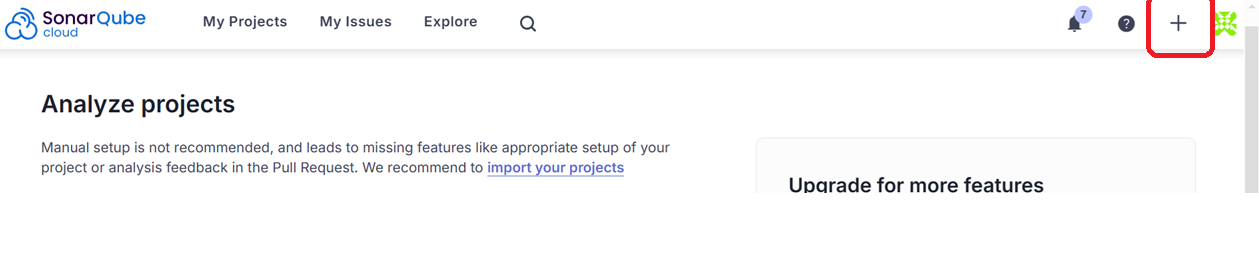
Signup and SetUp – SonarQube Cloud account using GitHub Objective: In this lab, you will signup SonarQube Cloud account using GitHub Creating a SonarQube Cloud account using GitHub 1.

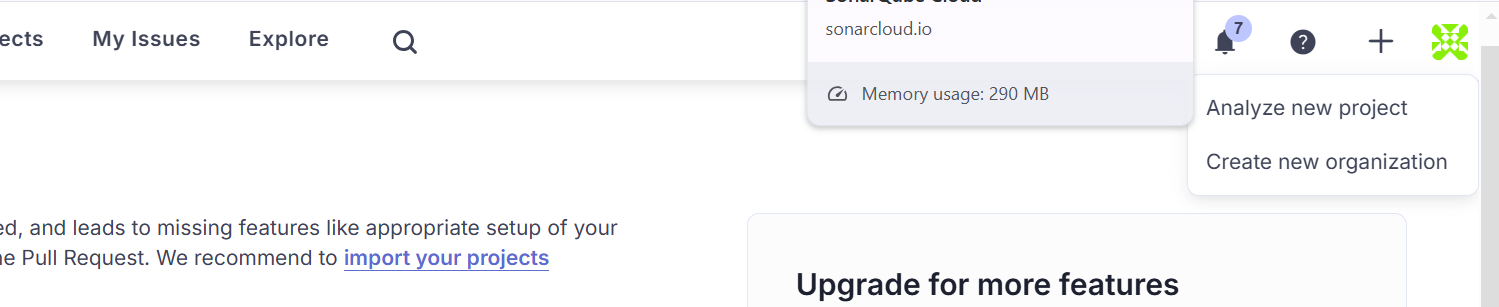
Go to SonarQube cloud website

https://www.sonarsource.com/products/sonarcloud/

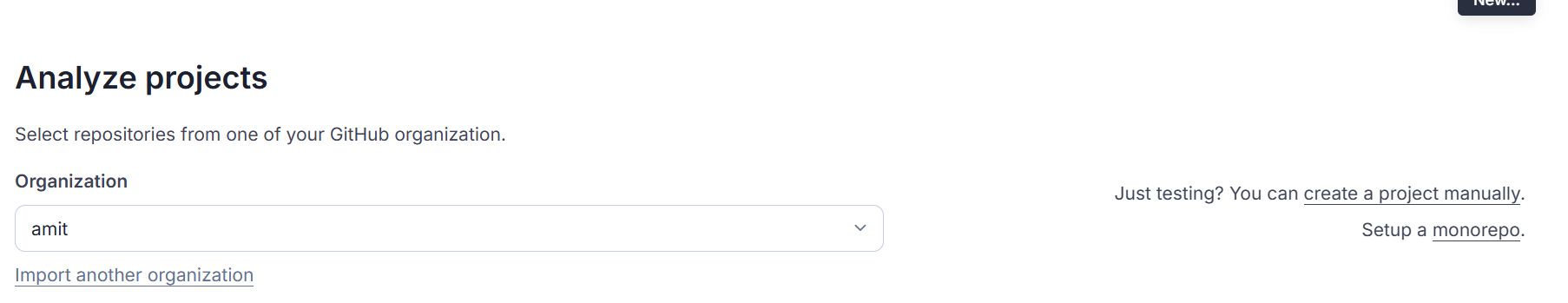
1. Click on sonarcloud + icon



Click on Analyze new project

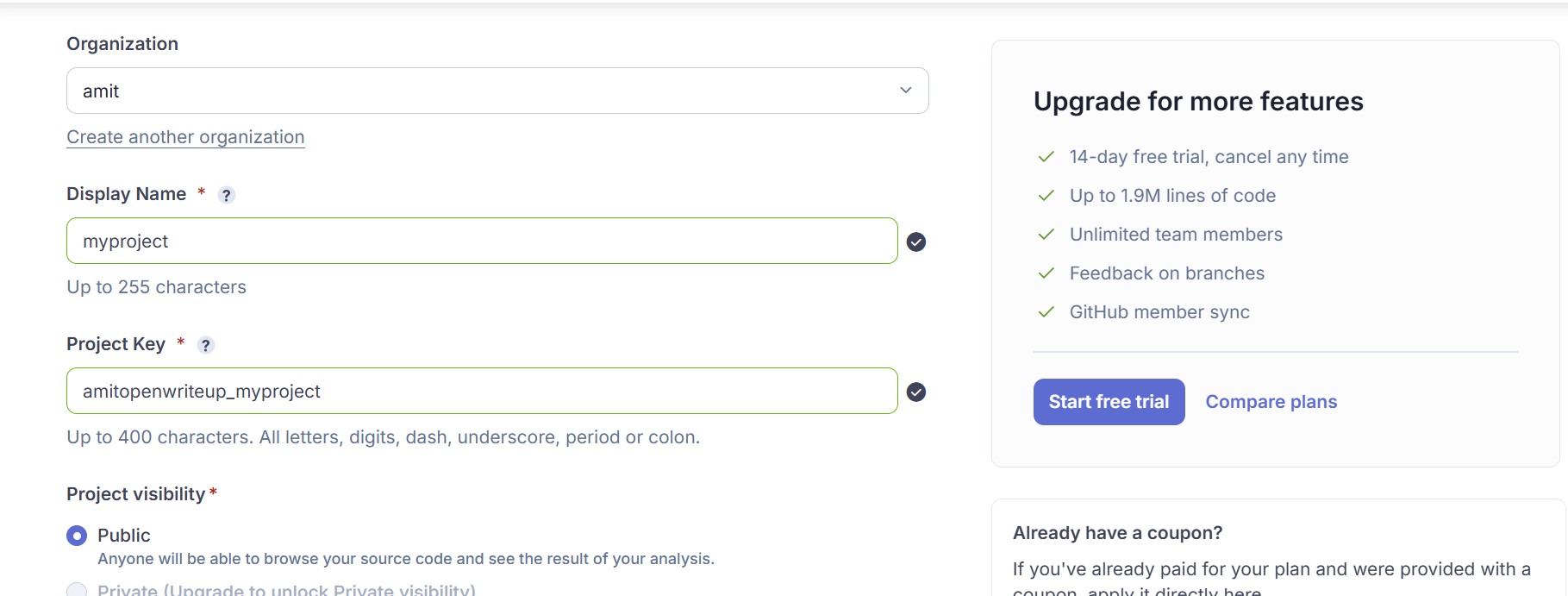


Click on Create project manually

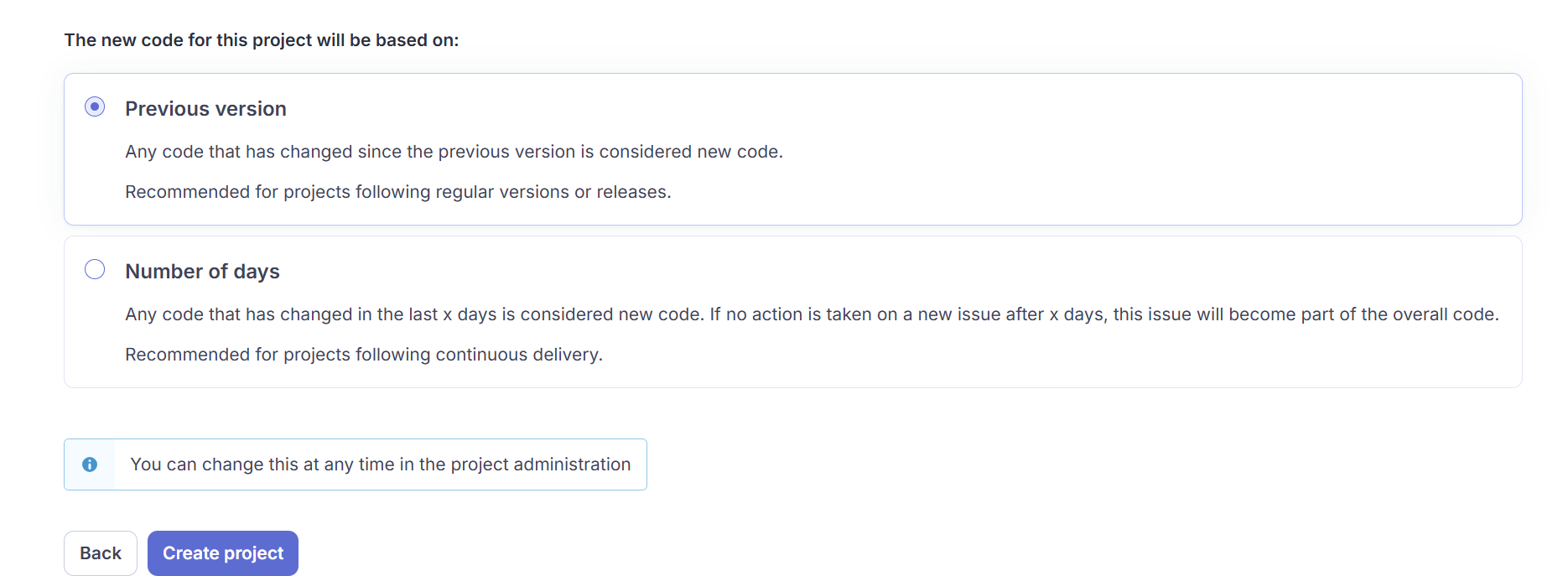


Provide organization and project

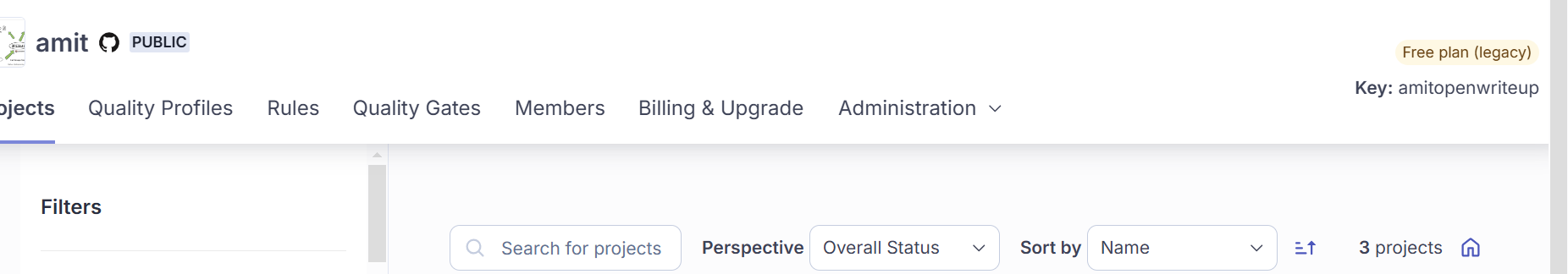
**Note: Note down the project key**



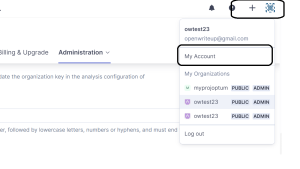
Select option and create project

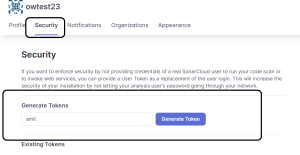


**Note down the organization key**



**get token**

[](https://usercontent.one/wp/www.openwriteup.com/wp-content/uploads/2023/09/an5.png?media=1728024675)

[](https://usercontent.one/wp/www.openwriteup.com/wp-content/uploads/2023/09/an6.png?media=1728024675)

generate the token and note it down

Add Jenkins users as part of sudoers group

vi /etc/sudoers

jenkins ALL=(ALL) NOPASSWD: ALL

Modify the below code, in the last stage, highlighter in Red, Provide organization key, project key and token

pipeline {

agent any

stages {

stage('Checkout SCM') {

steps {

checkoutSCM()

}

}

stage('Compiling and Running Test Cases') {

steps {

compileAndRunTests()

}

}

stage('Generating Cucumber Reports') {

steps {

generateCucumberReports()

}

}

stage('Creating Package') {

steps {

createPackage()

}

}

stage('Adding Generate Report') {

steps {

addGenerateReport()

}

}

stage('Install SonarQube CLI') {

steps {

installSonarQubeCLI()

}

}

stage('Analyzing Code Quality') {

steps {

analyzeCodeQuality()

}

}

}

}

def checkoutSCM() {

checkout scm: [$class: 'GitSCM', branches: [[name: '\*/master']],

userRemoteConfigs: [[url: 'https://github.com/hellokaton/java11-examples.git']]]

}

def compileAndRunTests() {

sh 'mvn clean'

sh 'mvn compile'

sh 'mvn test'

}

def generateCucumberReports() {

script {

sh 'mvn verify'

}

}

def createPackage() {

sh 'mvn package'

}

def addGenerateReport() {

sh 'mvn verify'

}

def installSonarQubeCLI() {

sh '''

sudo wget -O sonar-scanner.zip https://binaries.sonarsource.com/Distribution/sonar-scanner-cli/sonar-scanner-cli-5.0.1.3006-linux.zip

sudo unzip -o -q sonar-scanner.zip

sudo rm -rf /opt/sonar-scanner

sudo mv --force sonar-scanner-5.0.1.3006-linux /opt/sonar-scanner

sudo sh -c 'echo "#/bin/bash \nexport PATH=\\\"$PATH:/opt/sonar-scanner/bin\\\"" >/etc/profile.d/sonar-scanner.sh'

sudo chmod +x /opt/sonar-scanner/bin/sonar-scanner

. /etc/profile.d/sonar-scanner.sh

'''

}

def analyzeCodeQuality() {

sh '''

/opt/sonar-scanner/bin/sonar-scanner \

-Dsonar.projectKey=amitopenwriteup\_myproject \

-Dsonar.organization=amitopenwriteup \

-Dsonar.qualitygate.wait=false \

-Dsonar.qualitygate.timeout=300 \

-Dsonar.sources=src/main/java/ \

-Dsonar.java.binaries=target/classes \

-Dsonar.host.url=https://sonarcloud.io \

-Dsonar.login=8b3186250fc79f1aabd908035410cda5021290a2

'''

}

Create a job, pipeline and copy above code and run

