

SonarQube Rapport

12/7/2024

—

Management de qualité

—

Dr. Driss Essabbar

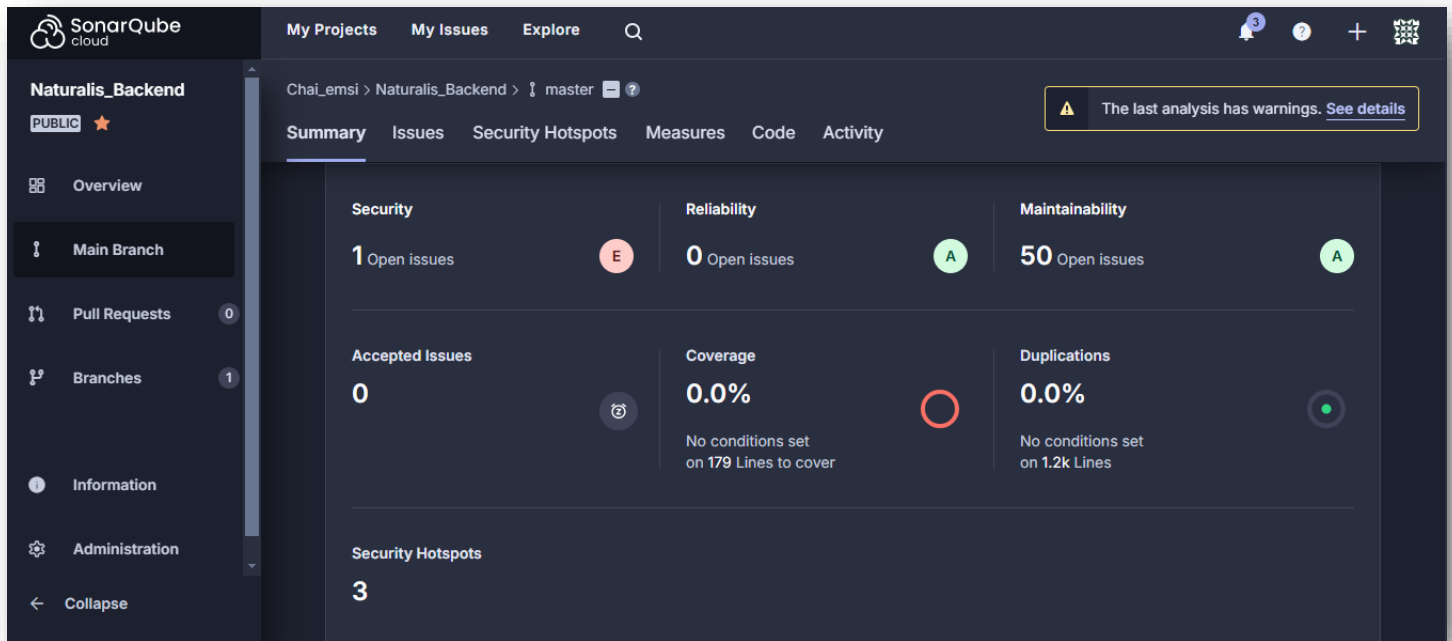
Les métriques initiales:

The screenshot shows the SonarQube cloud interface. On the left, there are filters for Quality Gate (Passed: 1, Failed: 1) and Reliability (A: 3, B: 0, C: 0). The main area displays a search bar, a 'Perspective' dropdown set to 'Overall Status', and a 'Sort by' dropdown set to 'Name'. Below this, a project card for 'Chai_emsi / Naturalis_Backend' is shown with a 'NEW' badge and 'PUBLIC' status. The card indicates 'Last analysis: 12/9/2024, 3:25 PM • 1k Lines of Code • Java, XML, ...'. The metrics are: Security (E 1), Reliability (A 0), Maintainability (A 50), Hotspots Reviewed (E 0.0%), Coverage (0.0%), and Duplications (0.0%). A 'Not computed' button is visible in the top right of the card.

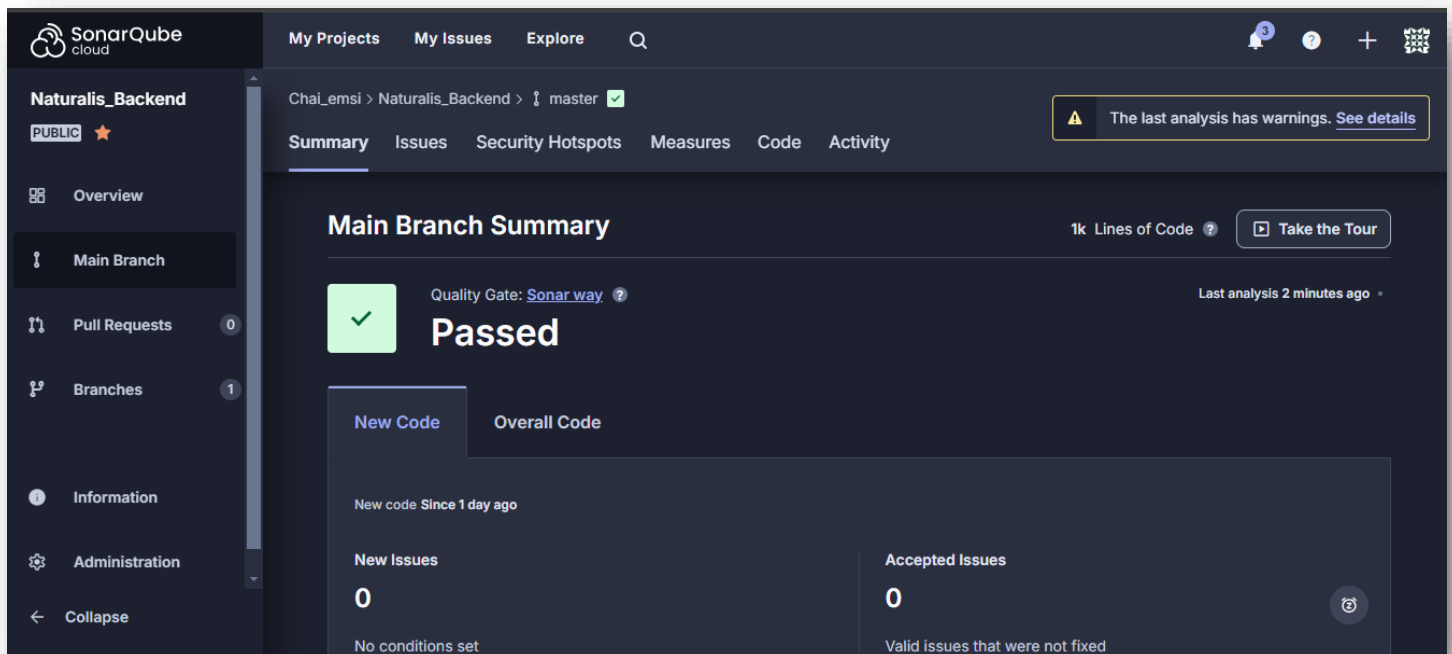
Metric	Value
Security	E 1
Reliability	A 0
Maintainability	A 50
Hotspots Reviewed	E 0.0%
Coverage	0.0%
Duplications	0.0%

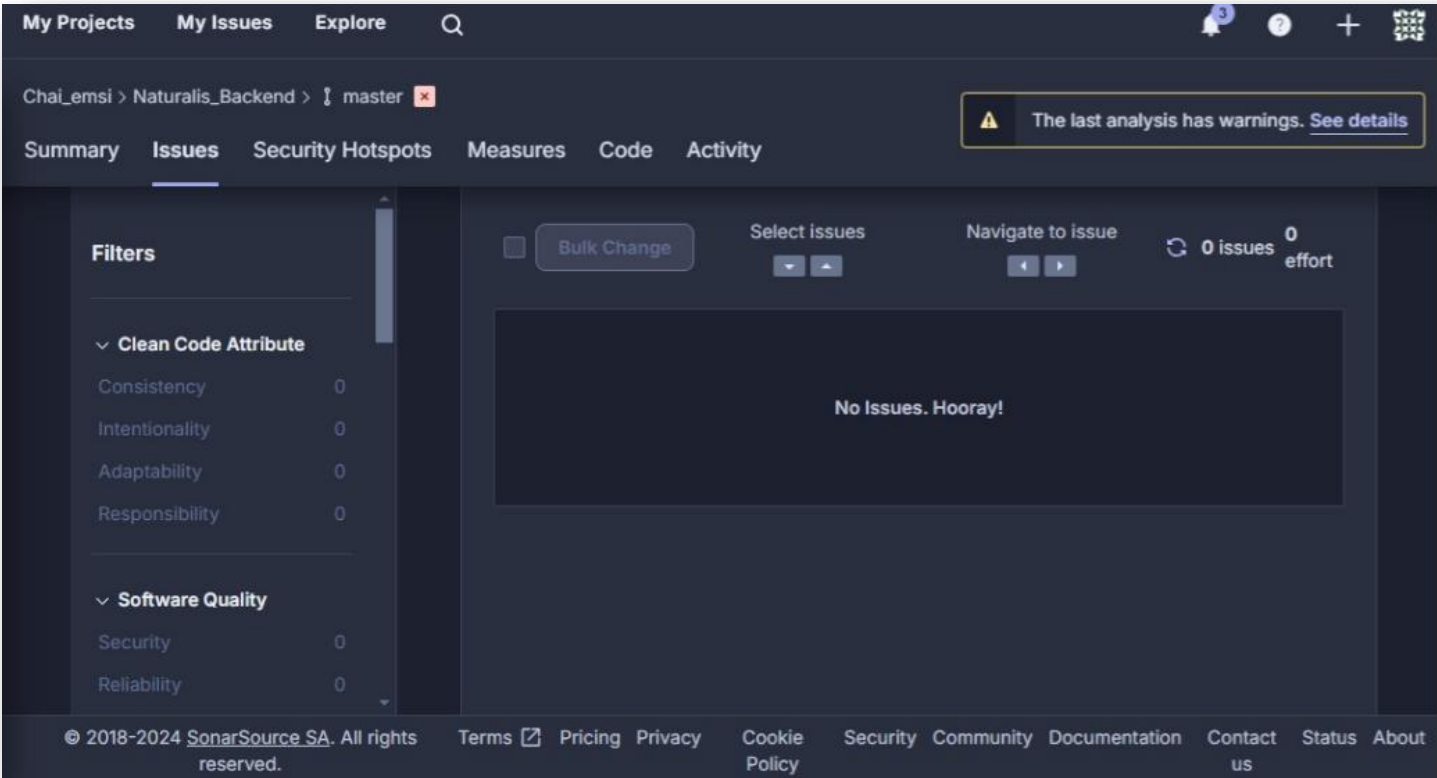
The screenshot shows the 'Latest Activity' section. It displays a 'FIRST ANALYSIS' event for 'Main Branch' on 'December 9 at 3:25 PM'. The event details include '51 Issues', '0.0% Coverage', '0.0% Duplications', and '1k Lines of Code'. A 'Not computed' button is visible in the top right of the activity card.

Event	Branch	Time	Issues	Coverage	Duplications	Lines of Code
FIRST ANALYSIS	Main Branch	December 9 at 3:25 PM	51	0.0%	0.0%	1k



Les métriques finales:







Summary Issues Security Hotspots Measures Code Activity

Naturalis_Backend View as Tree 3 files

Security Review Rating on New Code A New code: since previous version

idea.	A
src	A
pom.xml	A

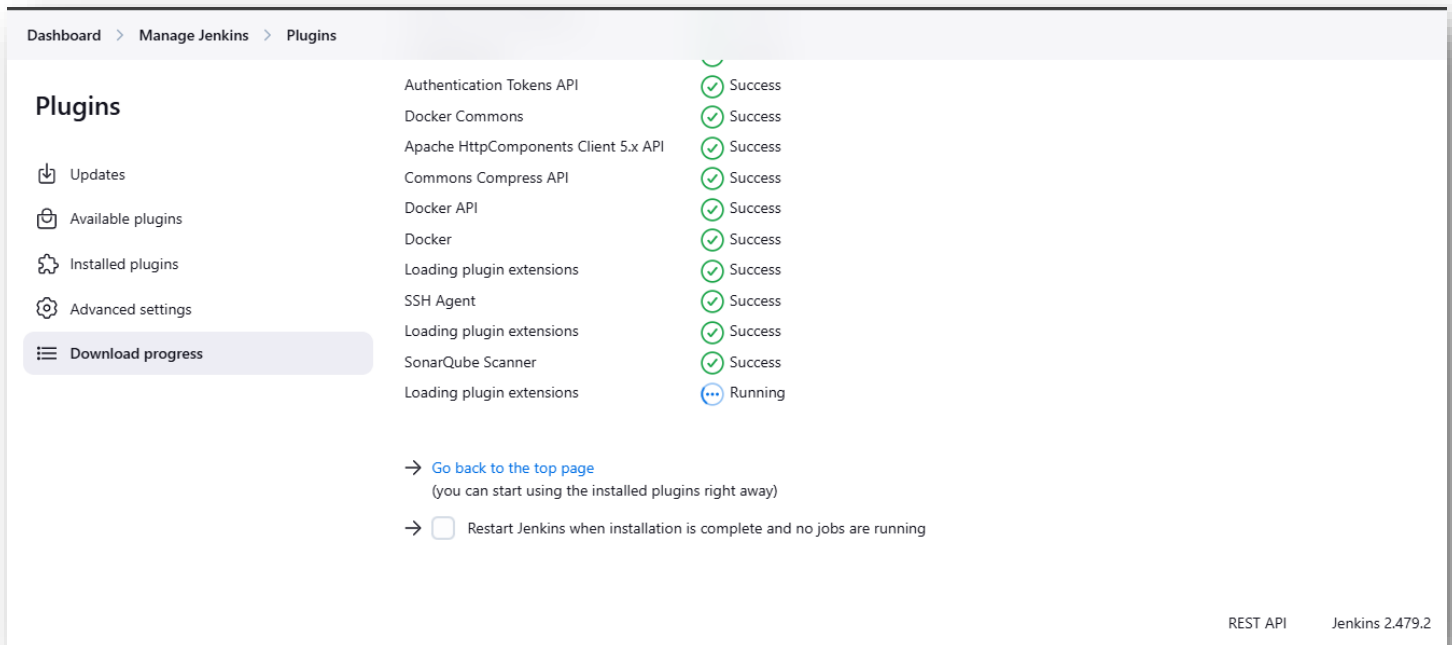
3 of 3 shown

© 2018-2024 SonarSource SA. All rights reserved. [Terms](#) [Pricing](#) [Privacy](#) [Cookie Policy](#) [Security](#) [Community](#) [Documentation](#) [Contact us](#) [Status](#) [About](#)

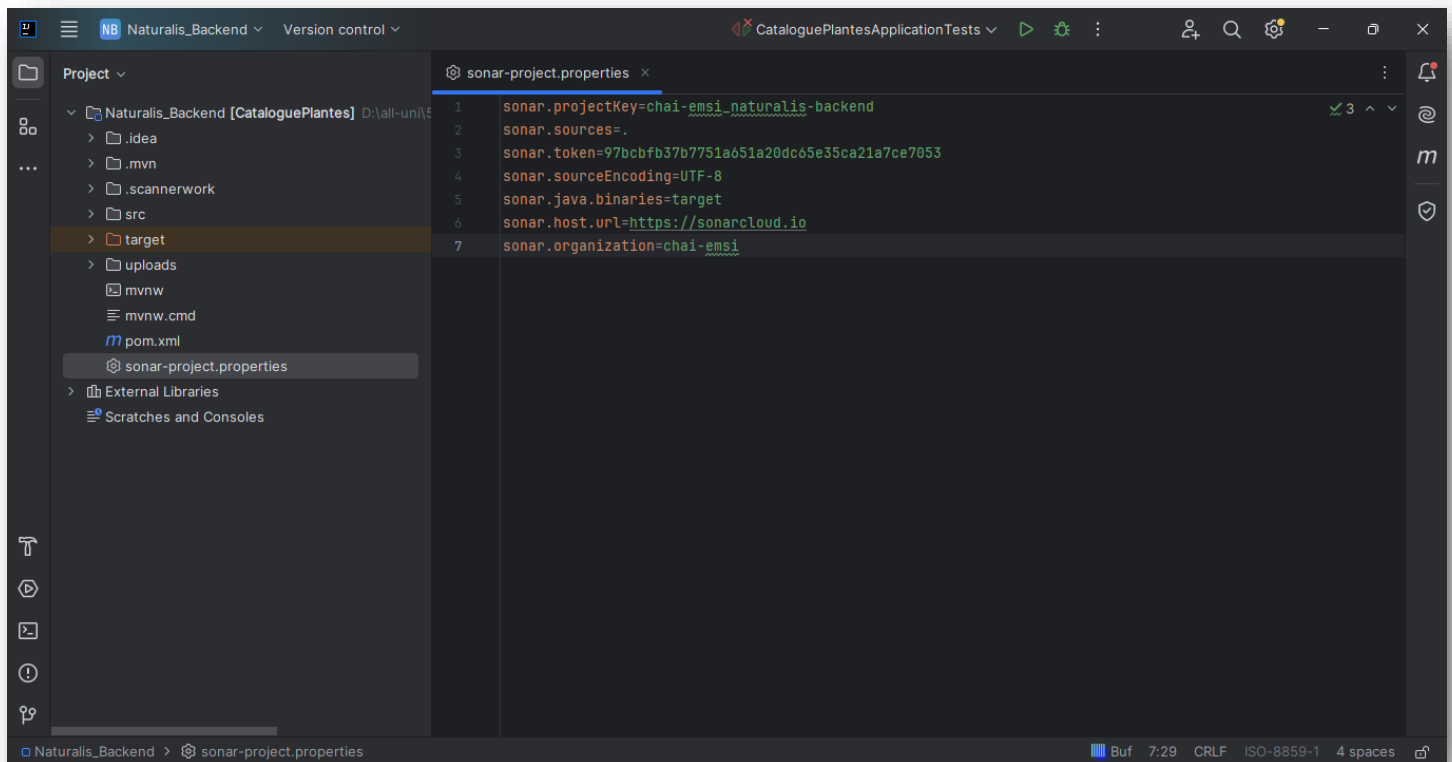


Partie CI/CD:

1. Installation de SonarQube Scanner dans Jenkins Plugins



2. Récupération de Token du projet



3. Configuration de SonarCloud dans Jenkins

Dashboard > Manage Jenkins > System >

Jenkins Credentials Provider: Jenkins

Domain

Global credentials (unrestricted) ▼

Kind

Secret text ▼

Scope ?

Global (Jenkins, nodes, items, all child items, etc) ▼

Secret

.....

ID ?

Save Apply

Dashboard > Manage Jenkins > System >

List of SonarQube installations

Name

SonarQube

Server URL

Default is http://localhost:9000

https://sonarcloud.io

Server authentication token

SonarQube authentication token. Mandatory when anonymous access is disabled.

- none - ▼

+ Add

Advanced ▼

Save Apply

4. Ajout de SonarScanner

Dashboard > Manage Jenkins > Tools

Add Gradle

SonarScanner for MSBuild installations

Add SonarScanner for MSBuild

SonarQube Scanner installations

Add SonarQube Scanner

Ant installations

Add Ant

Save Apply

Dashboard > Manage Jenkins > Tools

Add SonarQube Scanner

☰ **SonarQube Scanner** ✕

Name

SonarScanner

☒ Install automatically ?

☰ **Install from Maven Central** ✕


Version

SonarQube Scanner 6.2.1.4610 ▼

Add Installer ▼

Save Apply

5. Configuration de Jenkins Pipeline

 **Jenkins**

Search (CTRL+K) ? 1 Chaïma Rouita log out


Dashboard > All > New Item


New Item


Enter an item name

SonarCloudPipeline

Select an item type

 **Freestyle project**
Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.


 **Pipeline**
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.


 **Multi-configuration project**


OK

Dashboard > SonarCloudPipeline > Configuration

Configure

 General

 Advanced Project Options

 Pipeline

Pipeline

Definition

Pipeline script

Script ?

```
1 pipeline {
2   agent any
3   stages {
4     stage('Checkout') {
5       steps {
6         // Clone your project from your Git repository
7         git branch: 'main', url: 'https://github.com/Chaïmaert/Naturalis_Backend.git'
8       }
9     }
10    stage('Build') {
11      steps {
12        // Build your project (adjust command as per your build tool)
13        sh './mvnw clean install'
14      }
15    }
16    stage('SonarQube Analysis') {
17      environment {
18        // Provide the SonarCloud token (set in Jenkins credentials)
19      }
20    }
21  }
```

try sample Pipeline...

☒ Use Groovy Sandbox ?

Save Apply

6. Test du Pipeline

The screenshot shows the Jenkins web interface. At the top, there's a navigation bar with the Jenkins logo, a search bar, and user information. Below the navigation bar, the breadcrumb trail reads "Dashboard > SonarCloudPipeline > #10". The main content area is divided into two columns. The left column contains a list of links for various pipeline views: Status, Changes, Console Output, Edit Build Information, Timings, Git Build Data, Pipeline Overview, Pipeline Console, Thread Dump, Pause/resume, and Replay. The right column displays the build details for "#10 (Dec 20, 2024, 1:26:07 AM)". It shows a green checkmark icon, indicating a successful build. Below the icon, it states "Started by user Chaima Rouita" and "This run spent 4 ms waiting in the queue." The "Revision" is "23016c0f6bf416dc640b35b9809a2a2fef2479aa" and the "Repository" is "https://github.com/Chaimaert/Naturalis_Backend.git". The "git" icon is also present. Below the repository information, it says "refs/remotes/origin/main". The "No changes." message is displayed. On the right side of the build details, there are two buttons: "Add description" and "Keep this build forever".

```
Started by user Chaima Rouita
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in C:\ProgramData\Jenkins\.jenkins\workspace\SonarCloudPipeline
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Checkout)
[Pipeline] git
The recommended git tool is: NONE
No credentials specified
> git.exe rev-parse --resolve-git-dir C:\ProgramData\Jenkins\.jenkins\workspace\SonarCloudPipeline\.git # timeout=10
Fetching changes from the remote Git repository
> git.exe config remote.origin.url https://github.com/Chaimaert/Naturalis_Backend.git # timeout=10
Fetching upstream changes from https://github.com/Chaimaert/Naturalis_Backend.git
> git.exe --version # timeout=10
> git --version # 'git version 2.40.0.windows.1'
> git.exe fetch --tags --force --progress -- https://github.com/Chaimaert/Naturalis_Backend.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git.exe rev-parse "refs/remotes/origin/main^{commit}" # timeout=10
Checking out Revision 23016c0f6bf416dc640b35b9809a2a2fef2479aa (refs/remotes/origin/main)
> git.exe config core.sparsecheckout # timeout=10
> git.exe checkout -f 23016c0f6bf416dc640b35b9809a2a2fef2479aa # timeout=10
> git.exe branch -a -v --no-abbrev # timeout=10
> git.exe branch -D main # timeout=10
> git.exe checkout -b main 23016c0f6bf416dc640b35b9809a2a2fef2479aa # timeout=10
Commit message: "update"
First time build. Skipping changelog.
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Build)
[Pipeline] echo
Skipping Maven build or using a custom build tool for Windows.
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (SonarQube Analysis)
[Pipeline] withCredentials
Masking supported pattern matches of %SONAR_TOKEN%
[Pipeline] {
```