would see patients' information and when patients log in, they would see their records and appointments. The team already created a project repository on GitHub to manage the code. The project is written in Java language and they use maven as a build tool to compile the source code.

# Solution Statement:

# My role for this project:

-I am going to be the DevOps Engineer in this team because I am going to set up the SonarQube analysis process for the project and help them to produce good quality code.

-I am going to install and configure a Jenkins server for the project and integrate Maven in Jenkins as the build tool. Then, create a maven project in

Jenkins and link it to the GitHub project repository. Before, integrating SonarCloud with Jenkins to perform the code analysis, I need to create a SonarCloud account for the project which would send analysis results to the developers, so they can improve the quality of their code.

# Technology tools:

-Linux Centos server: the OS environment that I am going to run the Jenkins server on it.

-Jenkins: To build the project

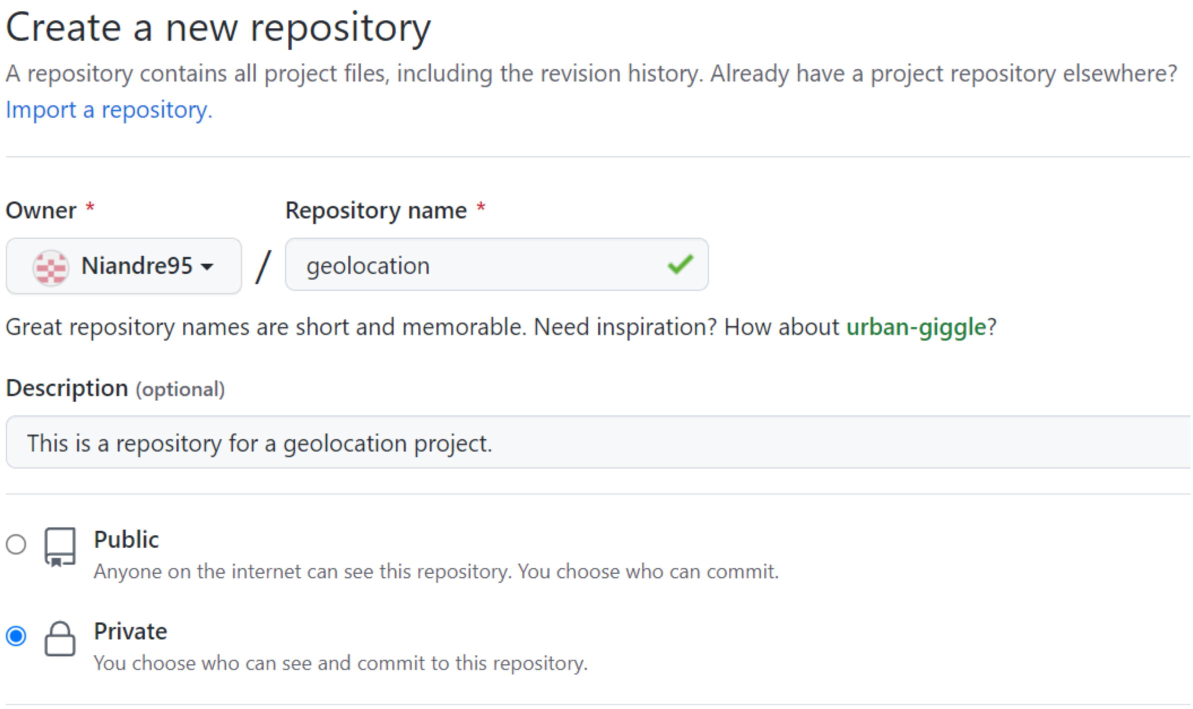
-Git: To connect and install the necessary package on the Jenkins server to build the project.

-GitHub: To host the project repository

-Maven: To build the Java source code

-SonarQube: To analyze the source code

# Creating a GitHub repository to store the project source code



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Owner * Niandre95 / Repository name * geolocation ✓

Great repository names are short and memorable. Need inspiration? How about urban-giggle?

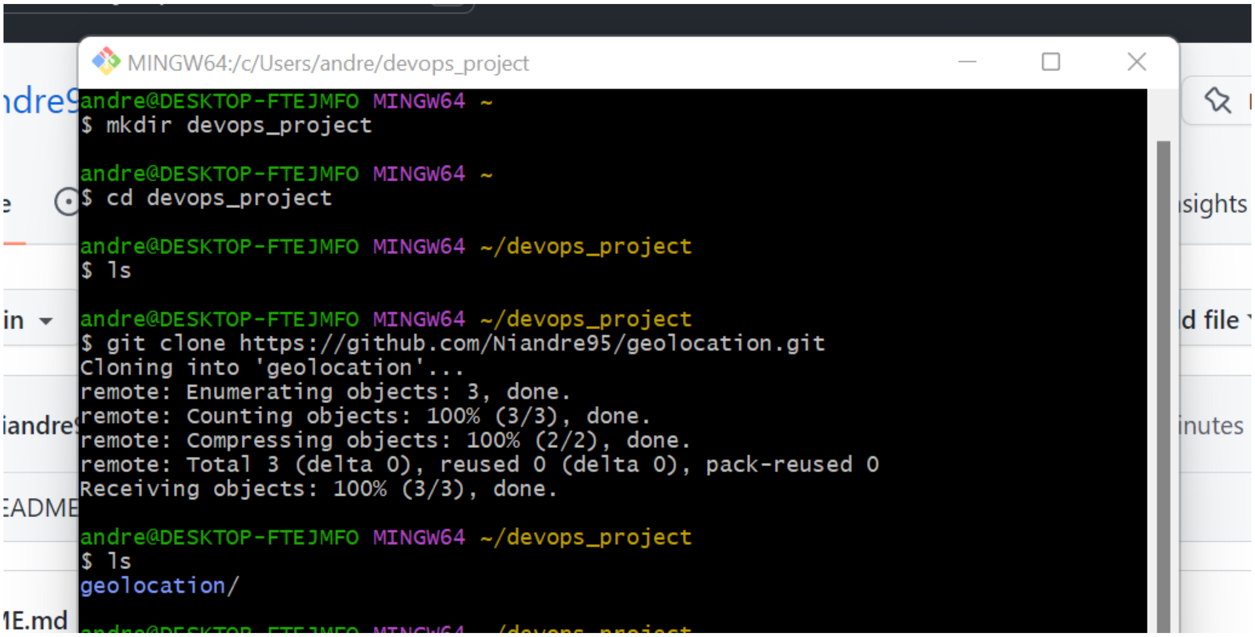
Description (optional)

This is a repository for a geolocation project.

○ **Public**
Anyone on the internet can see this repository. You choose who can commit.

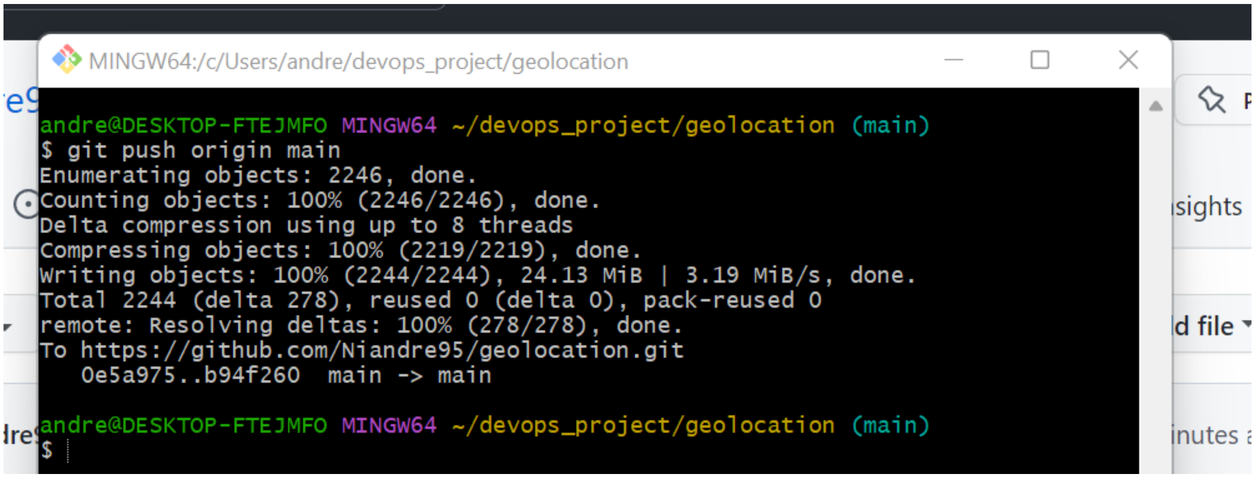
● **Private**
You choose who can see and commit to this repository.

# Clone the GitHub repository to my local repository



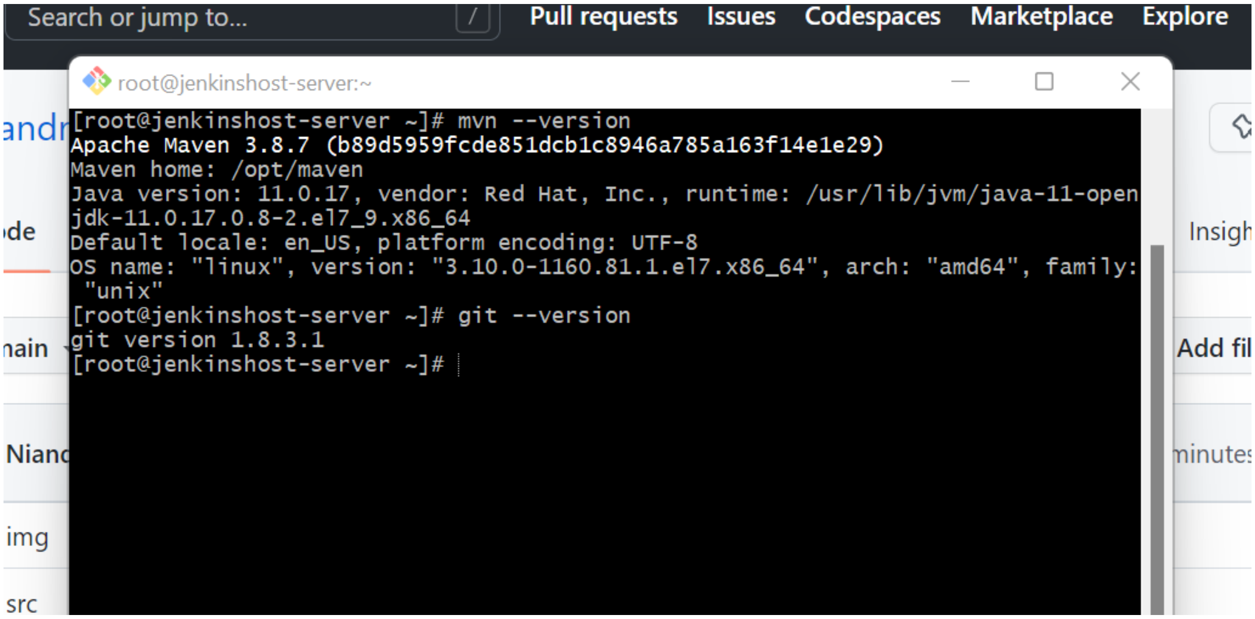
```
andre@DESKTOP-FTEJMFO MINGW64 ~
$ mkdir devops_project

andre@DESKTOP-FTEJMFO MINGW64 ~
$ cd devops_project

andre@DESKTOP-FTEJMFO MINGW64 ~/devops_project
$ ls

andre@DESKTOP-FTEJMFO MINGW64 ~/devops_project
$ git clone https://github.com/Niandre95/geolocation.git
Cloning into 'geolocation'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.

andre@DESKTOP-FTEJMFO MINGW64 ~/devops_project
$ ls
geolocation/

andre@DESKTOP-FTEJMFO MINGW64 ~/devops_project
```

# Pushing the code to my GitHub repository



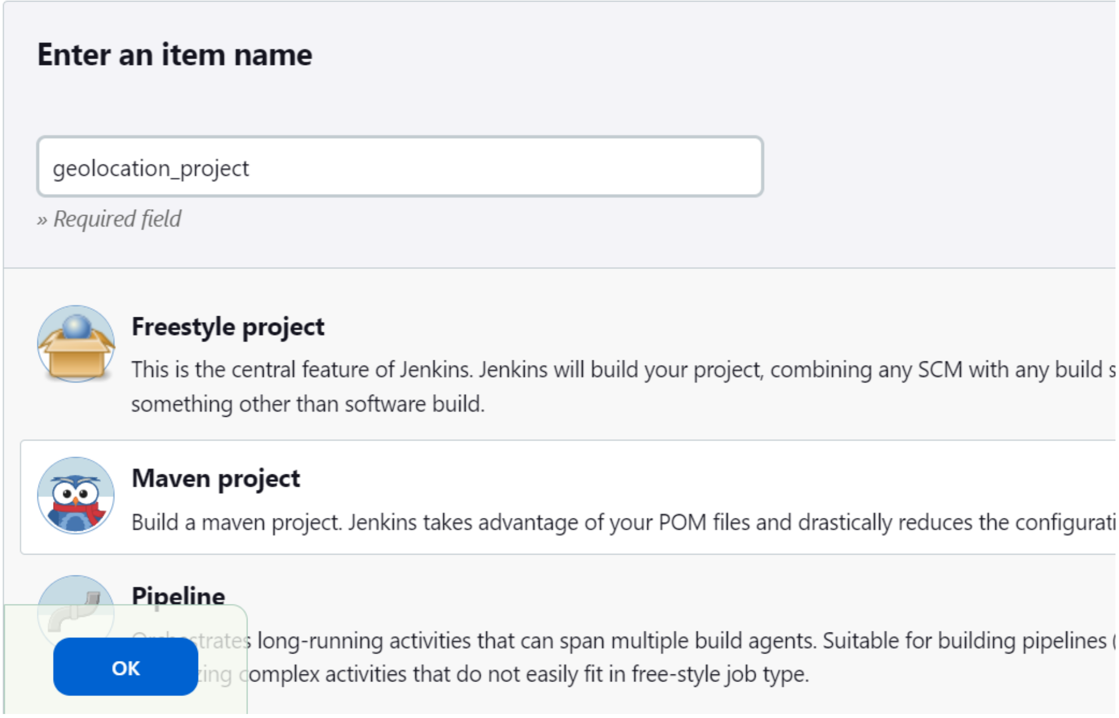
```
andre@DESKTOP-FTEJMFO MINGW64 ~/devops_project/geolocation (main)
$ git push origin main
Enumerating objects: 2246, done.
Counting objects: 100% (2246/2246), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2219/2219), done.
Writing objects: 100% (2244/2244), 24.13 MiB | 3.19 MiB/s, done.
Total 2244 (delta 278), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (278/278), done.
To https://github.com/Niandre95/geolocation.git
   0e5a975..b94f260  main -> main

andre@DESKTOP-FTEJMFO MINGW64 ~/devops_project/geolocation (main)
$
```

-Already configure my Maven and Git on the Jenkins server



```
[root@jenkinshost-server ~]# mvn --version
Apache Maven 3.8.7 (b89d5959fcde851dcb1c8946a785a163f14e1e29)
Maven home: /opt/maven
Java version: 11.0.17, vendor: Red Hat, Inc., runtime: /usr/lib/jvm/java-11-open
jdk-11.0.17.0.8-2.el7_9.x86_64
Default locale: en_US, platform encoding: UTF-8
OS name: "linux", version: "3.10.0-1160.81.1.el7.x86_64", arch: "amd64", family:
 "unix"
[root@jenkinshost-server ~]# git --version
git version 1.8.3.1
[root@jenkinshost-server ~]#
```

# Creating the maven project in Jenkins



Enter an item name

geolocation_project

» Required field

Freestyle project

This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build s
something other than software build.

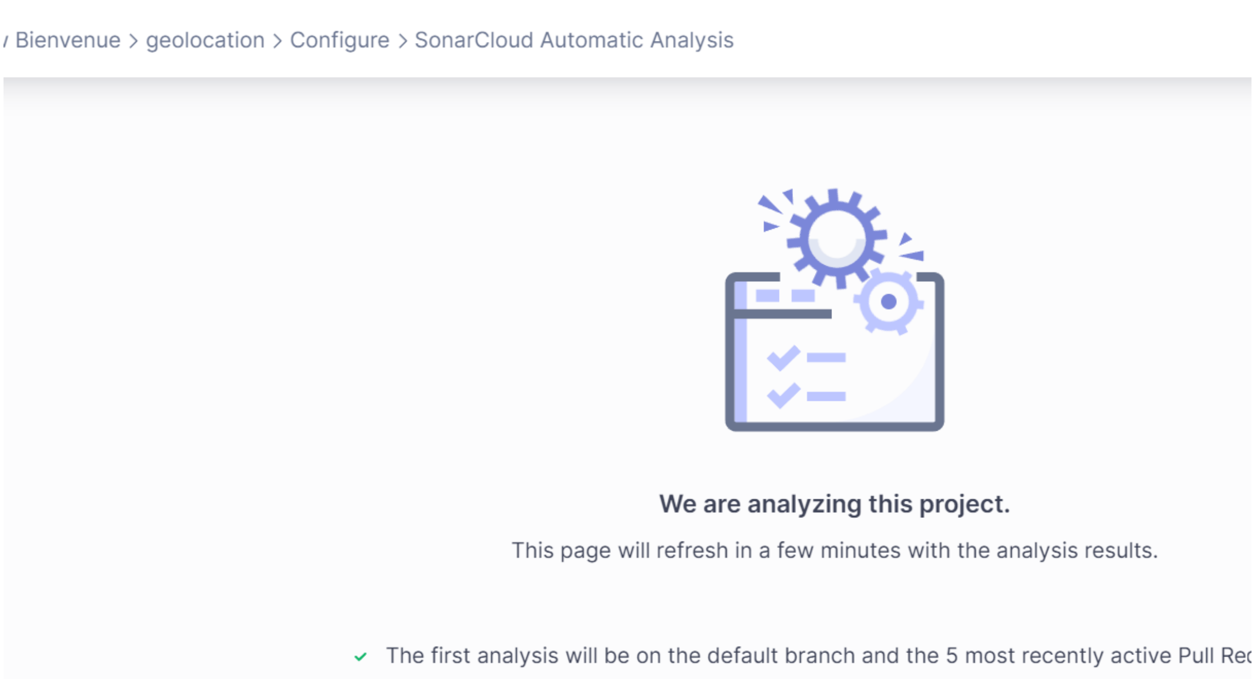
Maven project

Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configurati

Pipeline

Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (
ing complex activities that do not easily fit in free-style job type.

-Doing manual analysis for the project in SonarCloud. Next, I am going to integrate SonarCloud with Jenkins to automate the analysis process.



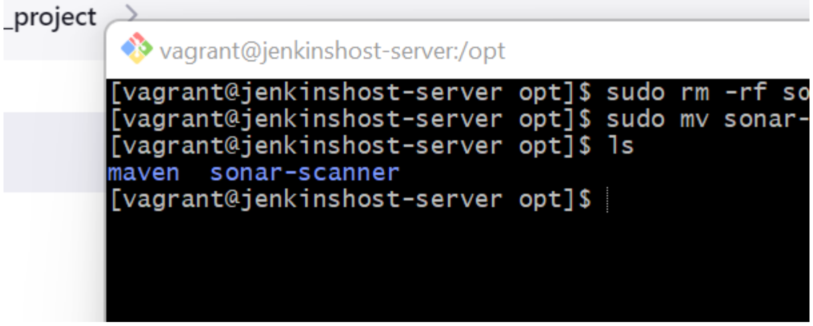
/ Bienvenue > geolocation > Configure > SonarCloud Automatic Analysis

We are analyzing this project.

This page will refresh in a few minutes with the analysis results.

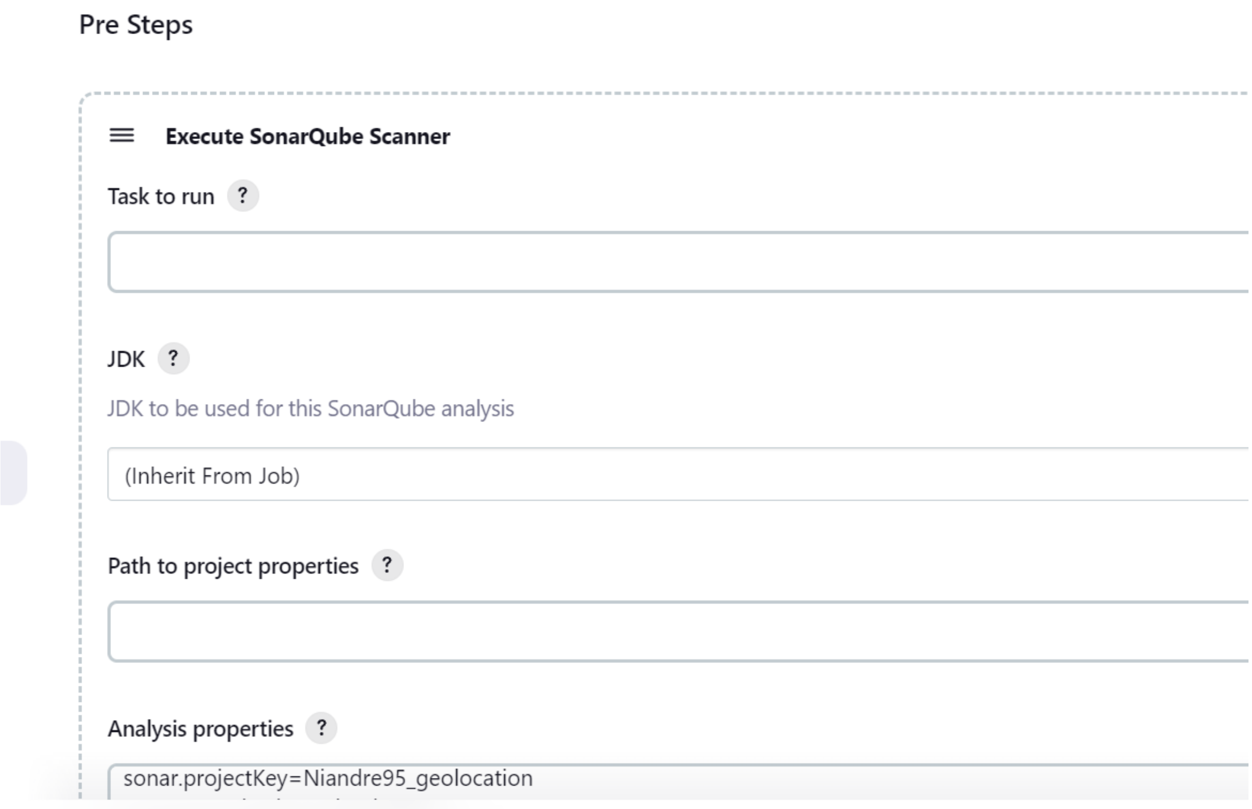
✓ The first analysis will be on the default branch and the 5 most recently active Pull Rec

# Configuring Sonar Scanner on the Jenkins server



```
[vagrant@jenkinshost-server opt]$ sudo rm -rf so
[vagrant@jenkinshost-server opt]$ sudo mv sonar-
[vagrant@jenkinshost-server opt]$ ls
maven  sonar-scanner
[vagrant@jenkinshost-server opt]$
```

-Adding a pre-build step in the project with SonarQube


Pre Steps

Execute SonarQube Scanner

Task to run ?

JDK ?

JDK to be used for this SonarQube analysis

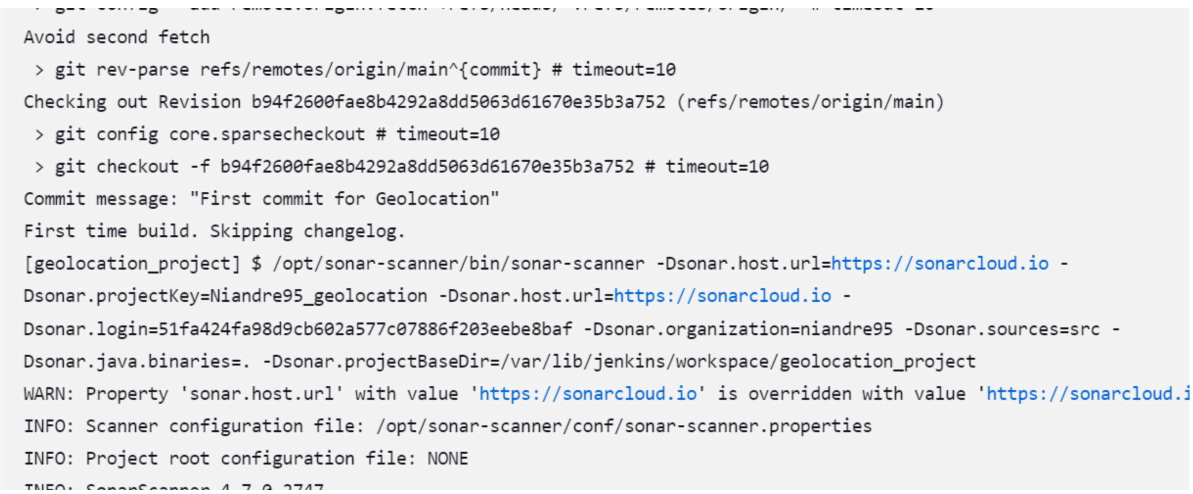
(Inherit From Job)

Path to project properties ?

Analysis properties ?

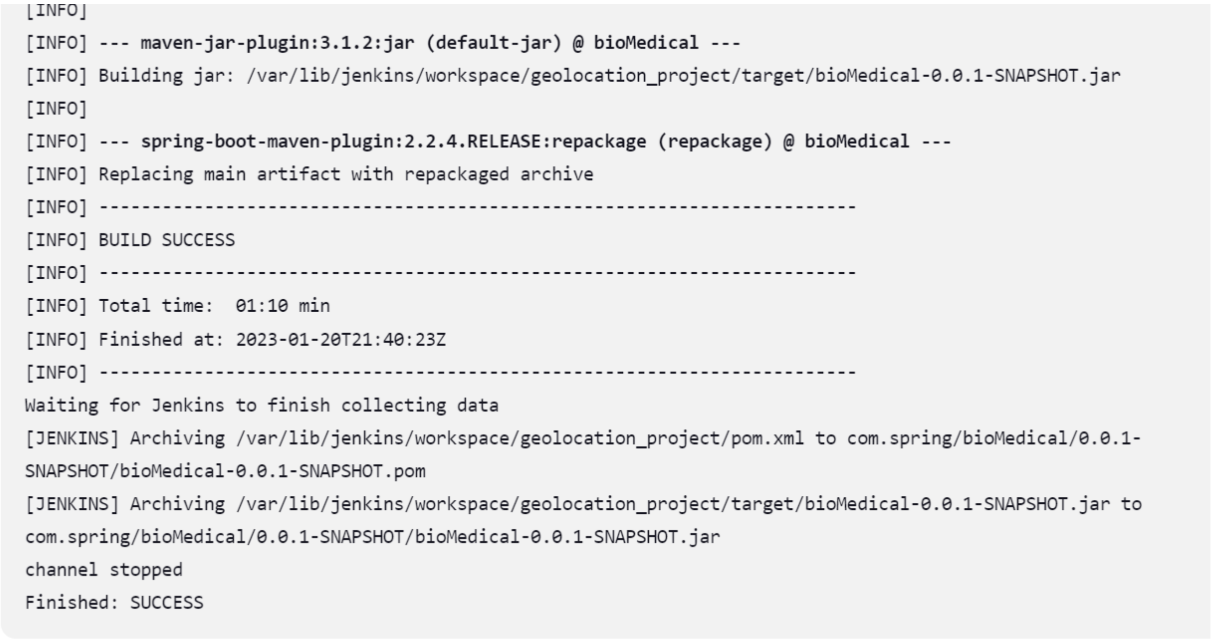
sonar.projectKey=Niandre95_geolocation

# Building the job with Jenkins. Displaying the building output screen.



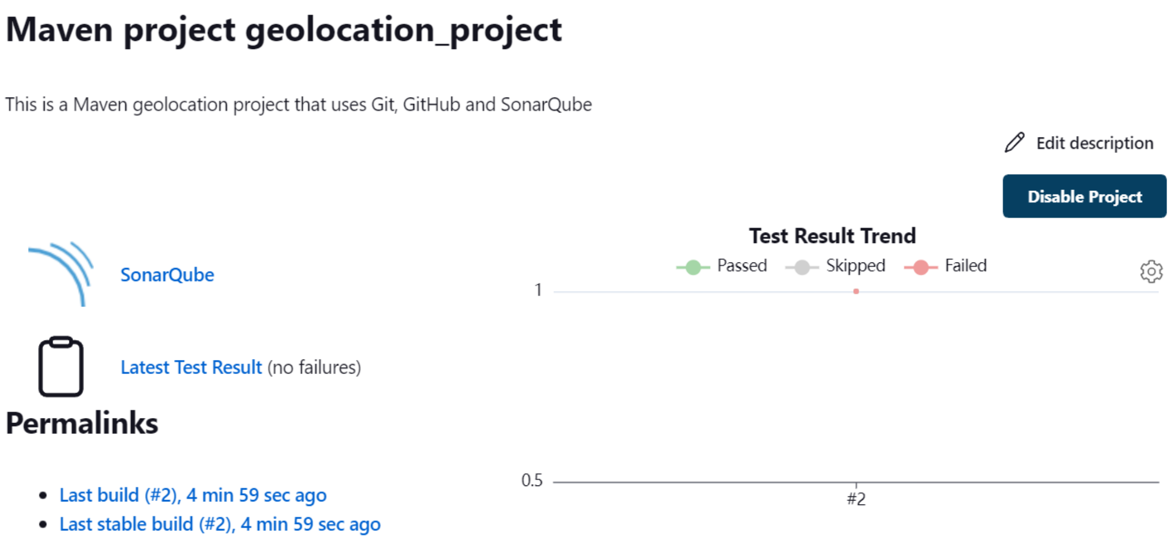
```
 > git config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* # timeout=10
Avoid second fetch
 > git rev-parse refs/remotes/origin/main^{commit} # timeout=10
Checking out Revision b94f2600fae8b4292a8dd5063d61670e35b3a752 (refs/remotes/origin/main)
 > git config core.sparsecheckout # timeout=10
 > git checkout -f b94f2600fae8b4292a8dd5063d61670e35b3a752 # timeout=10
Commit message: "First commit for Geolocation"
First time build. Skipping changelog.
[geolocation_project] $ /opt/sonar-scanner/bin/sonar-scanner -Dsonar.host.url=https://sonarcloud.io -
Dsonar.projectKey=Niandre95_geolocation -Dsonar.host.url=https://sonarcloud.io -
Dsonar.login=51fa424fa98d9cb602a577c07886f203eebe8baf -Dsonar.organization=niandre95 -Dsonar.sources=src -
Dsonar.java.binaries=. -Dsonar.projectBaseDir=/var/lib/jenkins/workspace/geolocation_project
WARN: Property 'sonar.host.url' with value 'https://sonarcloud.io' is overridden with value 'https://sonarcloud.i
INFO: Scanner configuration file: /opt/sonar-scanner/conf/sonar-scanner.properties
INFO: Project root configuration file: NONE
INFO: SonarScanner 4.7.0.2747
```

# The Jenkins Geolocation job was built successfully



```
[INFO]
[INFO] --- maven-jar-plugin:3.1.2:jar (default-jar) @ bioMedical ---
[INFO] Building jar: /var/lib/jenkins/workspace/geolocation_project/target/bioMedical-0.0.1-SNAPSHOT.jar
[INFO]
[INFO] --- spring-boot-maven-plugin:2.2.4.RELEASE:repackage (repackage) @ bioMedical ---
[INFO] Replacing main artifact with repackaged archive
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  01:10 min
[INFO] Finished at: 2023-01-20T21:40:23Z
[INFO] ------------------------------------------------------------------------
Waiting for Jenkins to finish collecting data
[JENKINS] Archiving /var/lib/jenkins/workspace/geolocation_project/pom.xml to com.spring/bioMedical/0.0.1-
SNAPSHOT/bioMedical-0.0.1-SNAPSHOT.pom
[JENKINS] Archiving /var/lib/jenkins/workspace/geolocation_project/target/bioMedical-0.0.1-SNAPSHOT.jar to
com.spring/bioMedical/0.0.1-SNAPSHOT/bioMedical-0.0.1-SNAPSHOT.jar
channel stopped
Finished: SUCCESS
```

-Build results in the Jenkins console



Maven project geolocation_project

This is a Maven geolocation project that uses Git, GitHub and SonarQube

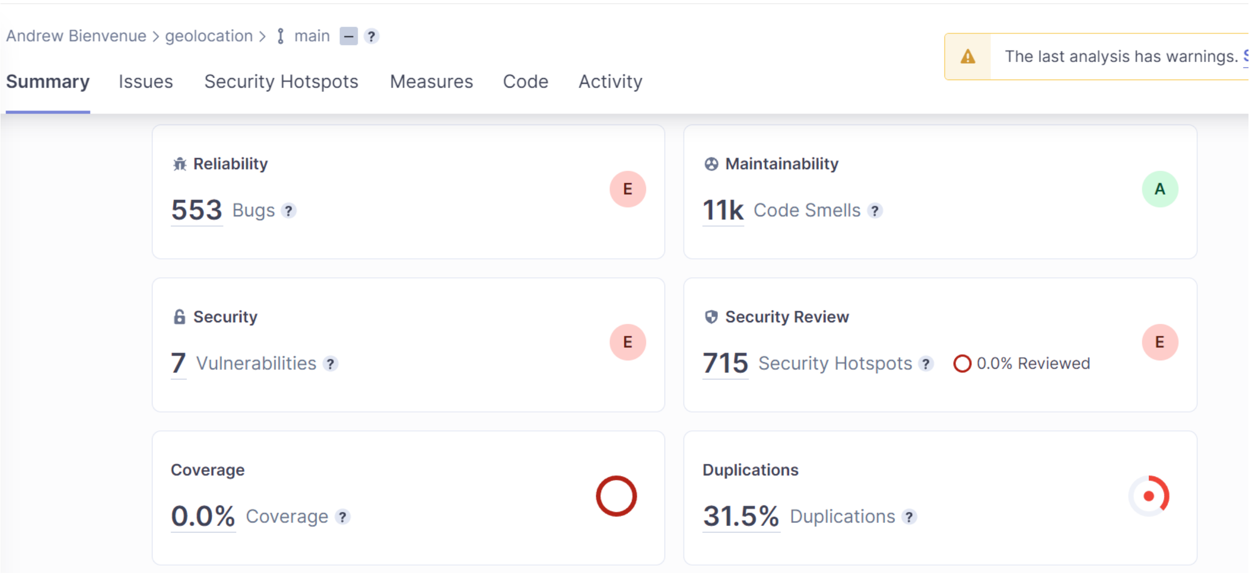
SonarQube

Latest Test Result (no failures)

Permalinks

- Last build (#2), 4 min 59 sec ago
- Last stable build (#2), 4 min 59 sec ago

-SonarCloud Analysis Results



The last analysis has warnings. S

Reliability 553 Bugs ? E

Maintainability 11k Code Smells ? A

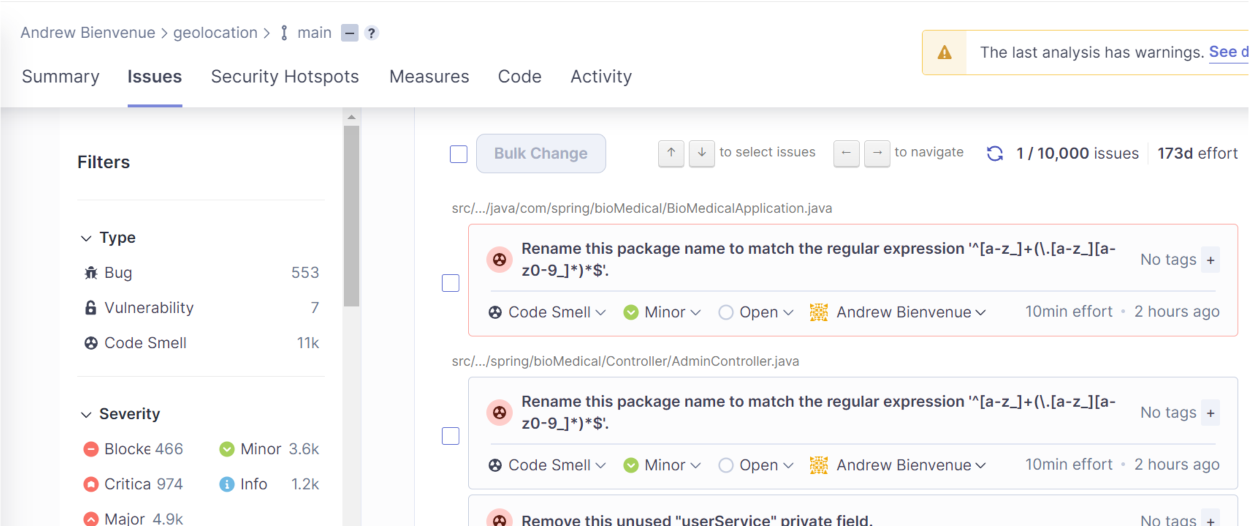
Security 7 Vulnerabilities ? E

Security Review 715 Security Hotspots ? ○ 0.0% Reviewed E

Coverage 0.0% Coverage ?

Duplications 31.5% Duplications ?

-More Output



Andrew Bienvenue > geolocation > main

The last analysis has warnings. See d

↑ ↓ to select issues ← → to navigate 1 / 10,000 issues 173d effort

Type

Bug	553
Vulnerability	7
Code Smell	11k

Severity

Blocke	466	Minor	3.6k
Critica	974	Info	1.2k
Major	4.9k		

src/.../java/com/spring/bioMedical/BioMedicalApplication.java

Rename this package name to match the regular expression '^[a-z_]+(\.[a-z_][a-z0-9_]*)*$'. No tags +

Code Smell Minor Open Andrew Bienvenue 10min effort • 2 hours ago

src/.../spring/bioMedical/Controller/AdminController.java

Rename this package name to match the regular expression '^[a-z_]+(\.[a-z_][a-z0-9_]*)*$'. No tags +

Code Smell Minor Open Andrew Bienvenue 10min effort • 2 hours ago

Remove this unused "userService" private field. No tags +

-Now, the project is integrated with SonarQube to support the code analysis of the project.