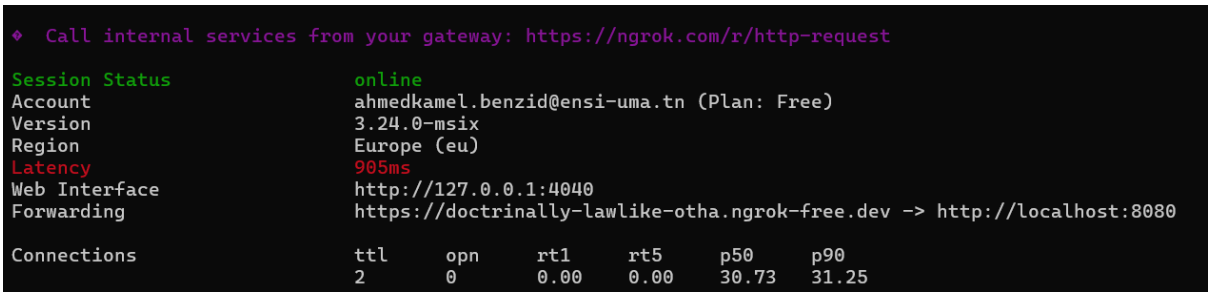
Mise en place d’un pipeline CI/CD

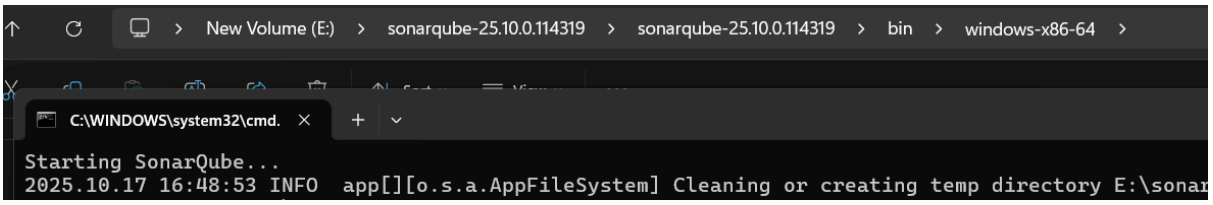
CONFIGURATION DE L'ENVIRONNEMENT

Utiliser ngrok pour créer une adresse publique



Objectif : Rendre Jenkins accessible depuis GitHub

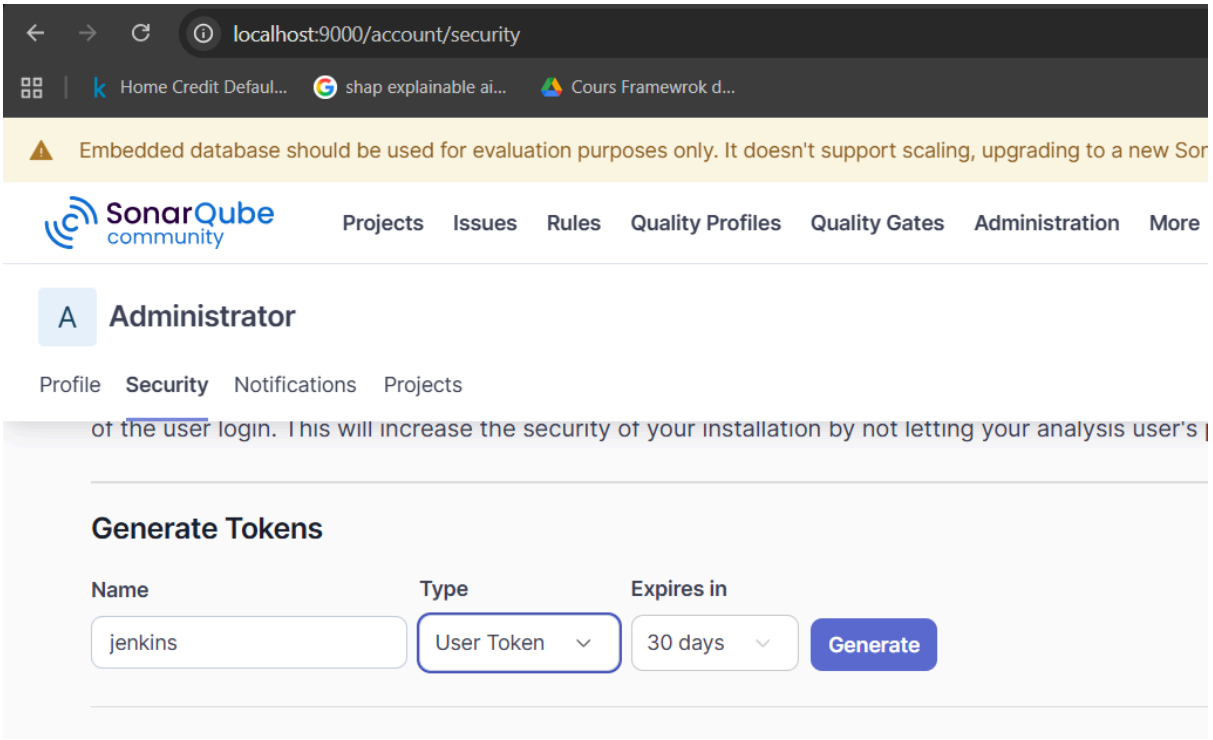
Démarrage SonarQube



Génération Token SonarQube

Objectif : Créer un token pour l'authentification Jenkins

Configuration :

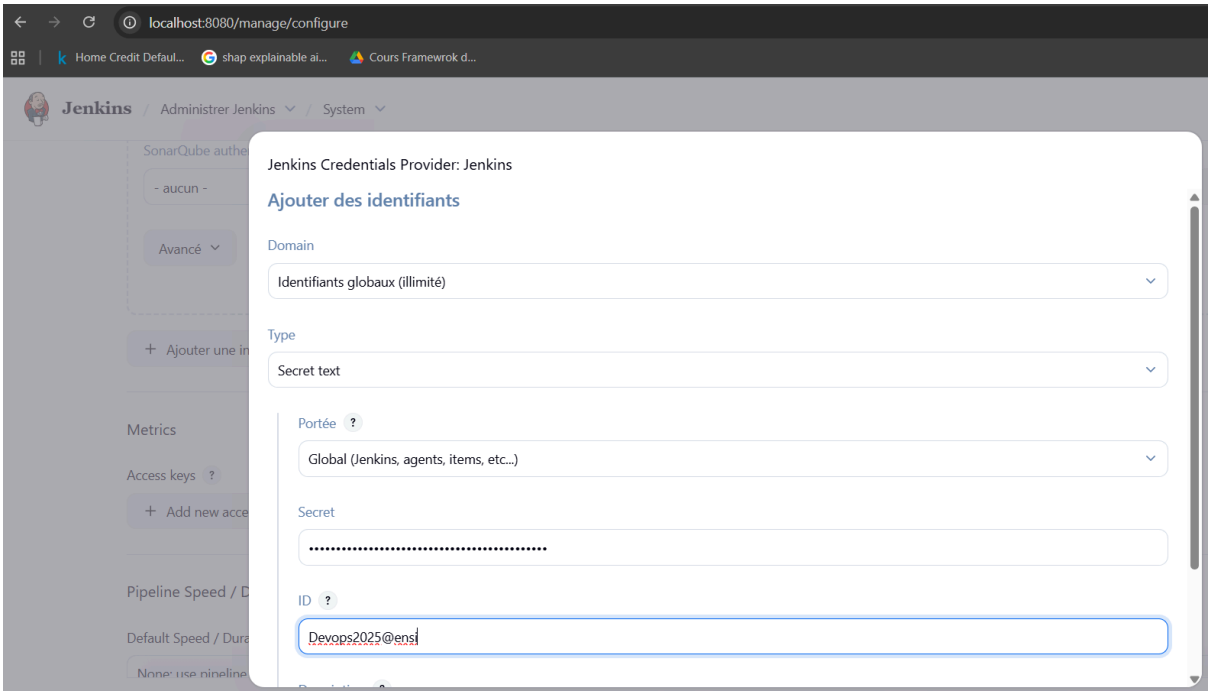


Nom : jenkins

Type : User Token

CONFIGURATION JENKINS

Identifiants SonarQube



Objectif : Stocker sécuriséement le token SonarQube

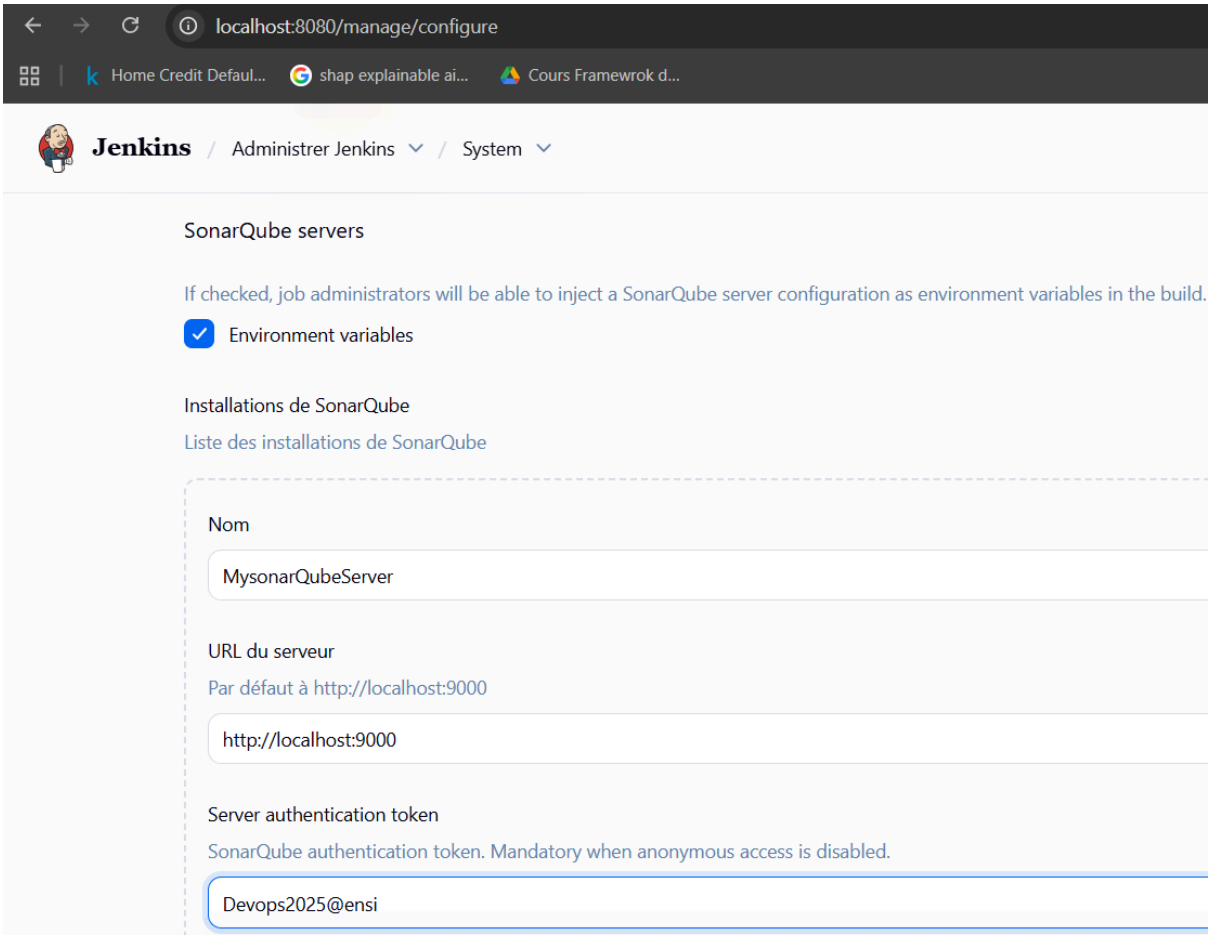
Configuration :

Type : Secret text

Portée : Globale

ID : Devops2025@ensl

Serveur SonarQube



Objectif : Intégrer SonarQube à Jenkins

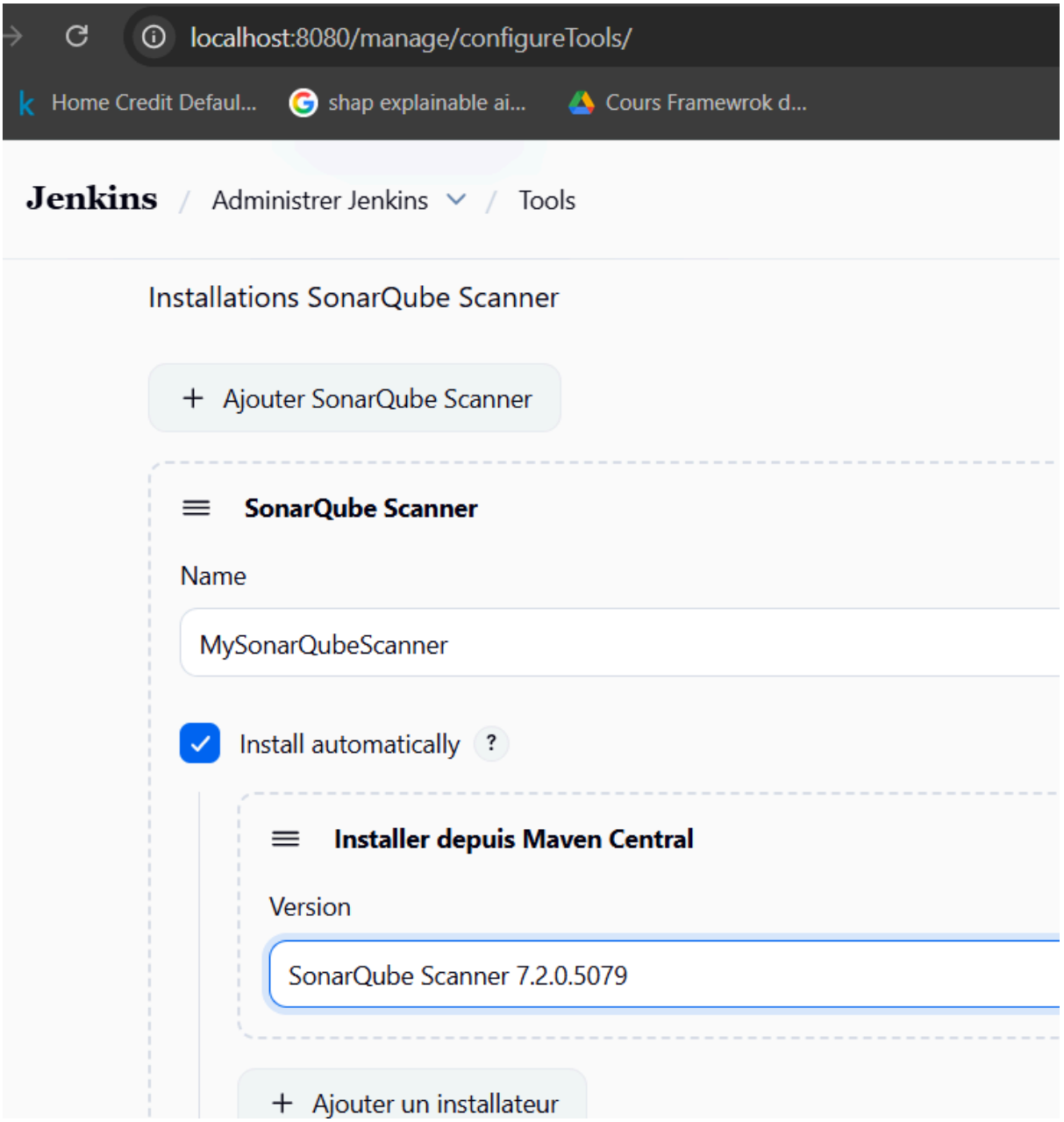
Configuration :

Nom : MysonarQubeServer

URL : http://localhost:9000

Token : Devops2025@ensi

Installation Scanner



Objectif : Installer SonarQube Scanner

Configuration :

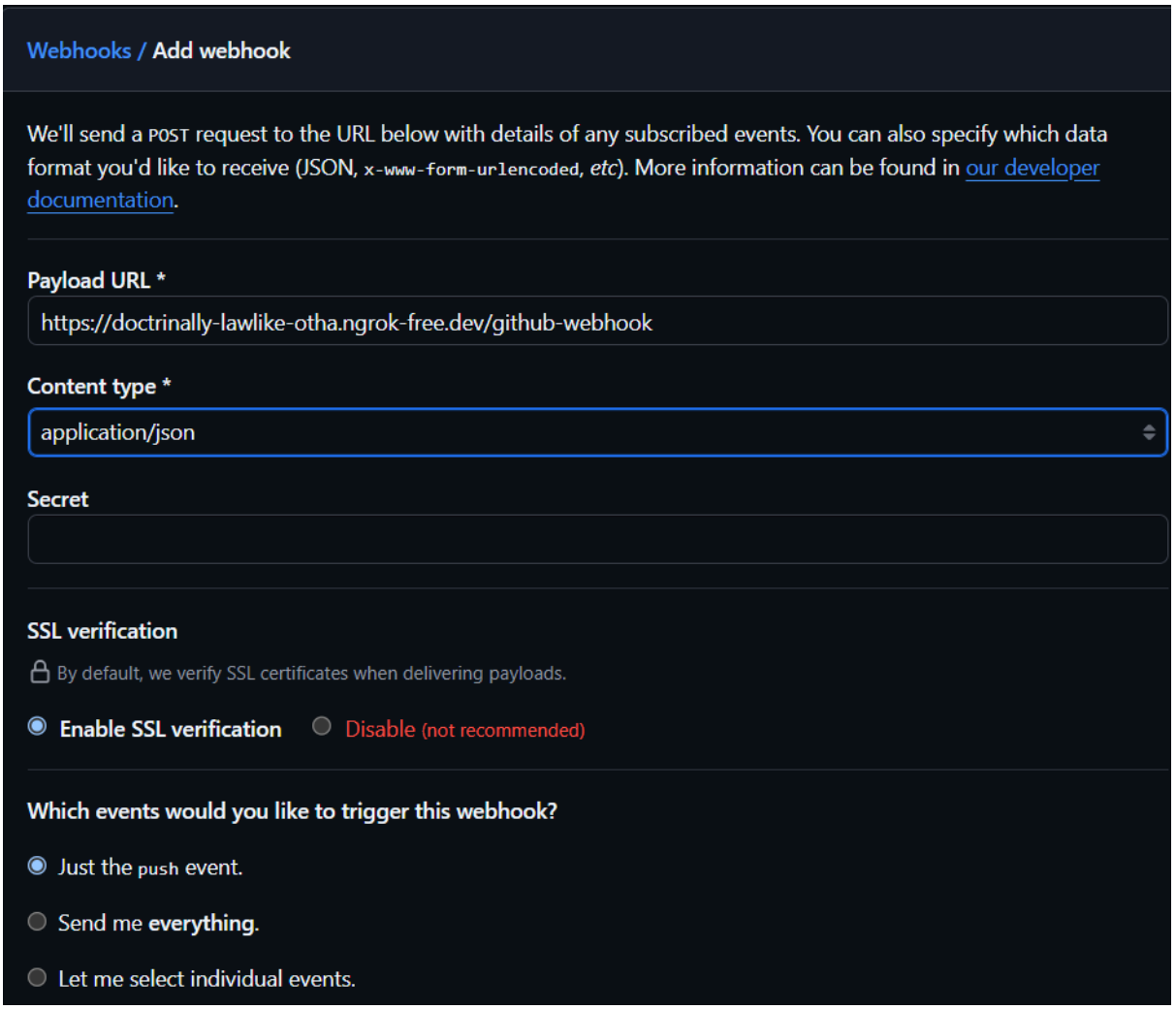
Nom : MySonarQubeScanner

Version : 7.2.0.5079

Source : Maven Central

INTÉGRATION GITHUB

Webhook GitHub



Objectif : Déclencher automatiquement les builds

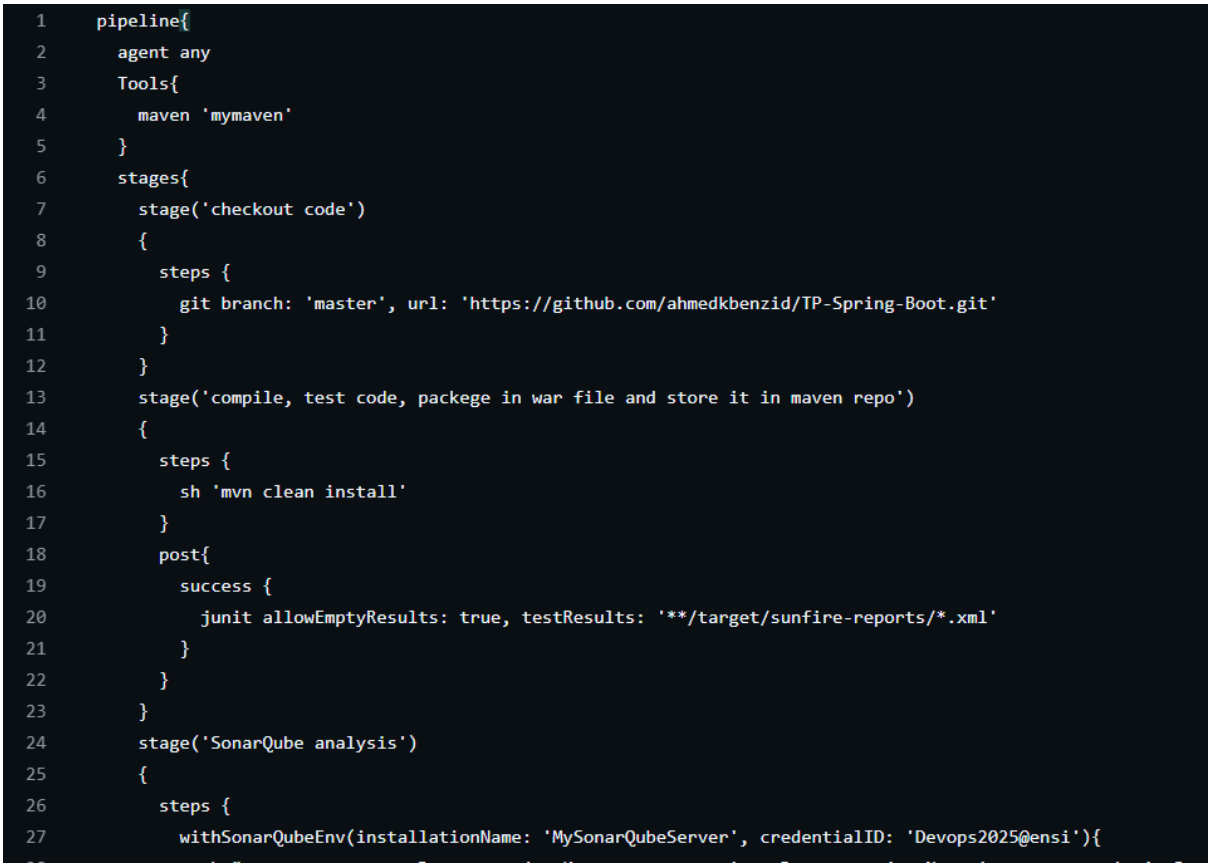
Configuration :

URL : https://doctrinally-lawlike-otha.ngrok-free.dev/github-webhook

Content type : JSON

Événements : Push events uniquement

Script Groovy



```
pipeline{
  agent any
  Tools{
    maven 'mymaven'
  }
  stages{
    stage('checkout code')
    {
      steps {
        git branch: 'master', url: 'https://github.com/ahmedkbenzid/TP-Spring-Boot.git'
      }
    }
    stage('compile, test code, packege in war file and store it in maven repo')
    {
      steps {
        sh 'mvn clean install'
      }
      post{
        success {
          junit allowEmptyResults: true, testResults: '**/target/sunfire-reports/*.xml'
        }
      }
    }
    stage('SonarQube analysis')
    {
      steps {
        withSonarQubeEnv(installationName: 'MySonarQubeServer', credentialID: 'Devops2025@ensi'){
```

Objectif : Définir le pipeline d'automatisation

Étapes principales :

Checkout code depuis GitHub

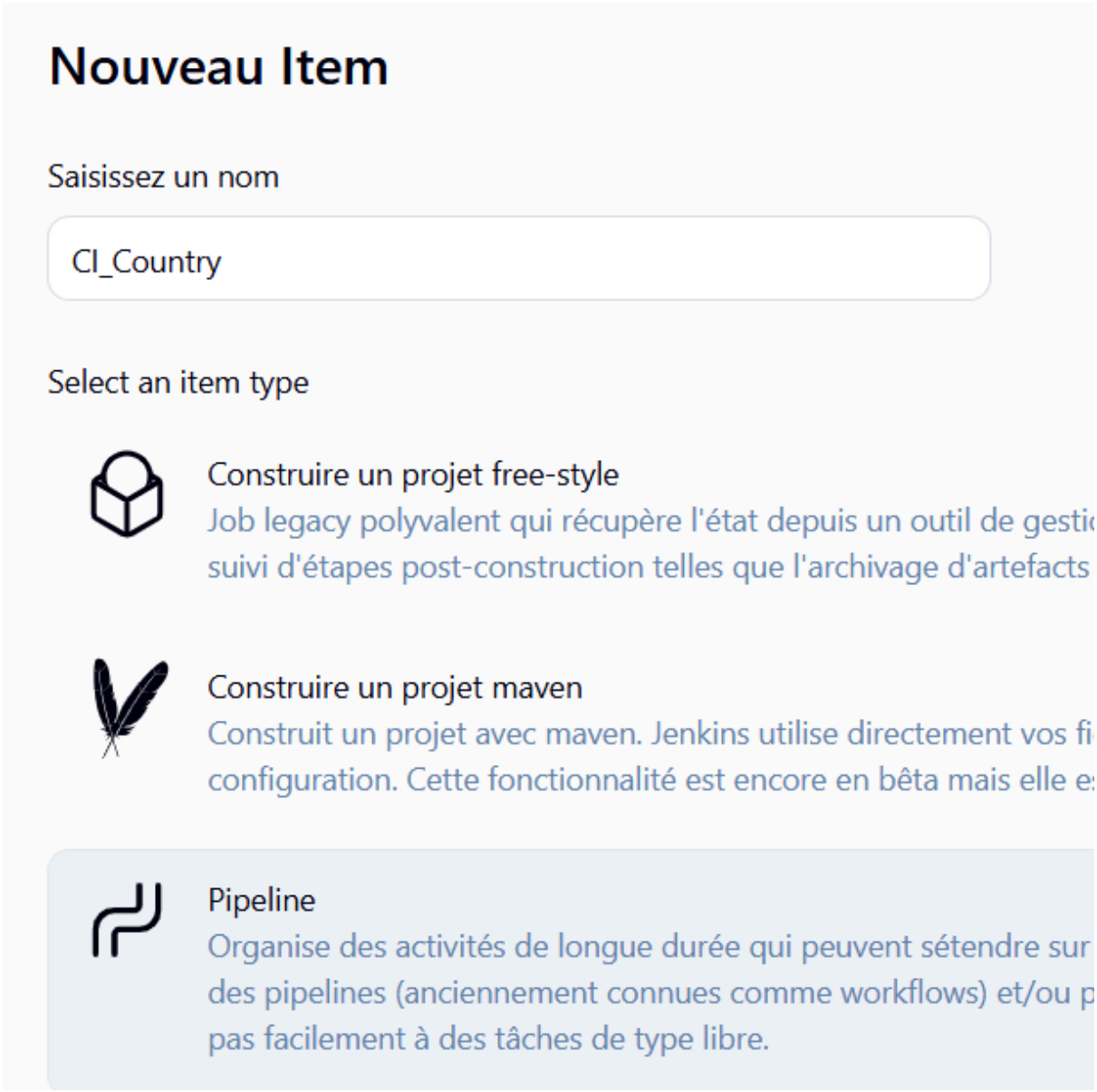
Compilation, tests avec Maven

Analyse SonarQube

Rapports JUnit

CONFIGURATION PROJET

Création Projet



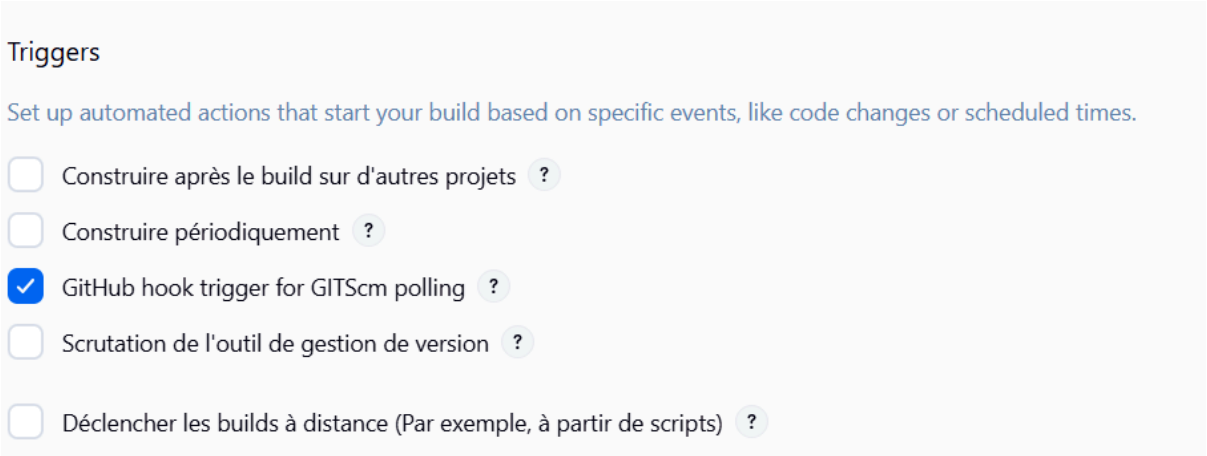
Objectif : Créer le projet Jenkins

Configuration :

Nom : CI_Country

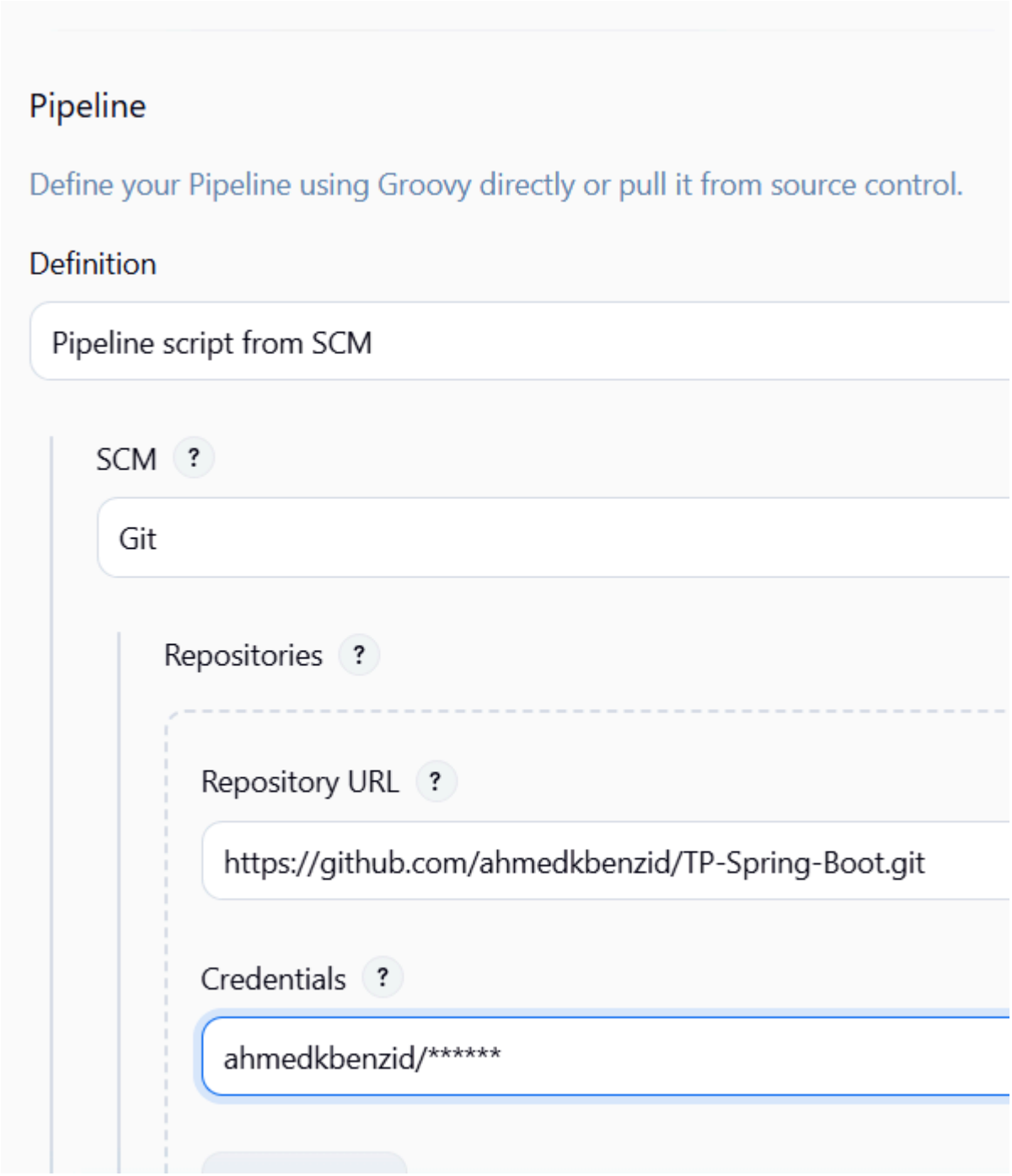
Type : Projet Maven

Triggers GitHub



Objectif : Activer le déclenchement automatique

Configuration



Objectif : Lier le repository Git

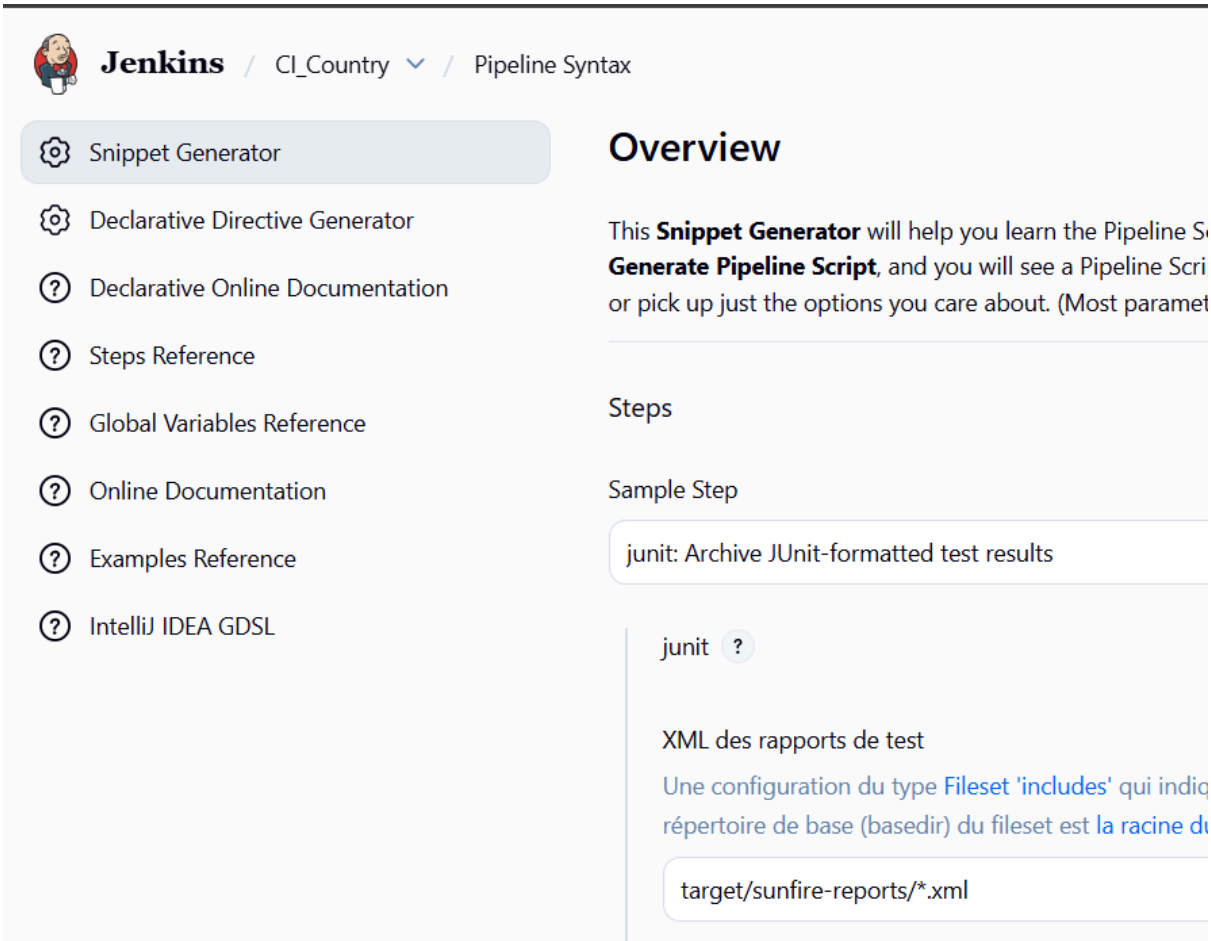
Configuration :

URL : https://github.com/ahmedkbenzid/TP-Spring-Boot.git

Credentials : ahmedkbenzid/*****

EXÉCUTION ET RÉSULTATS

Syntaxe Pipeline



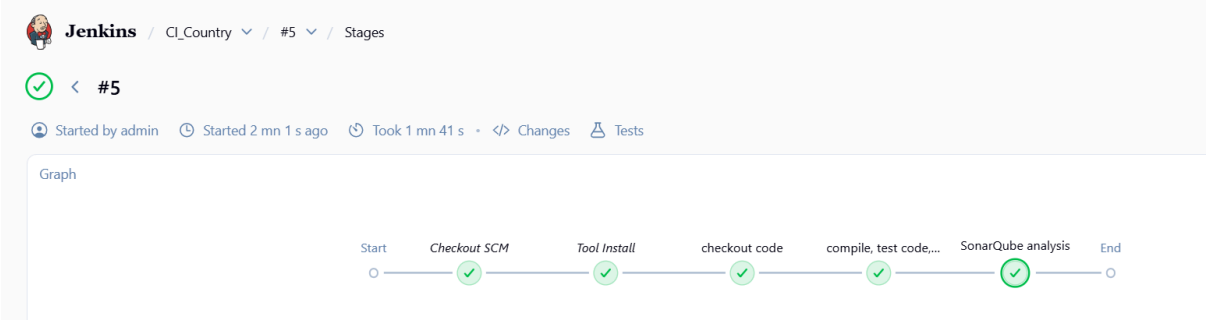
Objectif : Configurer les rapports de tests

Configuration :

Outil : Snippet Generator

Rapports JUnit : target/sunfire-reports/*.xml

Exécution Pipeline



Objectif : Visualiser l'exécution du build

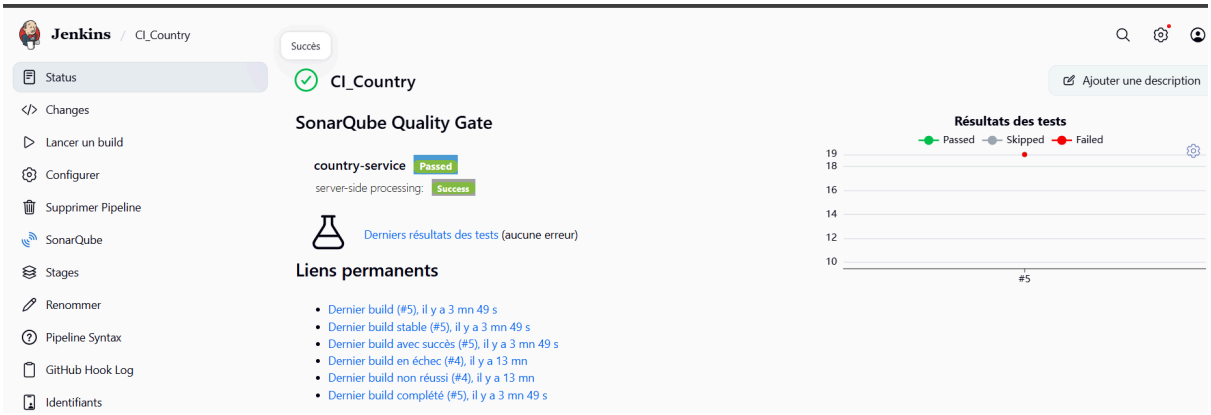
Résultats :

Build #5 réussi

Durée : 1 minute 41 secondes

Étapes : Checkout SCM → Tool Install → Compilation/Tests → SonarQube analysis

Résultats Finaux



Objectif : Analyser la qualité du code

Résultats :

SonarQube Quality Gate passé

Tests : Passed (aucune erreur)