**目录**

[**第1章** **SonarQube简介** 3](#_Toc53133512)

[**1.1** **SonarQube解决的问题** 3](#_Toc53133513)

[**1.2** **特性** 3](#_Toc53133514)

[**第2章** **准备工作** 4](#_Toc53133515)

[**2.1** **JDK** 4](#_Toc53133516)

[**2.2** **SonarQube(LTS)** 4](#_Toc53133517)

[**2.3** **PostgreSQL** 4](#_Toc53133518)

[**2.4** **Plugins** 4](#_Toc53133519)

[**2.5** **Visual Studio** 4](#_Toc53133520)

[**2.6** **Python** 5](#_Toc53133521)

[**第3章** **PostgreSQL安装** 5](#_Toc53133522)

[**3.1** **PostgreSQL for Windows** 5](#_Toc53133523)

[**3.1.1** **下载安装包** 5](#_Toc53133524)

[**3.1.2** **安装PostgreSQL** 6](#_Toc53133525)

[**3.2** **创建sonar数据库和用户** 6](#_Toc53133526)

[**3.2.1** **创建sonar用户** 7](#_Toc53133527)

[**3.2.2** **创建sonar数据库** 9](#_Toc53133528)

[**第4章** **SonarQube（LTS）安装** 12](#_Toc53133529)

[**4.1** **SonarQube安装** 12](#_Toc53133530)

[**4.2** **配置PostgreSQL连接信息** 12](#_Toc53133531)

[**4.3** **启动** 13](#_Toc53133532)

[**第5章** **MSBuild** 15](#_Toc53133533)

[**5.1** **安装Visual Studio编译器** 15](#_Toc53133534)

[**5.2** **配置MSBuild** 15](#_Toc53133535)

[**第6章** **安装第三方分析器** 17](#_Toc53133536)

[**6.1** **ReSharper** 17](#_Toc53133537)

[**6.1.1** **安装SonarQube ReSharper插件** 17](#_Toc53133538)

[**6.1.2** **配置ReSharper插件** 18](#_Toc53133539)

[**6.1.3** **将ReSharper规则添加到质量配置文件** 19](#_Toc53133540)

[**6.2** **FxCop** 20](#_Toc53133541)

[**6.2.1** **Visual Studio中安装FxCop分析器** 20](#_Toc53133542)

[**6.2.2** **安装SonarQube FxCop插件** 21](#_Toc53133543)

[**6.2.3** **配置FxCop插件** 22](#_Toc53133544)

[**6.2.4** **将FxCop规则添加到质量配置文件** 23](#_Toc53133545)

[**6.3** **Dependency-Check** 24](#_Toc53133546)

[**6.3.1** **安装SonarQube Dependency-Check插件** 24](#_Toc53133547)

[**6.3.2** **配置Dependency-Check插件** 24](#_Toc53133548)

[**6.3.3** **查看Dependency-Check报告** 25](#_Toc53133549)

[**6.4** **Rules Compliance Index(RCI)** 26](#_Toc53133550)

[**6.4.1** **安装SonarQube RCI插件** 26](#_Toc53133551)

[**6.4.2** **计算RCI指数** 26](#_Toc53133552)

[**6.4.3** **配置度量标准** 26](#_Toc53133553)

[**6.4.4** **查看代码的RCI** 27](#_Toc53133554)

[**6.5** **SoftVis3D** 27](#_Toc53133555)

[**6.5.1** **安装SonarQube SoftVis3D插件** 28](#_Toc53133556)

[**6.5.2** **查看SoftVis3D视图** 28](#_Toc53133557)

[**6.6** **Csv issue report** 29](#_Toc53133558)

[**6.6.1** **安装SonarQube CSV插件** 29](#_Toc53133559)

[**6.6.2** **导出Csv issue报告** 30](#_Toc53133560)

[**6.7** **CNES Report** 30](#_Toc53133561)

[**6.7.1** **安装SonarQube** CNES**插件** 31](#_Toc53133562)

[**6.7.2** **导出CNES报告** 31](#_Toc53133563)

[**6.8** **Code Cracker** 32](#_Toc53133564)

[**6.8.1** **Visual Studio中安装Code Cracker分析器** 32](#_Toc53133565)

[**6.8.2** **安装SonarQube CodeCracker插件** 33](#_Toc53133566)

[**6.8.3** **将CodeCracker规则添加到质量配置文件** 34](#_Toc53133567)

1. **SonarQube简介**

Sonar是一个用于代码质量管理的开源平台，用于管理源代码的质量，可以从七个维度检测代码质量。通过插件形式，可以支持包括java,C#,C/C++,PL/SQL,Cobol,JavaScrip,Groovy等等二十几种编程语言的代码质量管理与检测。SonarQube能够提供对代码的一整套检查扫描和分析功能，拥有一套服务器端程序，然后再通过客户端或者别的软件的插件的形式完成对各开发环境和软件的支持。

1. **SonarQube解决的问题**
2. 不遵循代码标准：sonar可以通过PMD,CheckStyle,Findbugs等等代码规则检测工具规范代码编写。
3. 潜在的缺陷：sonar可以通过PMD,CheckStyle,Findbugs等等代码规则检测工具检测出潜在的缺陷。
4. 糟糕的复杂度分布：文件、类、方法等，如果复杂度过高将难以改变，这会使得开发人员难以理解它们。且如果没有自动化的单元测试，对于程序中的任何组件的改变都将可能导致需要全面的回归测试。
5. 重复：显然程序中包含大量复制粘贴的代码是质量低下的，sonar可以展示 源码中重复严重的地方。
6. 注释不足或者过多：没有注释将使代码可读性变差，特别是当不可避免地出现人员变动时，程序的可读性将大幅下降。而过多的注释又会使得开发人员将精力过多地花费在阅读注释上，亦违背初衷。
7. 缺乏单元测试：sonar可以很方便地统计并展示单元测试覆盖率。
8. 糟糕的设计：通过sonar可以找出循环，展示包与包、类与类之间的相互依赖关系，可以检测自定义的架构规则。通过sonar可以管理第三方的jar包，可以利用LCOM4检测单个任务规则的应用情况，检测耦合。
9. **特性**
   1. 编写干净代码：SonarQube提供了源代码的整体健康的概述，更重要的是可以高亮显示在代码中发现的问题。
   2. 检测错误：SonarQube提出的问题是明显错误的代码，或是不会给出预期行为的代码。如包括空指针，内存泄漏和逻辑错误等。
   3. 多语言：SonarQube提供了各种主要编程语言的代码分析器：如C/C ++，JavaScript，C＃，Java，COBOL，PL /SQL，PHP，ABAP，VB.NET，Python，RPG，Flex，Objective-C，Swift等。
   4. DevOps集成
   5. 质量管控
10. **准备工作**
11. **JDK**

SonarQube扫描程序需要JVM的版本8或11，而SonarQube服务器需要的版本11。官方不支持Java 11以外的版本，所以运行SonarQube的唯一先决条件是在计算机上安装Java（Oracle JRE 11或OpenJDK 11）。

Version: jdk-11.0.5\_windows-x64

下载地址：https://www.oracle.com/java/technologies/javase-jdk11-downloads.html

1. **SonarQube(LTS)**

SonalQube Long term support Community Edition

Version:7.9.4。

下载地址：https://www.sonarqube.org/downloads/

1. **PostgreSQL**

SonarQube7.9及以后的版本不再支持MySQL，支持Oracle、PostgreSQL 和Microsoft SQLServer。

EDB Postgre Advanced Server

Version：12.3.4

下载地址：https://www.enterprisedb.com/downloads/postgres-postgresql-downloads

1. **Plugins**

resharper-clt-plugin-1.2.0.202013.jar

sonar-fxcop-plugin-1.4.1.jar

sonar-dependency-check-plugin-2.0.5.jar

sonar-rci-plugin-1.0.1.jar

sonar-softvis3d-plugin-1.1.0.jar

sonar-cnes-report-3.2.2.jar

codecrackercsharp-plugin-1.1.0.jar

下载地址：<https://www.sonarplugins.com/>

sonar-csv-export-plugin-0.4.1.jar

下载地址：<https://github.com/VillageChief/sonarqube-csv-export-plugin>

1. **Visual Studio**

Version：Enterprise 2019

下载地址：<https://visualstudio.microsoft.com/zh-hans/downloads/>

1. **Python**

Version：3.7.4 amd64

下载地址：<https://www.python.org/downloads/windows/>

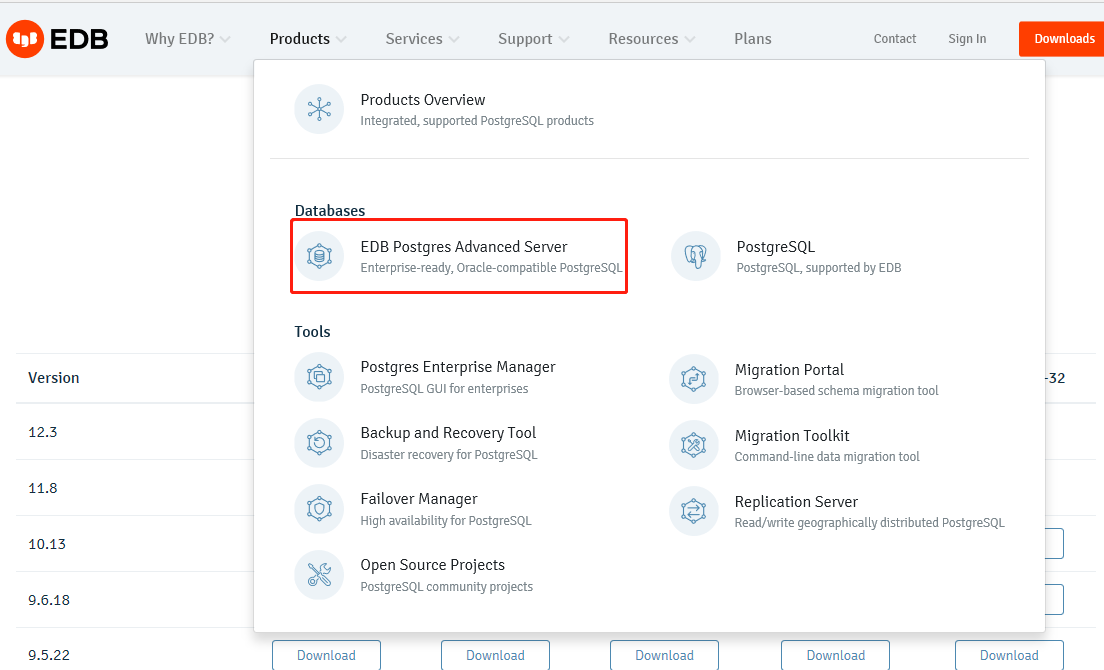
1. **PostgreSQL安装**

这里使用 EnterpriseDB 来下载安装，EnterpriseDB 是全球唯一一家提供基于 PostgreSQL 企业级产品与服务的厂商。

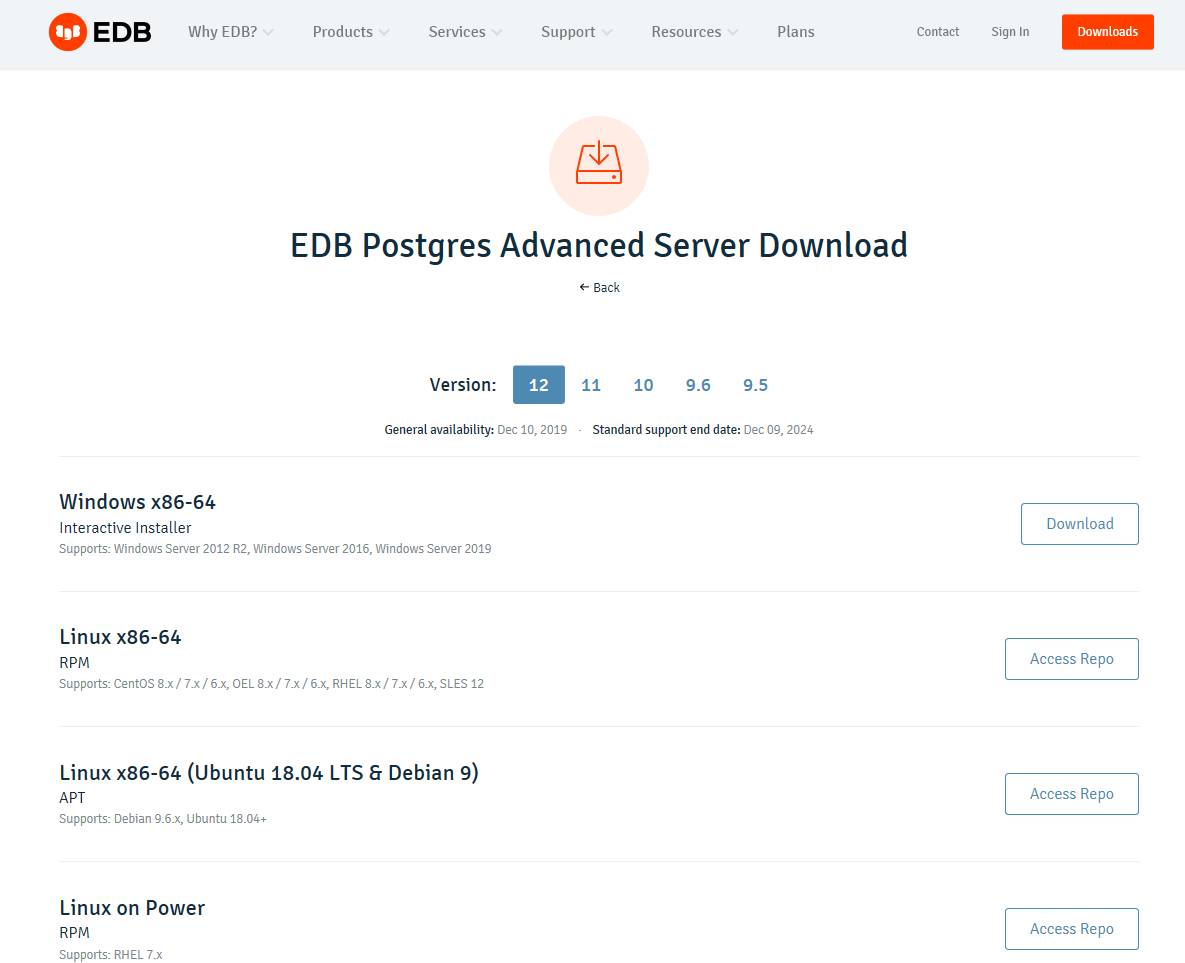
1. **PostgreSQL for Windows**

官网地址：https://www.enterprisedb.com/downloads/postgres-postgresql-downloads

1. **下载安装包**
2. 进入官网后，点击【Products】，选择【EDB Postgre Advanced Server】



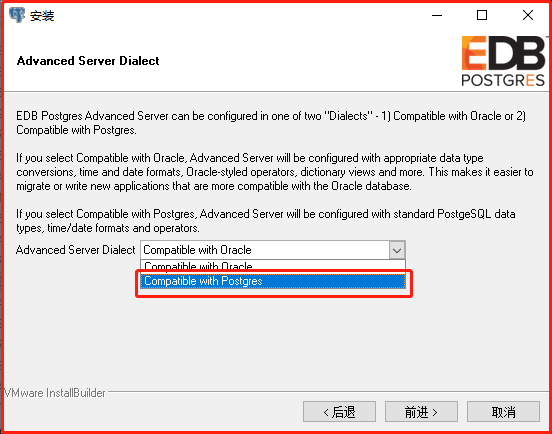
1. 点击【Download】，根据Version选择所需要的操作系统版本，进行下载。



1. **安装PostgreSQL**

安装步骤：https://www.runoob.com/postgresql/windows-install-postgresql.html

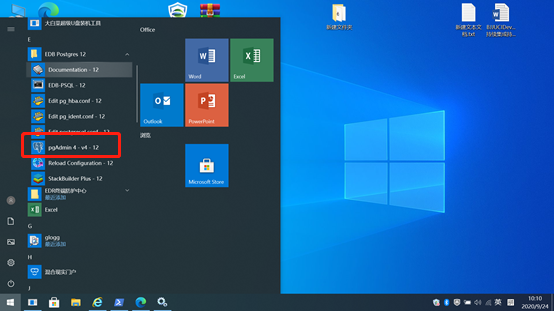
根据此安装步骤进行安装，需要注意的是在选择完安装路径之后，在【Advanced Server Dialect】页面进行选择时，选择【Compatible with Postgres】。



1. **创建sonar数据库和用户**

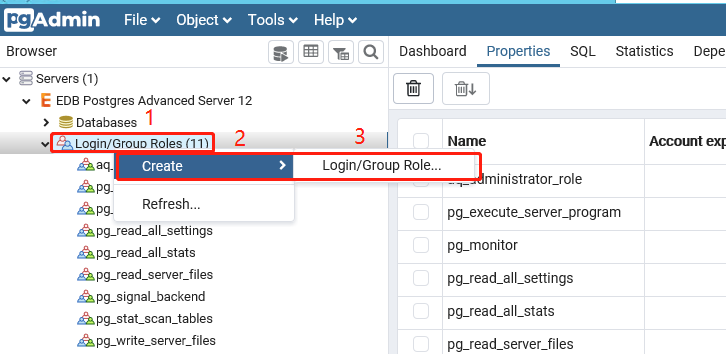
安装完成之后，在【开始】菜单栏中点击【pgAdmin4-v4-12】打开数据库。在PostgreSQL数据库中创建sonar数据库和用户。

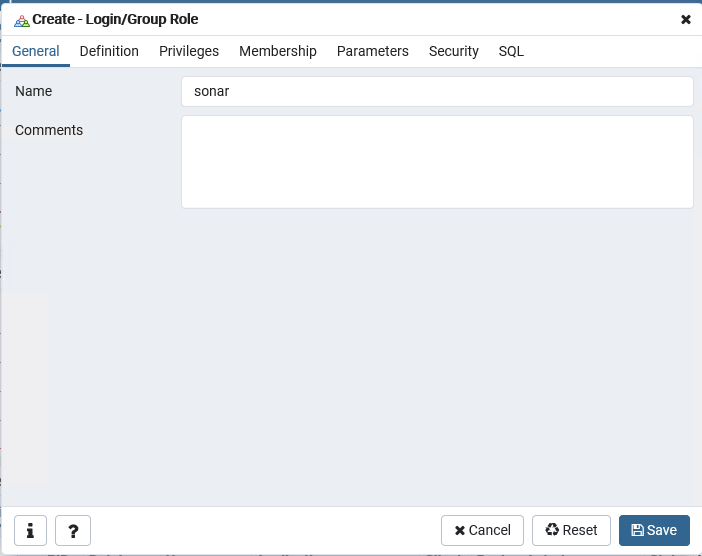
参考文档：https://www.cnblogs.com/Simple-Small/p/12882948.html

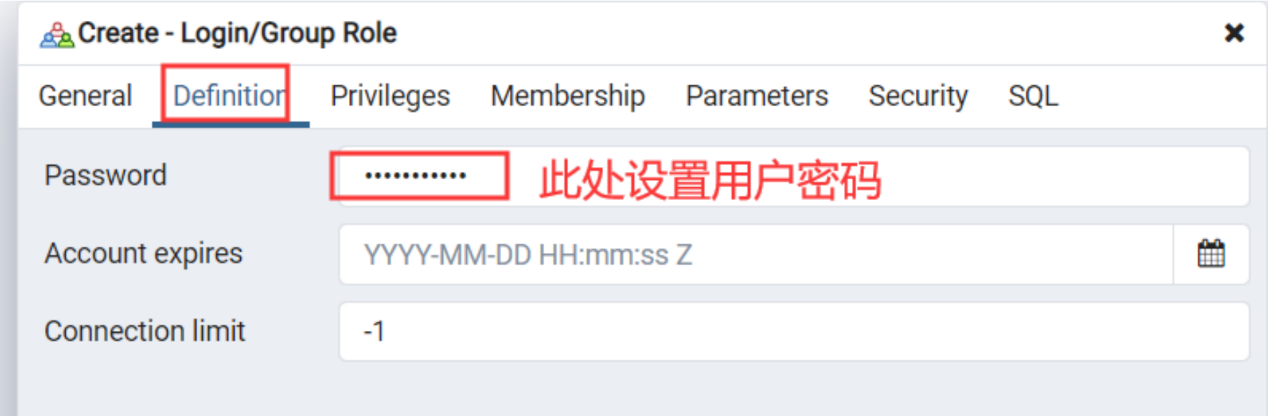


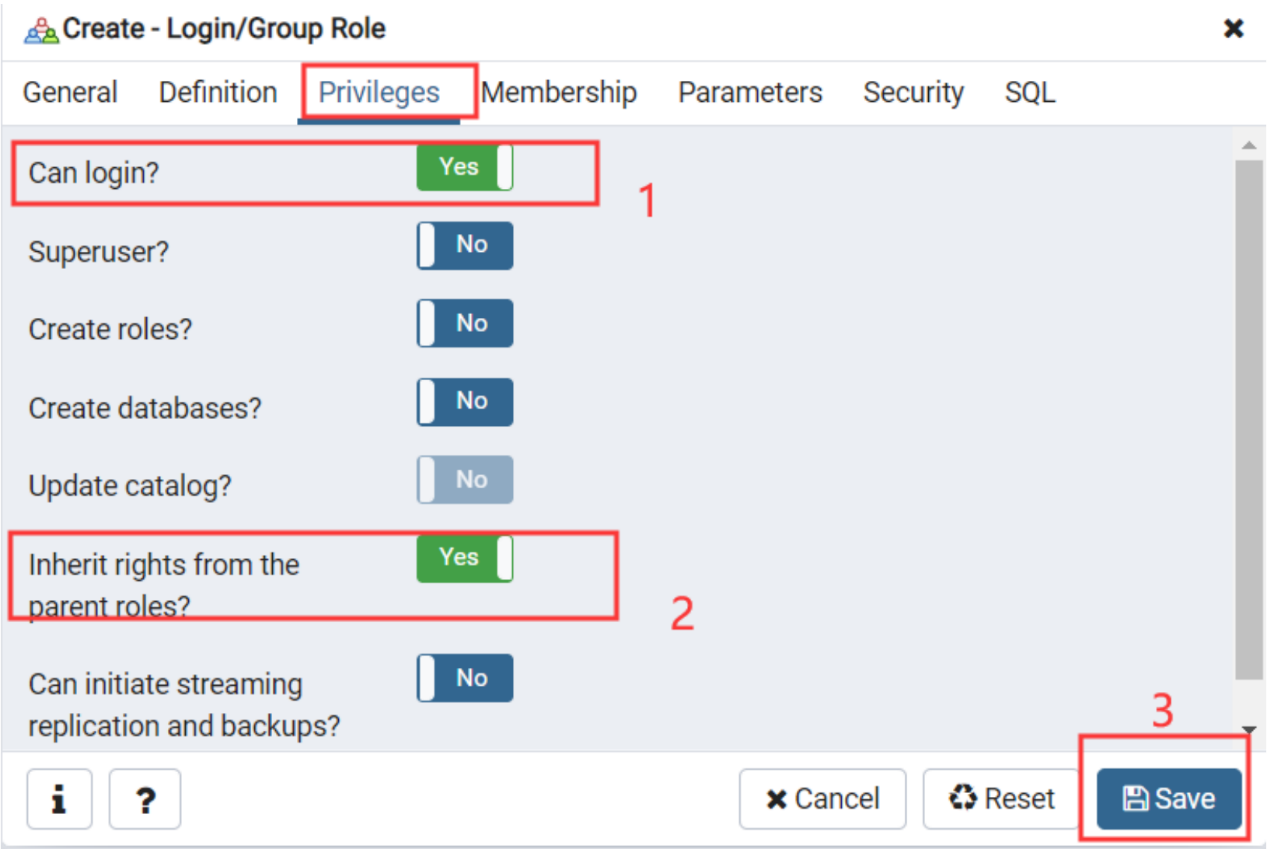
1. **创建sonar用户**

以postgres用户（密码：postgres）登录成功之后，创建为SonarQube专用的sonar用户。如下图，在General中设置用户名为sonar，Definitor中设置密码，Privileges中控制权限。用户名为sonar，密码为sonar。

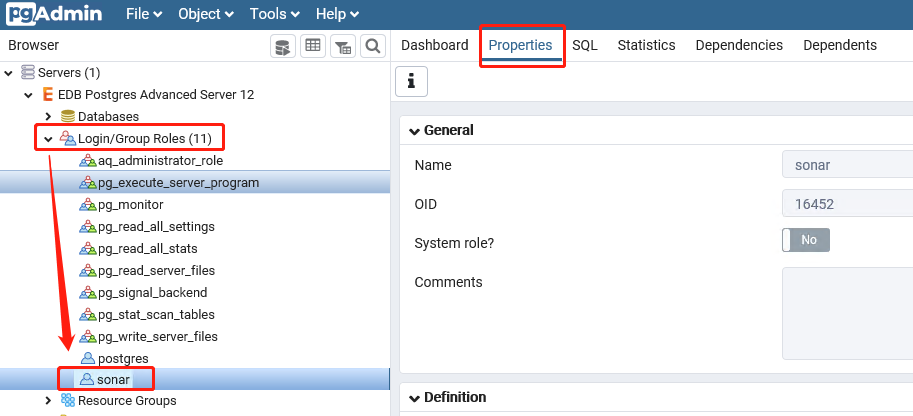






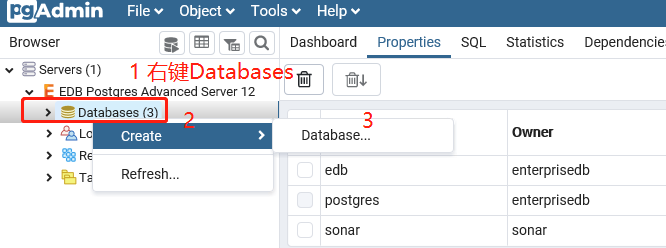


创建成功之后，在面板中可看到sonar用户。



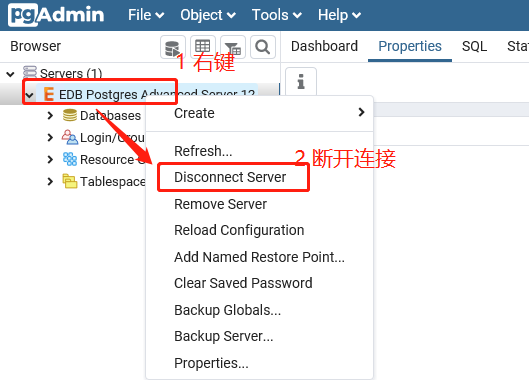
1. **创建sonar数据库**

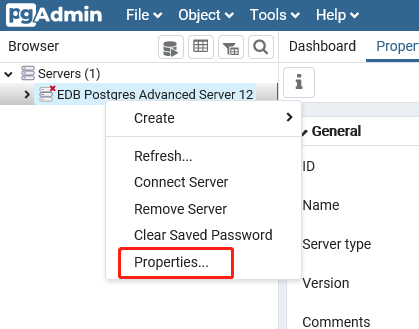
点开Databases，右键 ，创建数据库。数据库名称为：sonar

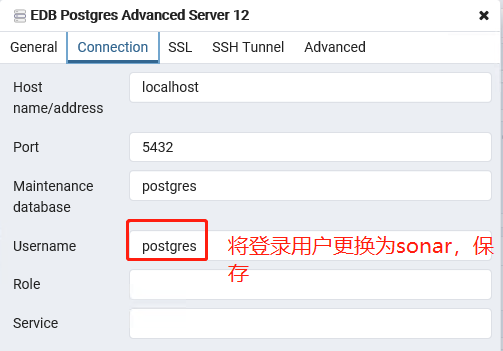


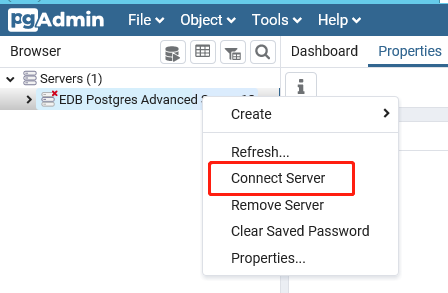


保存成功之后，可以切换为sonar用户登录。







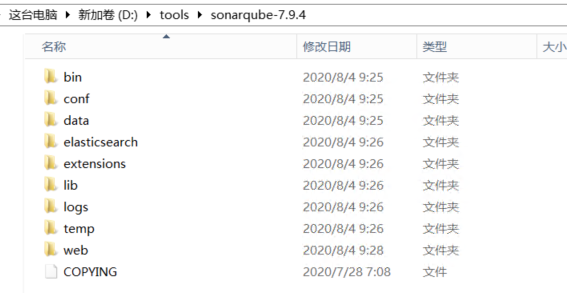




1. **SonarQube（LTS）安装**
2. **SonarQube安装**

官网下载LTS 版本的 zip 安装包到本地，解压执行即可。

解压目录结构如下：



相关目录介绍：

1. bin 用来启动 SonarQube 服务，这里已经提供好了不同系统启动 | 停止脚本了，目前提供了 linux-x86-32、linux-x86-64、macosx-universal-64、windows-x86-32、windows-x86-64
2. conf 用来存放配置文件，若需要修改配置，修改 sonar.properties 文件即可。
3. data 用来存放数据，SonarQube默认使用 h2 数据库存储，同时支持其他如Orace、Mssql、Postgresql数据库存储。
4. extensions 用来存放插件 jar 包，以后我们需要安装插件就放在这里。
5. lib 用来存放各种所依赖的 jar 包，包括上边各数据库驱动包 (默认已提供一个版本，如果版本不匹配，则在这里手动更新下)。
6. logs 用来存放各日志信息
7. web 用来提供 SonarQube web 网页服务。
8. **配置PostgreSQL连接信息**

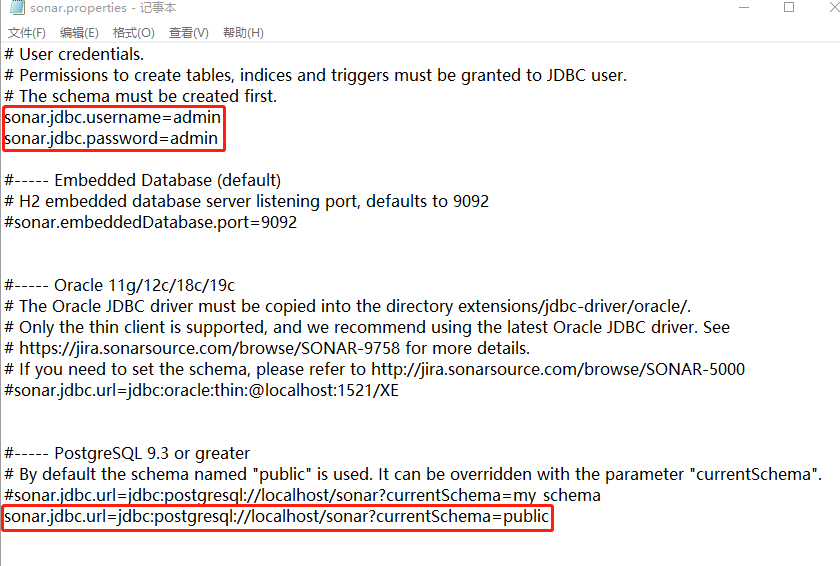
在SonarQube的安装目录的conf目录下，打开sonar.propertie。此文件当中，所有支持的数据类型的连接方式都已经放进来了。只需要开放注释，进行配置工作即可。

sonar.propertie修改如下：

sonar.jdbc.url=jdbc:postgresql://localhost/sonar?currentSchema=public

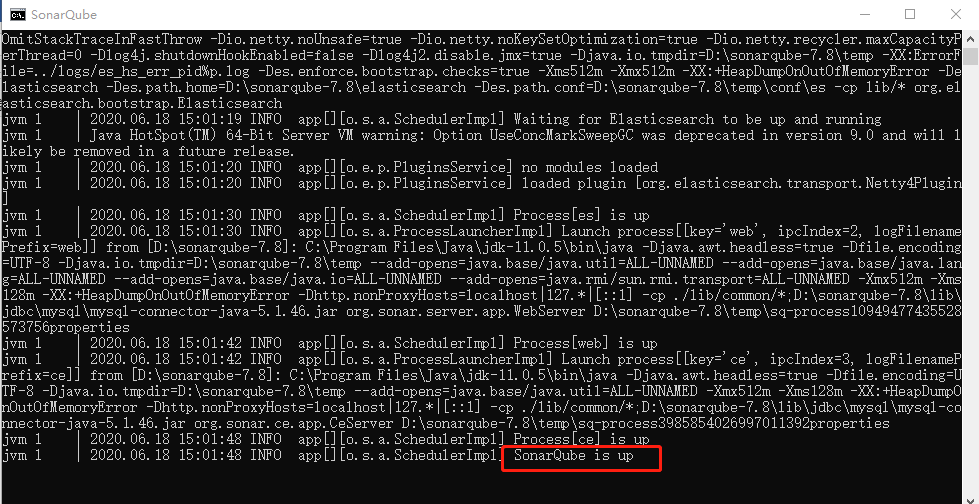
sonar.jdbc.username=admin

sonar.jdbc.password=admin



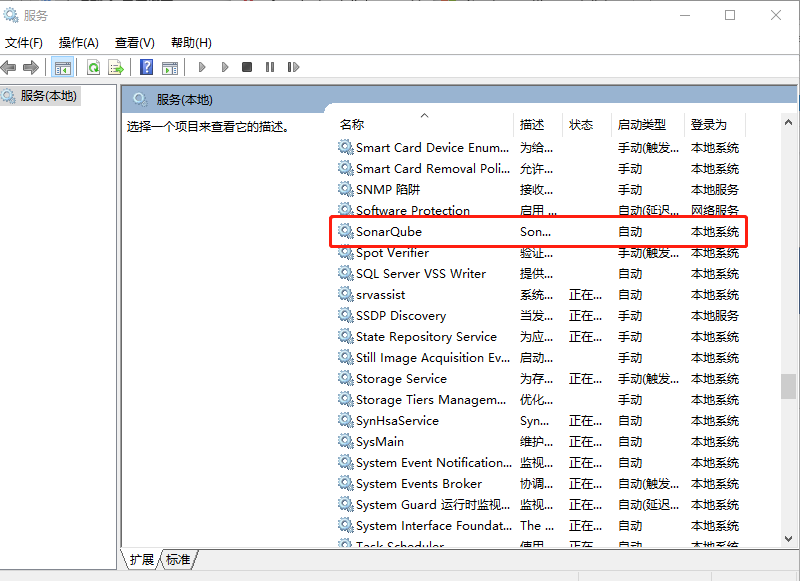
1. **启动**

以管理员身份运行安装目录bin\ StartSonar.bat, 启动成功的界面：

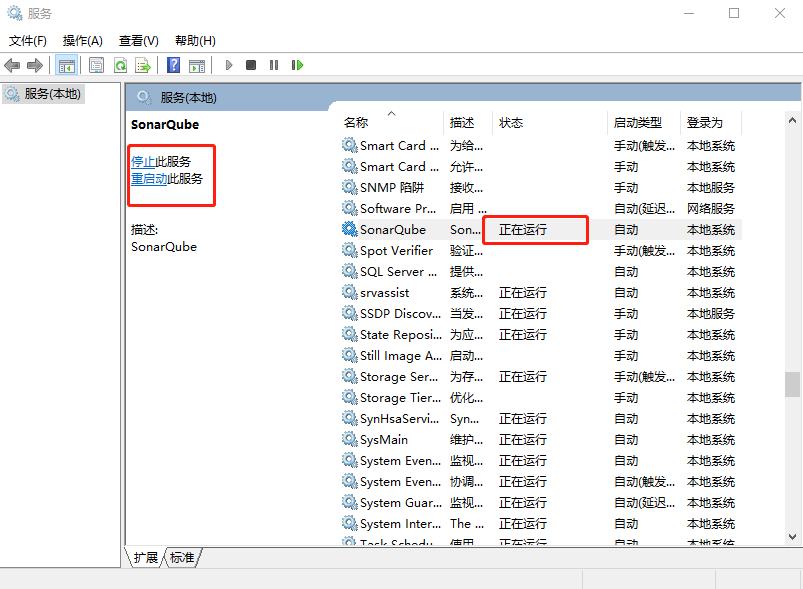


将SonarQube服务器添加到本地服务：

以管理员身份运行安装目录bin\InstallNTService.bat，在Windows本地安装SonarQube服务器。



此时服务未启动，需手动启动。

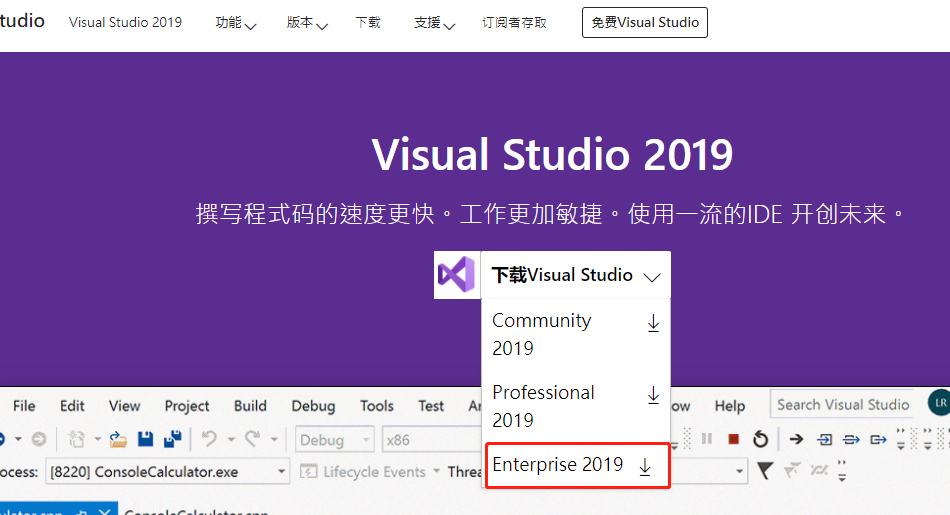


打开浏览器：<http://localhost:9000/>

默认的登陆账号和密码： admin/admin

1. **MSBuild**
2. **安装Visual Studio编译器**

官网下载Visual Studio Enterprise 2019安装程序，运行vs\_Enterprise.exe选择配置进行安装。

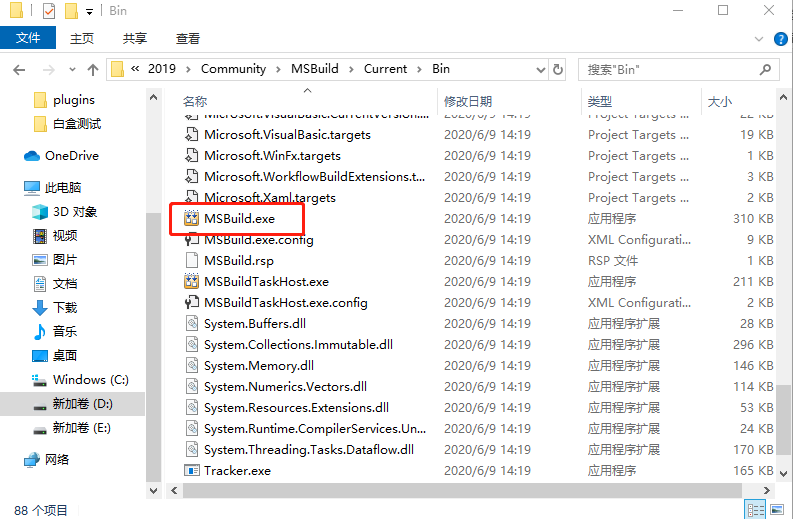


1. **配置MSBuild**

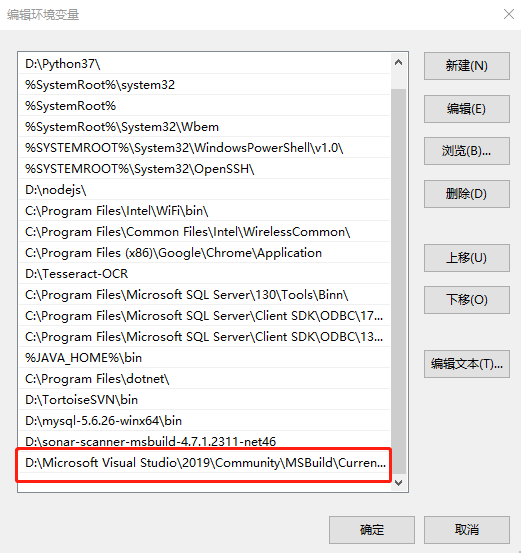
Visual Studio IDE在编译\*.sln解决方案时默认是调用msbuild.exe来实现的。如果你的机器上没有装有Visual Studio，那么也可以单独使用MSBuild来编译.sln（工程解决方案）或.csproj（项目）。MSBuild可以直接通过.NETFramework来安装获得。

msbuild.exe的路径在Visual Studio安装路径下，例如：D:\Microsoft Visual Studio\2019\Enterprise\MSBuild\Current\Bin

msbuild.exe目录如下所示：



将MSBuild.exe所在路径添加到系统变量的Path下，便于后面在命令行中调用MSBuild。



1. **安装第三方分析器**

SonarQube 插件安装非常简单。SonarQube已经默认给我们安装了一些很常用的插件，我们可以通过admin登录，点击 Administration -> Marketplace 查看下载。如果此处没有相匹配的插件，也可在SonarQube Marketplace：<https://www.sonarplugins.com/> 进行插件的下载。

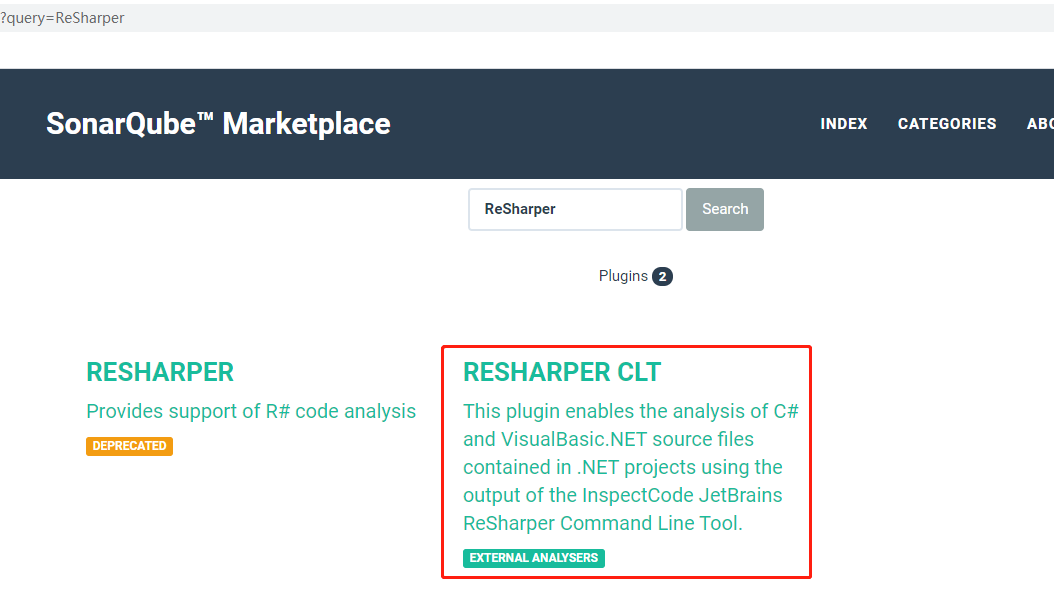
1. **ReSharper**

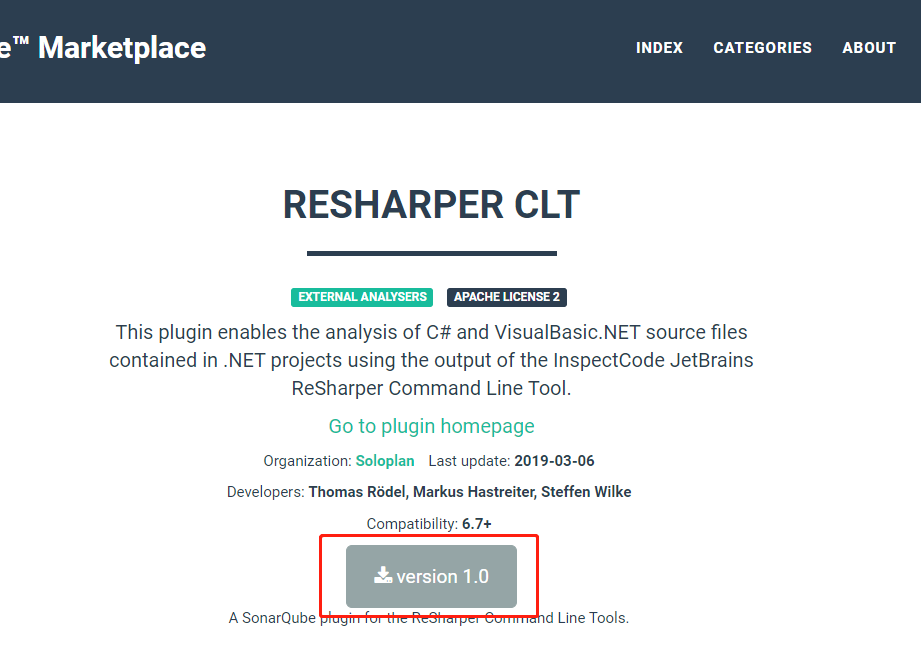
ReSharper Command Line Tools是一组免费的独立工具，这些工具可以在不安装ReSharper的情况下使用，也可以与Continuous Integration或代码质量管理服务器集成。

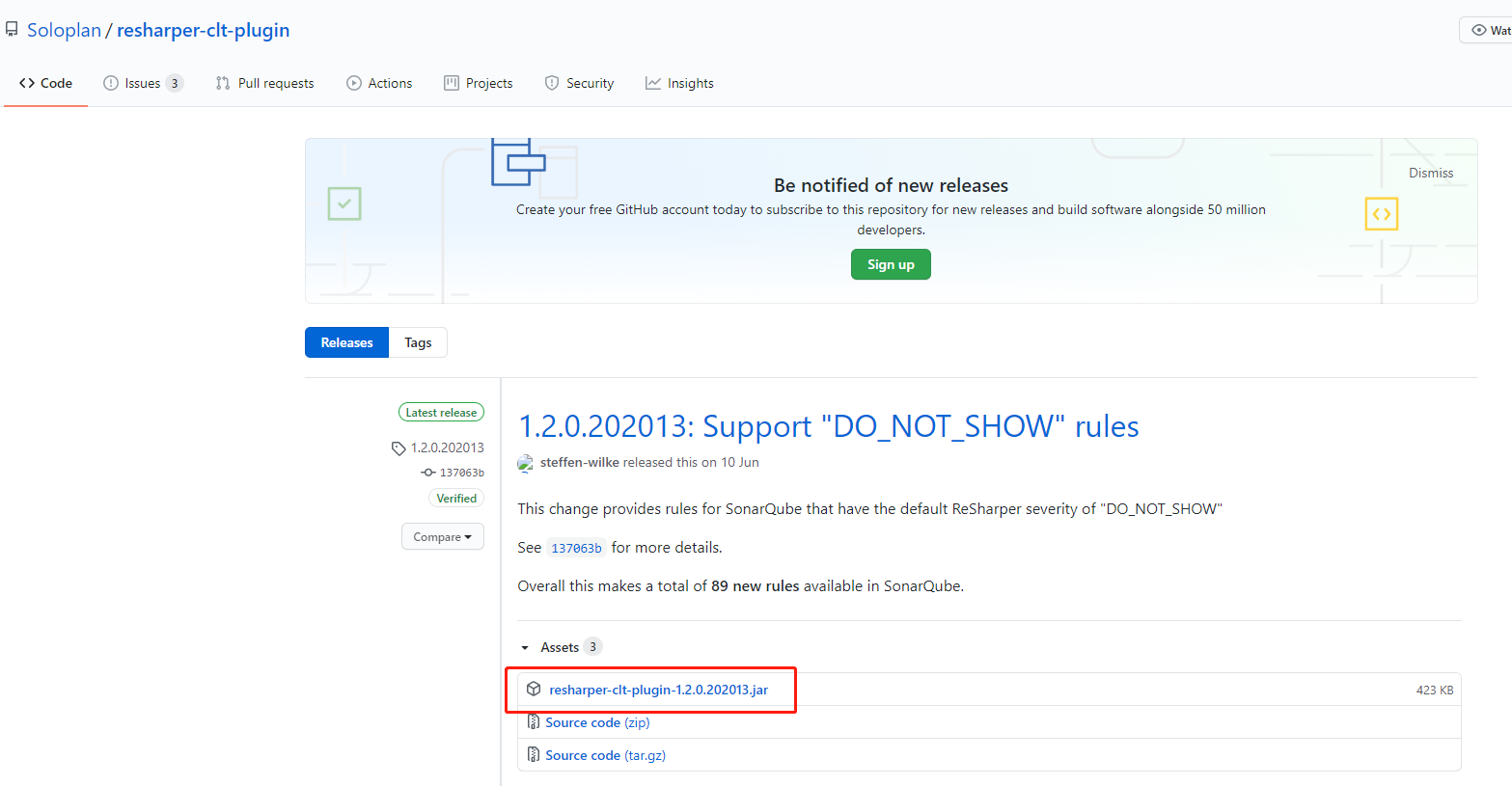
参考地址：https://github.com/Soloplan/resharper-clt-plugin

1. **安装SonarQube ReSharper插件**

在 SonarQube Marketplace中搜索ReSharper插件，选择CLT版本的插件进行下载，此版本的插件可使用ReSharper Command Line Tools的输出工具InspectCode来分析.NET项目中包含的C＃和VisualBasic.NET源文件。



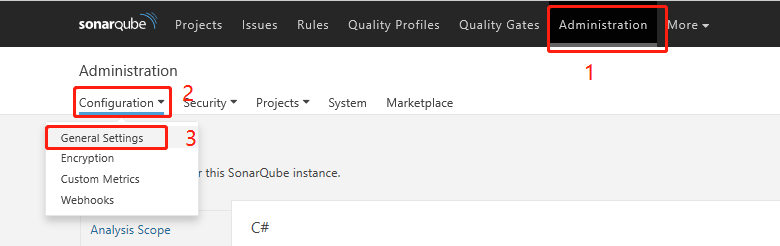




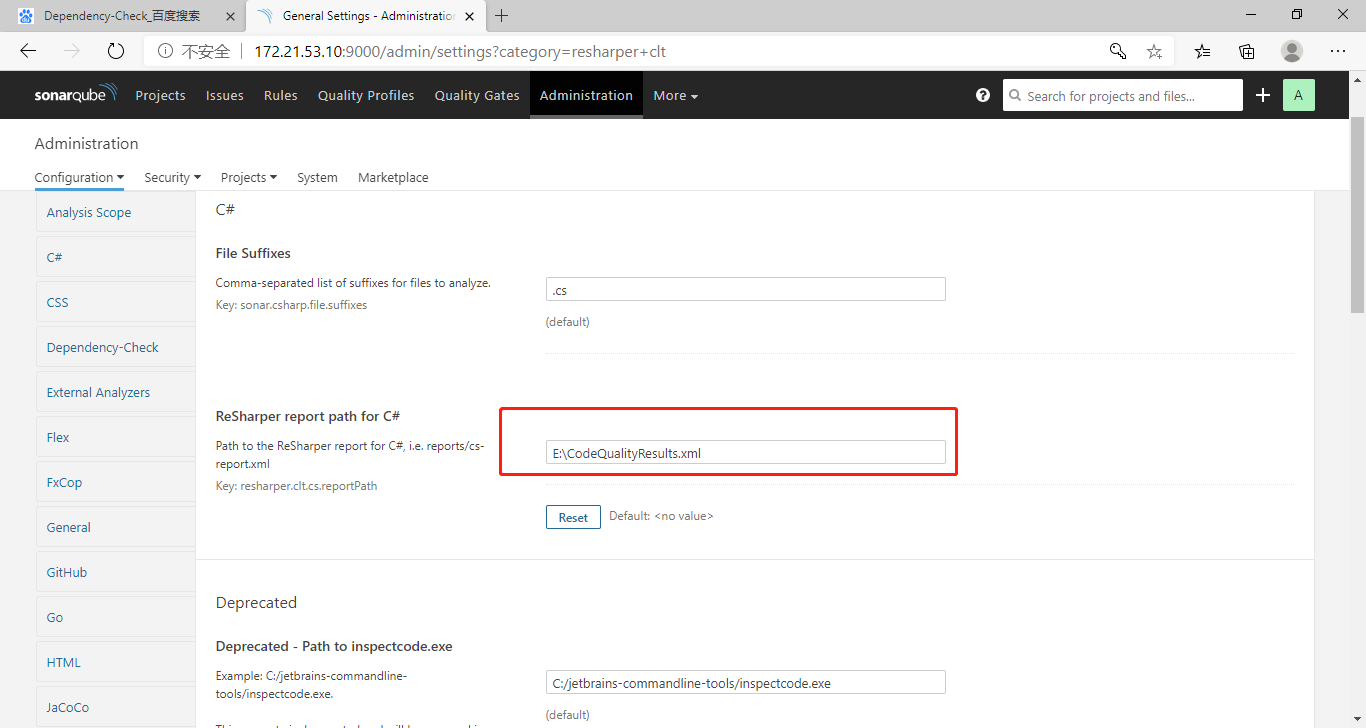
将下载的插件.jar文件复制到SonarQube插件目录extensions/plugins下，重启SonarQube。

1. **配置****ReSharper插件**

在SonarQube导航栏中选择【Administration】，在【Configuration】下拉列表中选择【General Settings】，在左侧栏中选择【ReSharper CLT】。



【ReSharper report path for C#】中输入ReSharper报告的存放路径，例如：E:\CodeQualityResults.xml。

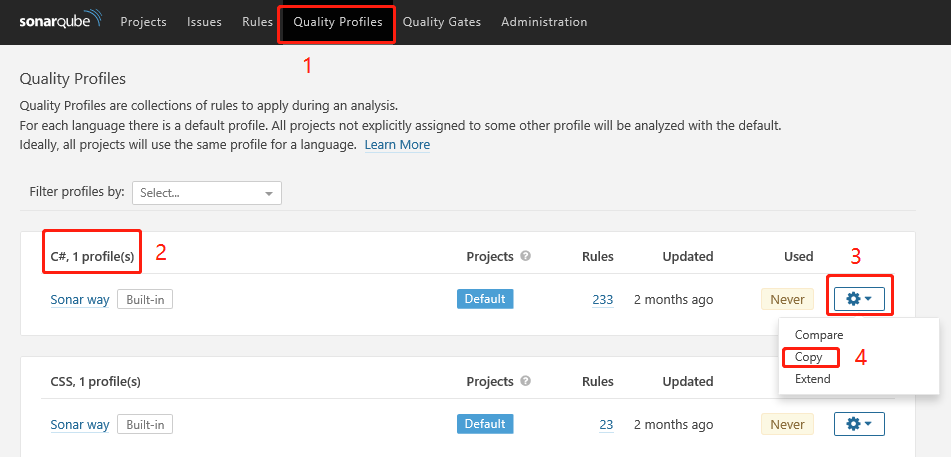


1. **将ReSharper规则添加到质量配置文件**

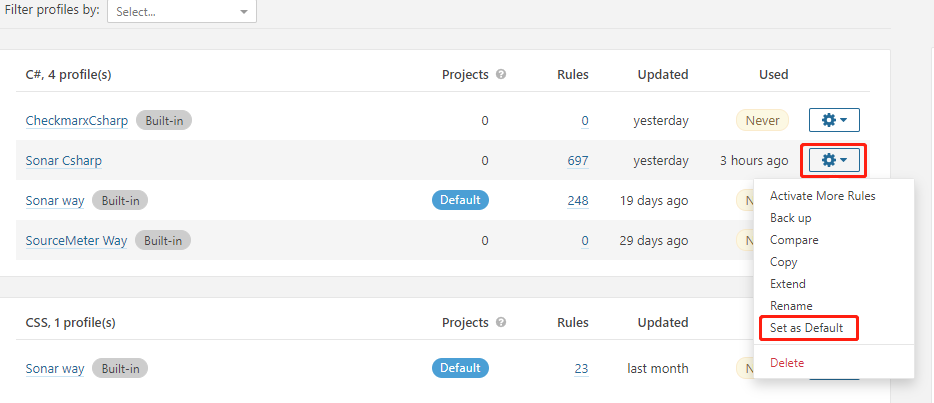
SonarQube中的质量配置文件实质上是用于将一组规则进行分组的机制。每一个SonarQube项目的每种分析语言都有一个关联的质量配置文件，确定了该语言针对该项目运行的规则。

安装ReSharper插件并重新启动服务后，SonarQube将读取ReSharper规则，这些规则并没有在项目关联的质量文件中被激活，需要自行激活。默认的Sonar Way配置文件，无法新增规则。通过复制默认的【Sonar Way】配置文件来创建命名为【Sonar Csharp】的配置文件，将ReSharper规则添加到该文件中。

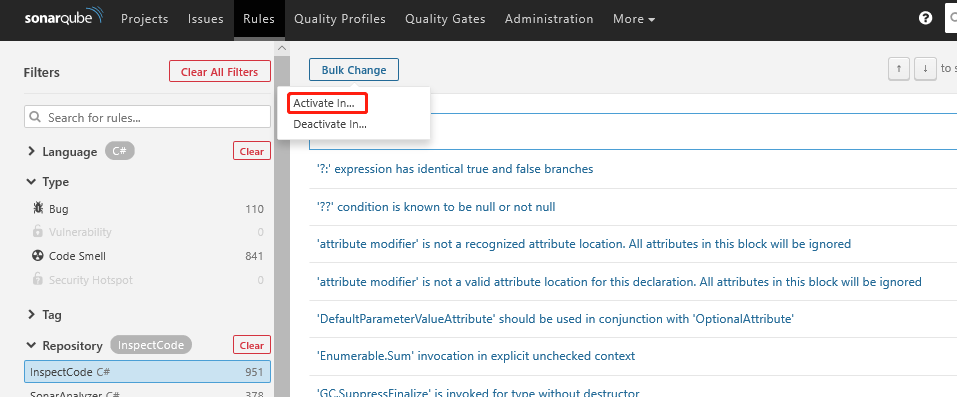
* + 1. 以管理员身份登录SonarQube服务器（您必须是管理员才能更新质量配置文件）。
    2. 选择质量配置文件，然后选择默认的Sonar Way配置文件。
    3. 选择复制，然后创建一个名为【Sonar Csharp】的新质量配置文件。



* + 1. 在【Sonar Csharp】条目中选择【Set as Default】将其设置为C#项目的默认值。



* + 1. 在顶部导航工具栏中选择【Rules】，然后在左侧栏中选择【Repository】，选择【InspectCode C#】查看ReSharper规则列表。
    2. 选择【Bulk Change】，然后选择激活输入。从下拉列表中，选择【Sonar Csharp】配置文件。



1. **FxCop**

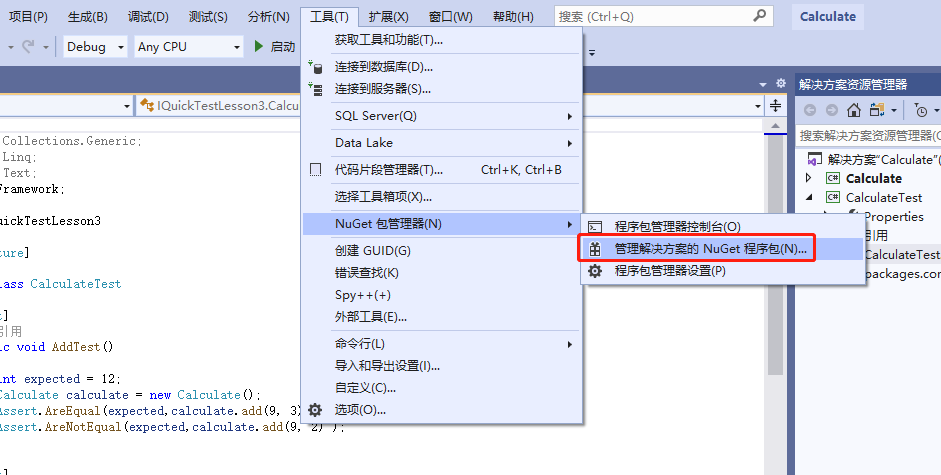
FxCop是一个代码分析工具，它依照微软.NET框架的设计规范对托管代码assembly(可称为程序集，assembly实际上指的就是.net中的.exe或者.dll文件（不包括netmodule文件）。

SonarQube FxCop插件，可以使用FxCop工具扫描C＃和VB.net项目，需在Sonar Qube服务器质量配置文件中激活FxCop的规则。

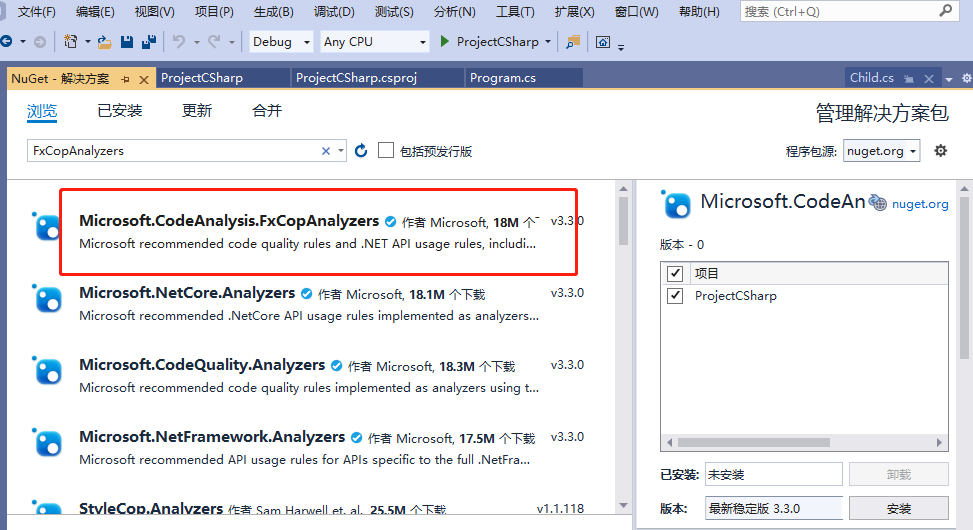
参考地址：<https://github.com/DanielHWe/sonar-fxcop>

1. **Visual Studio中安装FxCop分析器**

打开VS，在导航栏的【工具】选项中打开NuGet 包管理器（联网状态）。

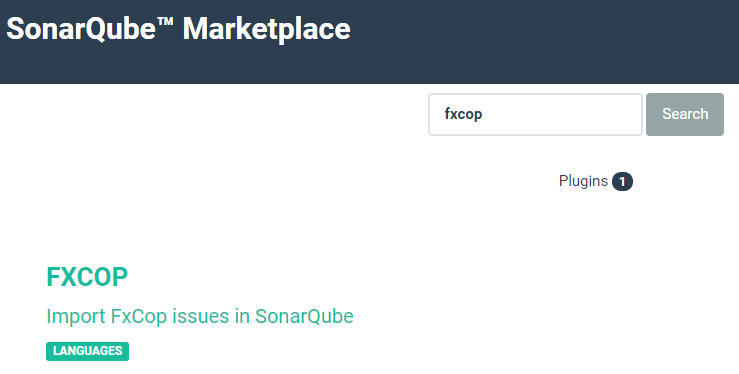


在浏览页面的文本框中输入FxCopAnalyzers，选择Microsoft.CodeAnalysis.FxCopAnalyzers进行安装。



1. **安装SonarQube FxCop插件**

在 SonarQube Marketplace中搜索FxCop插件, 将下载的插件.jar文件复制到SonarQube插件目录extensions/plugins下，重启SonarQube。





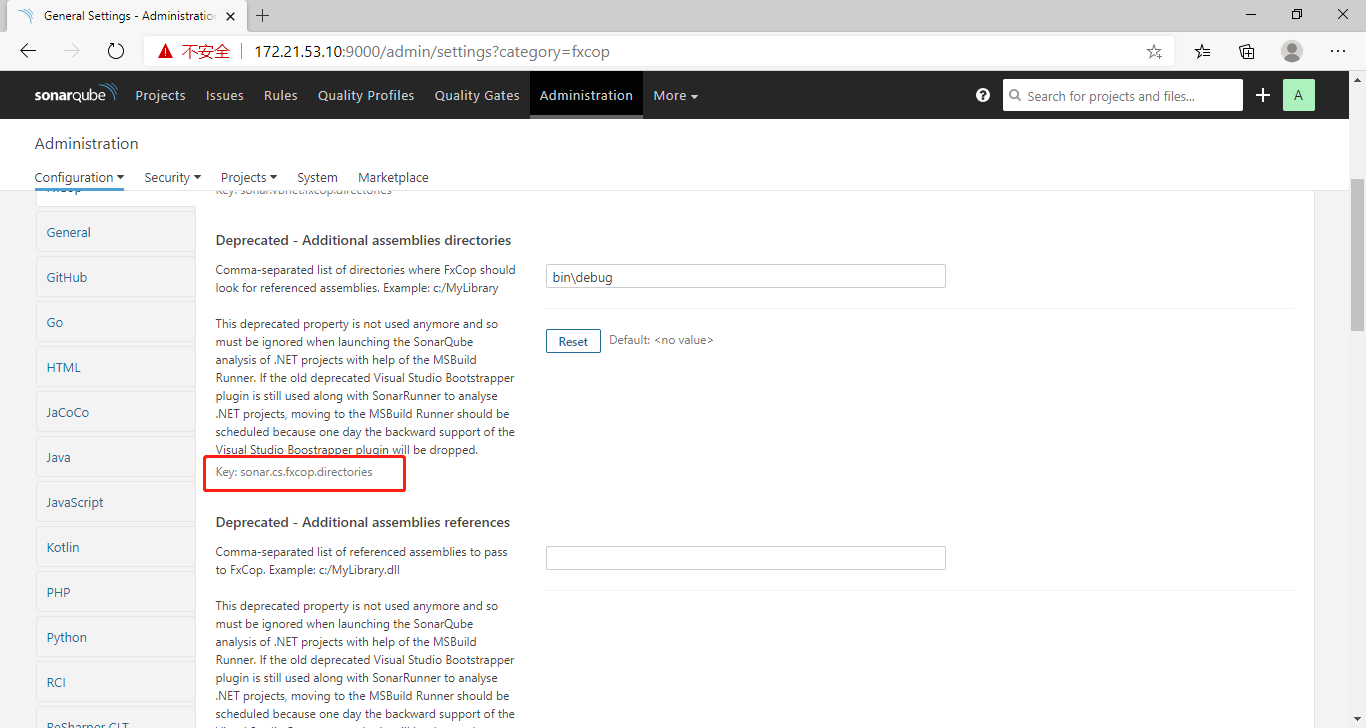
1. **配置FxCop插件**

在SonarQube导航栏中选择【Administration】，在【Configuration】下拉列表中选择【General Settings】，在左侧栏中选择【FxCop】。

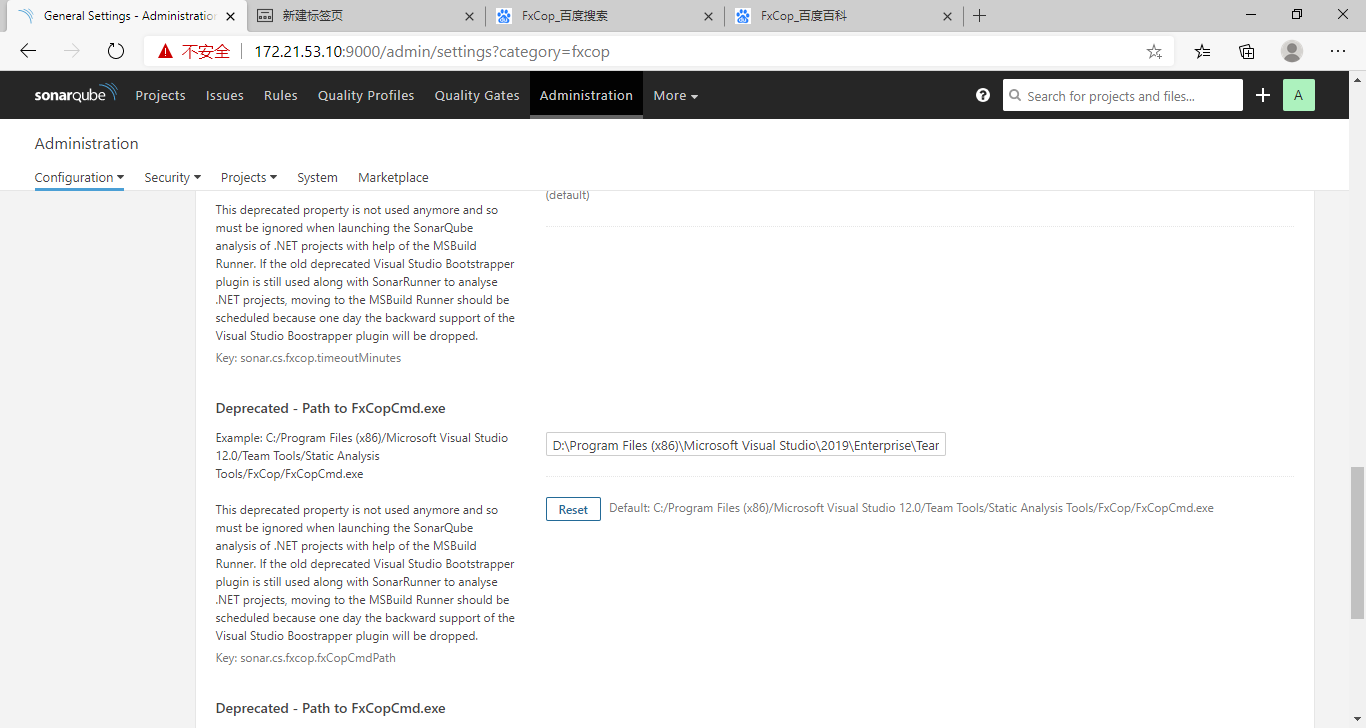
因为FxCop工具扫描C＃和VB.net项目，而我们的是C＃项目，所以只需配置关于C#的属性即可。

【注】：C＃和VB.net的属性以每一项中的Key值区分，VB.net属性：Key: sonar.vbnet.fxcop.xxxx；C＃属性：Key: sonar.cs.fxcop.xxxx

在【Deprecated - Additional assemblies directories】中输入“bin\debug”。



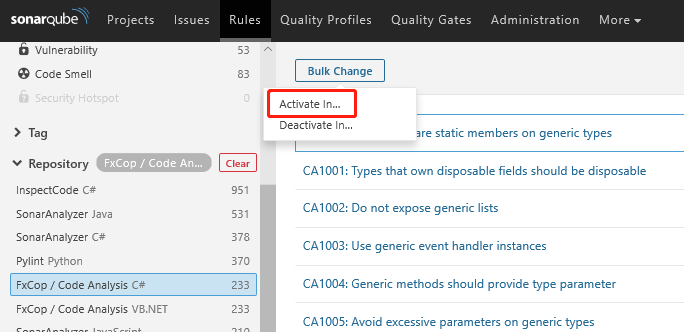
【Deprecated - Path to FxCopCmd.exe】中输入本地FxCopCmd.exe的路径+ FxCopCmd.exe，FxCopCmd.exe一般在vs2019安装路径下的2019\Enterprise\Team Tools\Static Analysis Tools\FxCop文件中。



1. **将FxCop规则添加到质量配置文件**

在顶部导航工具栏中选择【Rules】，然后在左侧栏中选择【Repository】，选择【FxCop/Code Analysis C#】查看FxCop规则列表。

选择【Bulk Change】，然后选择激活输入。从下拉列表中，选择【Sonar Csharp】配置文件。



1. **Dependency-Check**

Dependency-Check是一款根据项目依赖来检查漏洞的白盒审计工具。它根据项目中的依赖库，搜集依赖的版本，厂商等信息，然后匹配NVD漏洞库，对比其中的CPE来确定此版本的依赖包是否存在漏洞。

Dependency-Check支持以多种不同语言（包括Java，.NET，Node.js，Ruby和Python）识别项目依赖项。无论是jar包，还是目录，或者压缩文件，它都能自动识别。

参考地址：https://github.com/dependency-check/dependency-check-sonar-plugin

1. **安装SonarQube Dependency-Check插件**

在 SonarQube Marketplace中搜索Dependency-Check插件。将下载的Dependency-Check插件.jar文件复制到SonarQube插件目录extensions/plugins下。重启SonarQube。

1. **配置Dependency-Check插件**

在SonarQube导航栏中选择【Administration】，在【Configuration】下拉列表中选择【General Settings】，在左侧栏中选择【Dependency-Check】。

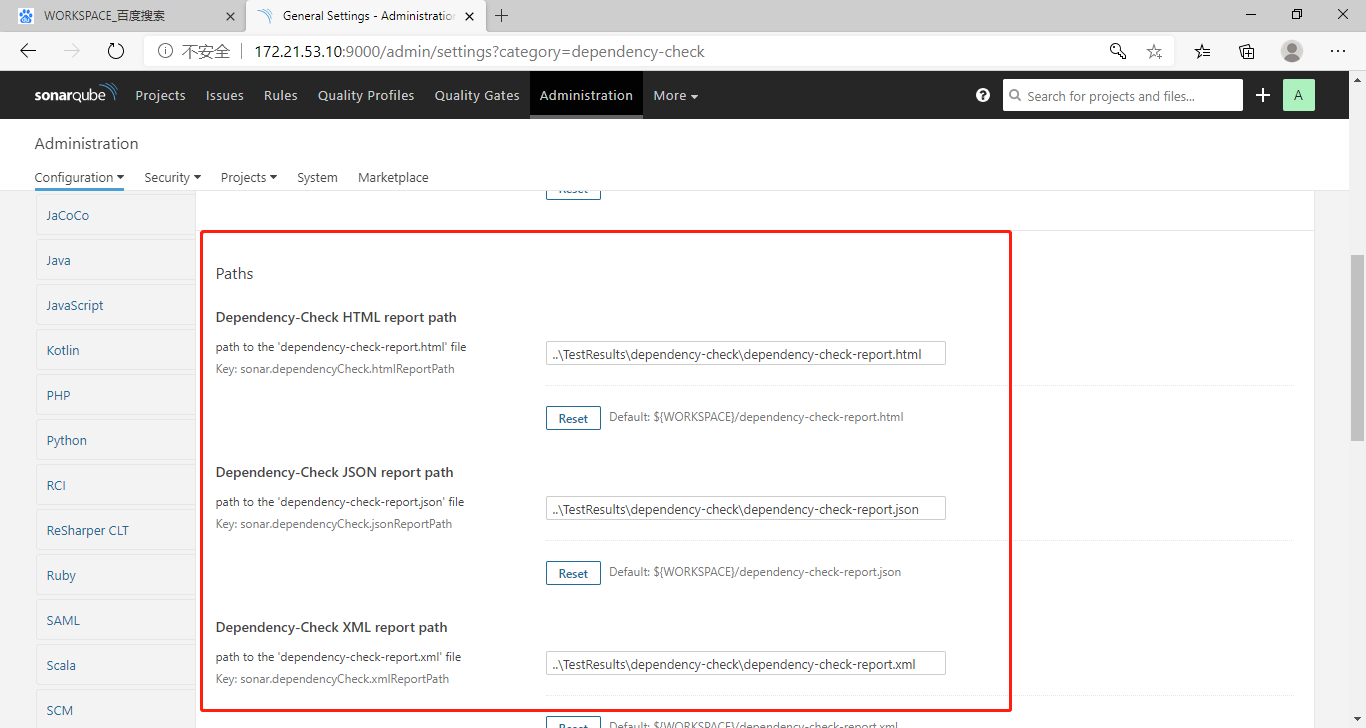
在Paths中分别输入Dependency-Check以HTML、JSON、XML三种格式生成报告的存放路径。

HTML：..\TestResults\dependency-check\dependency-check-report.html

JSON：..\TestResults\dependency-check\dependency-check-report.json

XML：..\TestResults\dependency-check\dependency-check-report.xml

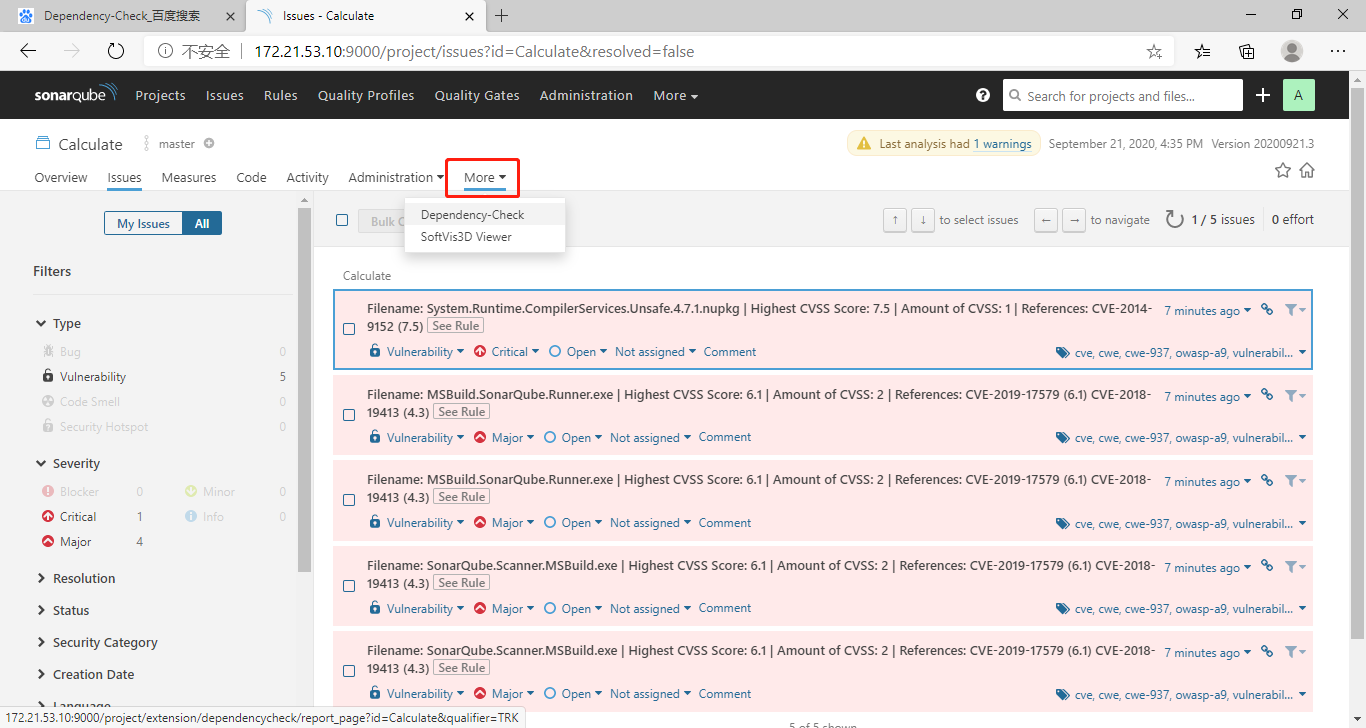
注：${WORKSPACE}/dependency-check-report.html中的WORKSPACE为Pipelines的WORKSPACE。

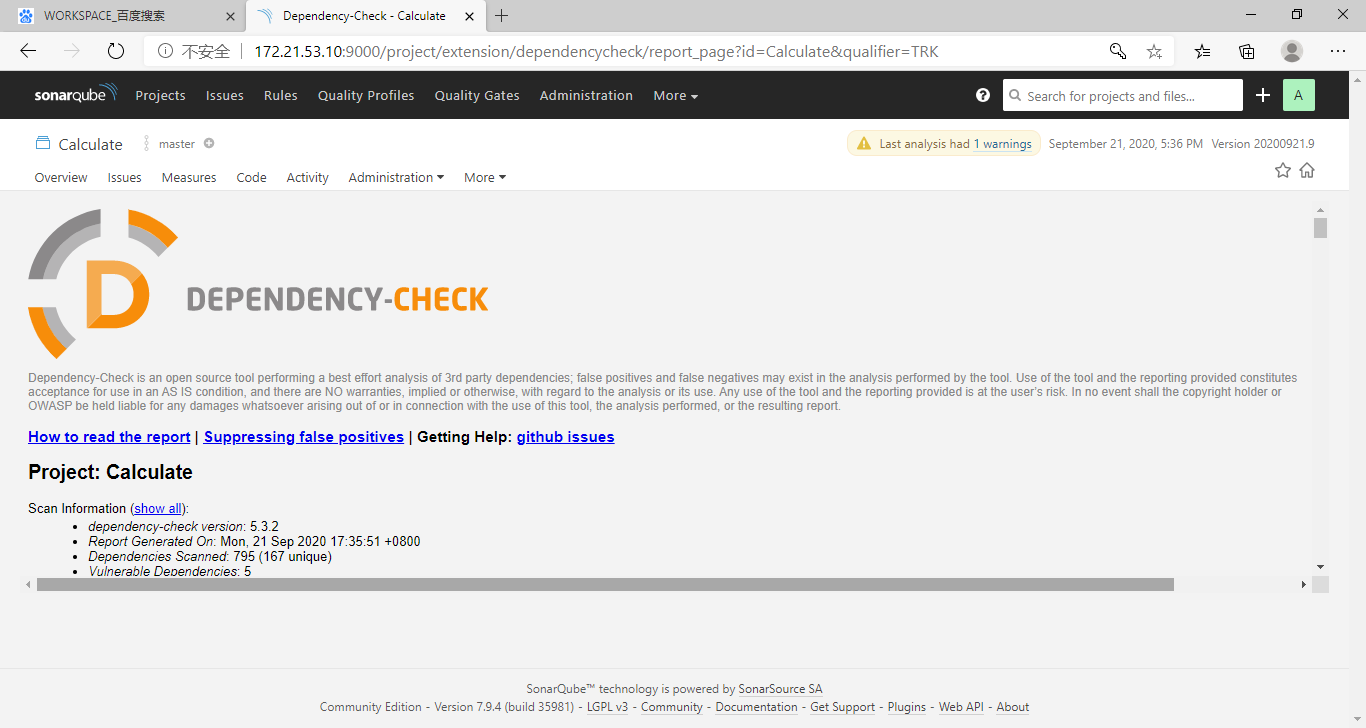


1. **查看Dependency-Check报告**

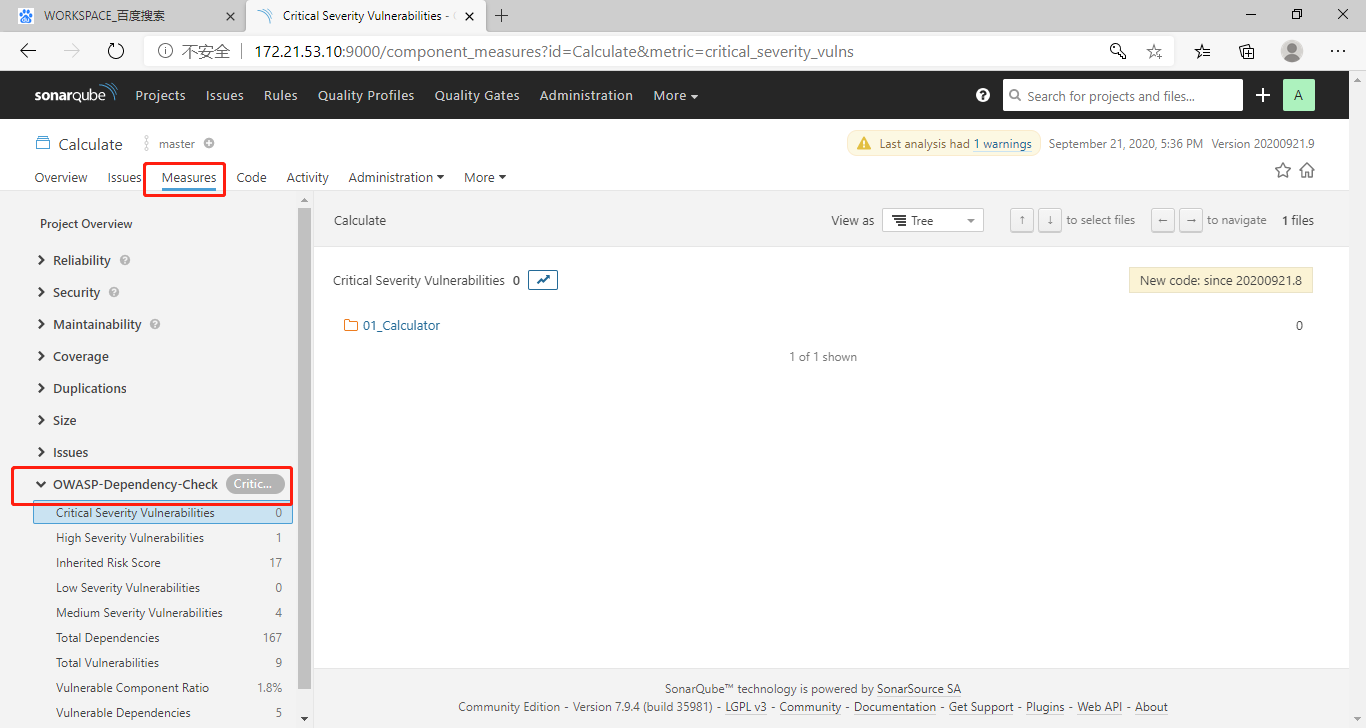
查看Dependency-Check报告有两种方式：

第一种方式，可在【Projects】中，打开检测项目，在项目导航栏中点击【More】选项，点击【Dependency-Check】，即可打开Dependency-Check报告。





第二种方式，可在【Projects】中，打开检测项目，在项目导航栏中点击【Measures】选项，在左侧栏中选择【OWASP-Dependency-Check】查看相关信息。



1. **Rules Compliance Index(RCI)**

此插件的作用是计算SonarQube项目的规则遵从指数(RCI)，是衡量代码规则合格性的标准。

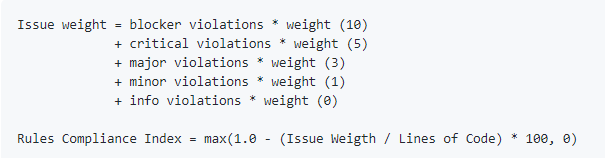
参考地址：<https://github.com/willemsrb/sonar-rci-plugin>

1. **安装****SonarQube RCI插件**

在 SonarQube Marketplace中搜索RCI插件，将下载的SonarQube RCI插件.jar文件复制到SonarQube插件目录extensions/plugins下，重启SonarQube。

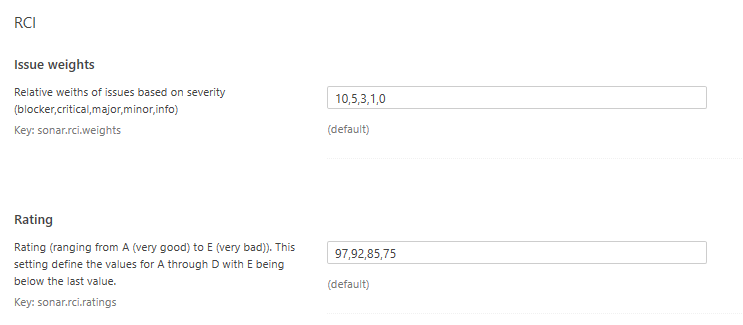
1. **计算RCI指数**

根据问题的加权值和代码行数计算一个度量，公式如下：



1. **配置度量标准**

该插件可自行配置规则合规等级度量标准，在SonarQube导航栏中选择【Administration】，在【Configuration】下拉列表中选择【General Settings】，在左侧栏中选择【RCI】。



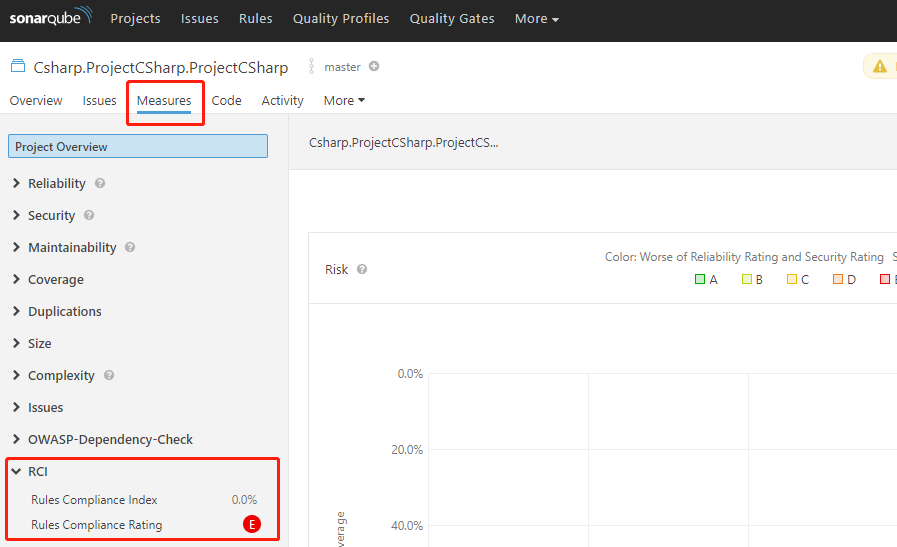
Issue weights：基于问题的严重性，严重等级分为blocker、critical、major、minor、info，文本框的数字是权重，顺序一一对应，可做修改。

Rating：评分，从A（非常好）到E（非常差）。文本框中设置定义A到D的值，其中E的值默认小于D的值。

当RCI指数大于97，则评分为A。

1. **查看代码的RCI**

可在【Projects】中，打开检测项目，在【Measures】的左侧栏中选择【RC I】进行查看。



1. **SoftVis3D**

软件系统是复杂的无形结构，很难理解。因此，在不同视图中可视化软件属性，结构和依赖关系在软件系统的生命周期中起着重要作用。SoftVis3D是用于3D可视化软件结构和指标的工具，可以使用SonarQube将其集成到软件开发过程中。

从项目的源头开始构建软件，使其具有形状。从开发人员到管理人员，都可以通过可视化来交流有关结构的信息。

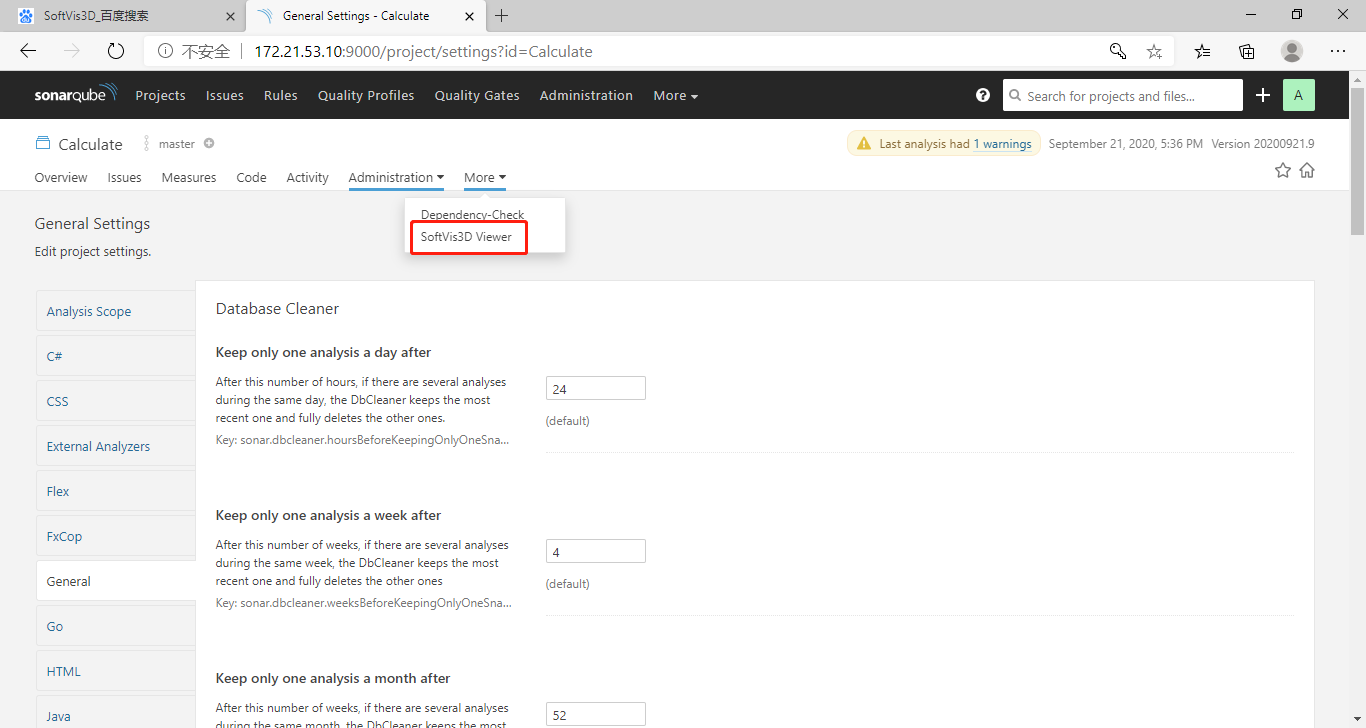
参考地址：<https://softvis3d.com/>

1. **安装SonarQube SoftVis3D插件**

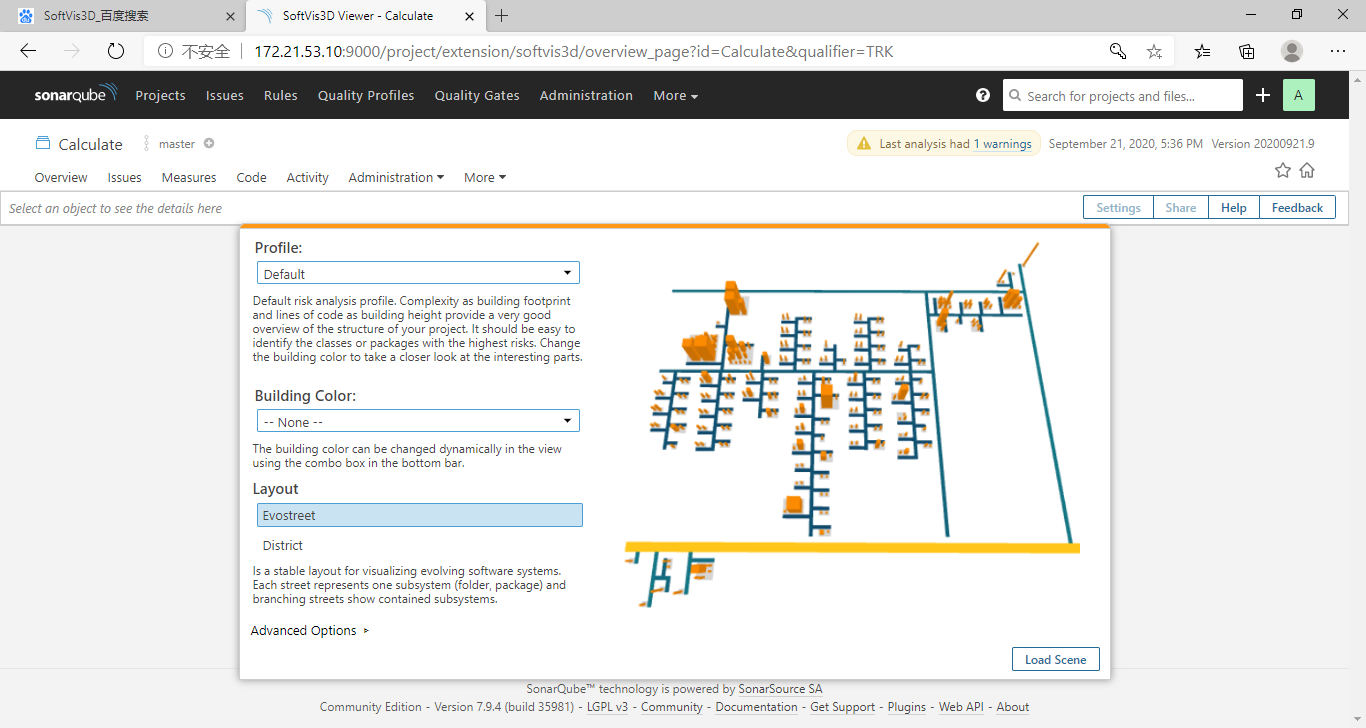
在 SonarQube Marketplace中搜索SoftVis3D插件，将下载的SonarQube SoftVis3D插件.jar文件复制到SonarQube插件目录extensions/plugins下，重启SonarQube。

1. **查看****SoftVis3D视图**

可在【Projects】中，打开检测项目，在项目导航栏中点击【More】选项，点击【SoftVis3D Viewer】，即可查看SoftVis3D视图。



在该界面，可以设置相关配置信息，使用SonarSource中的任何指标，轻松浏览3D可视化，帮助您轻松分析分析软件质量。

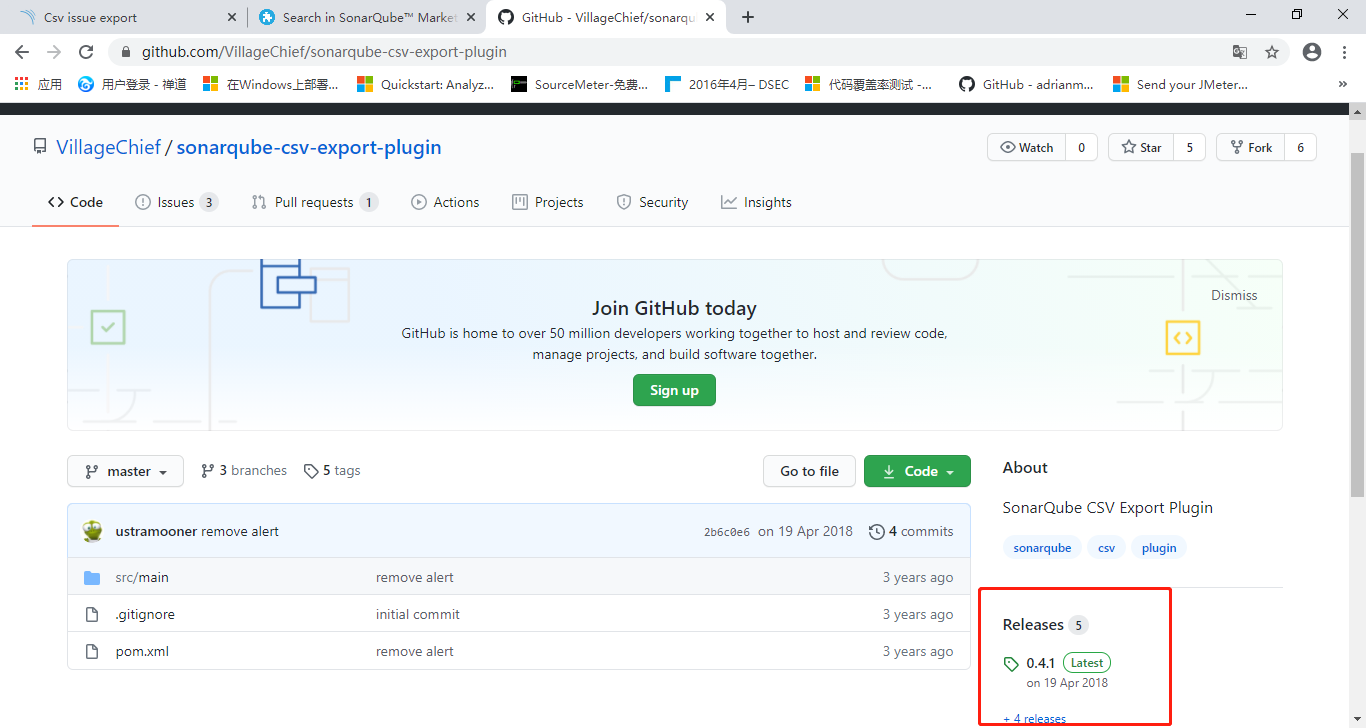


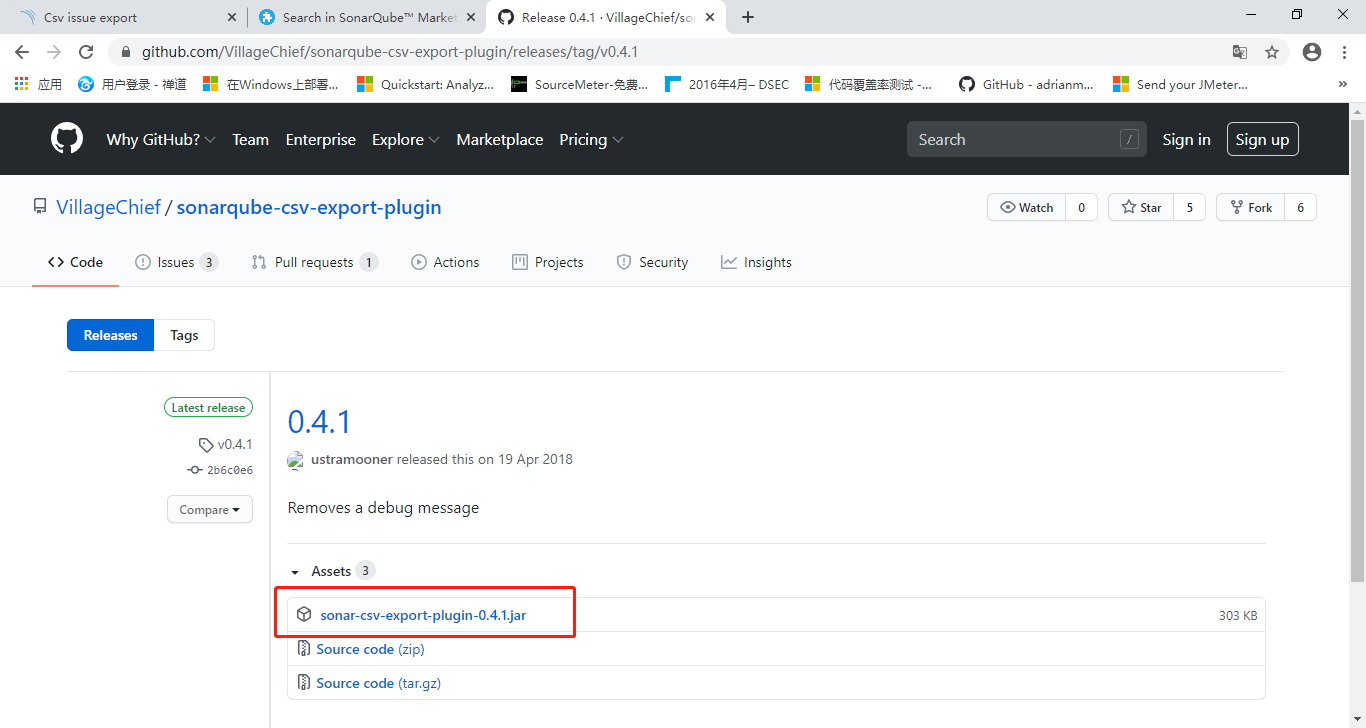
1. **Csv issue report**

该插件可以将每个项目所存在的违规问题导出到一个CSV文件。

1. **安装SonarQube CSV插件**

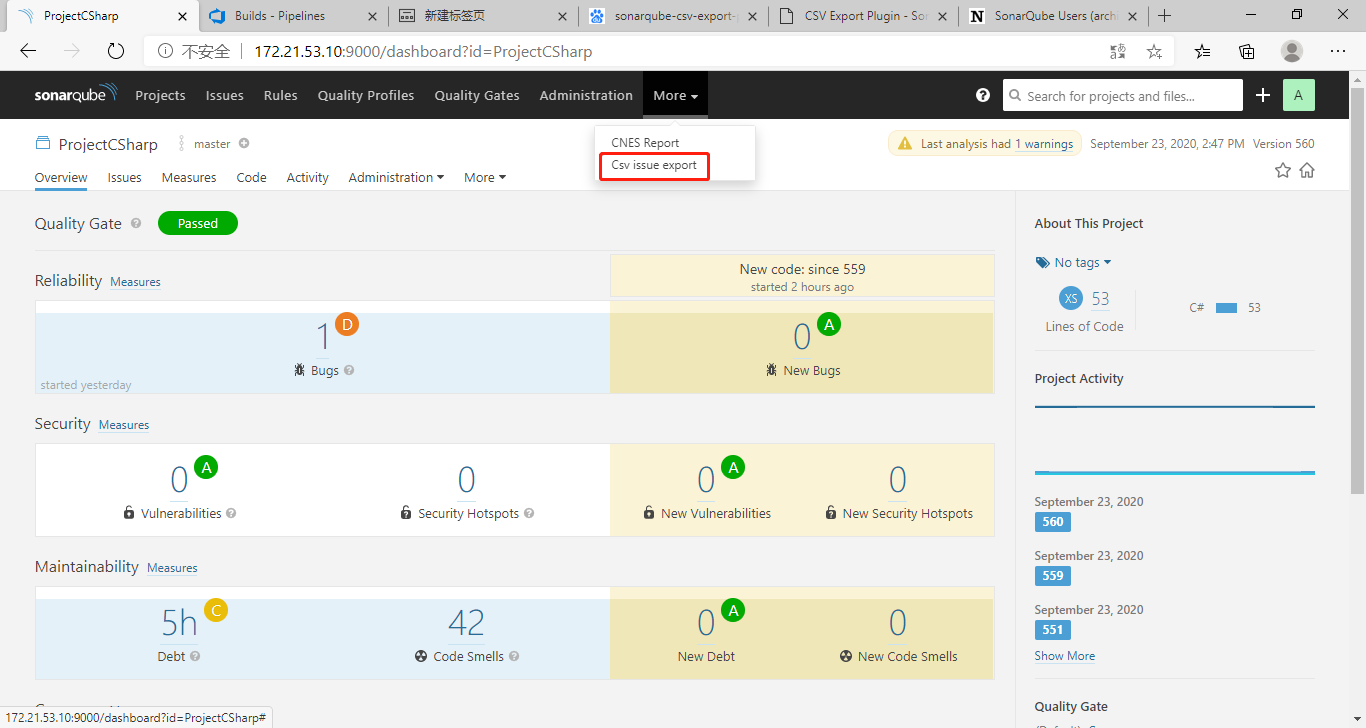
SonarQube Marketplace中没有此插件，需到github 中下载，下载地址：<https://github.com/VillageChief/sonarqube-csv-export-plugin>。将下载的插件.jar文件复制到SonarQube插件目录extensions/plugins下，重启SonarQube。



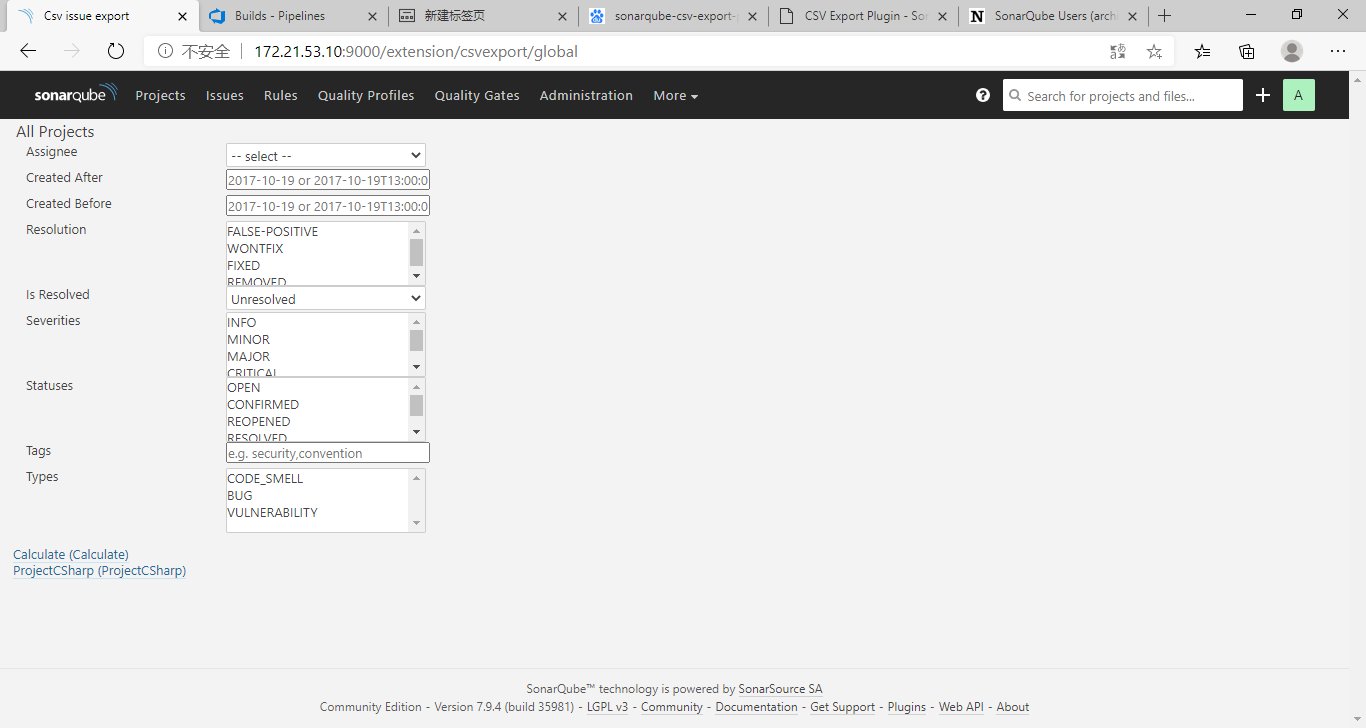


1. **导出Csv issue报告**

在SonarQube导航栏中点击【More】选项，点击【Csv issue Report】。



选择导出的信息，在选择项下会列出当前SonarQube服务器中分析的项目，选择要导出的项目，即可导出一个CSV报告。

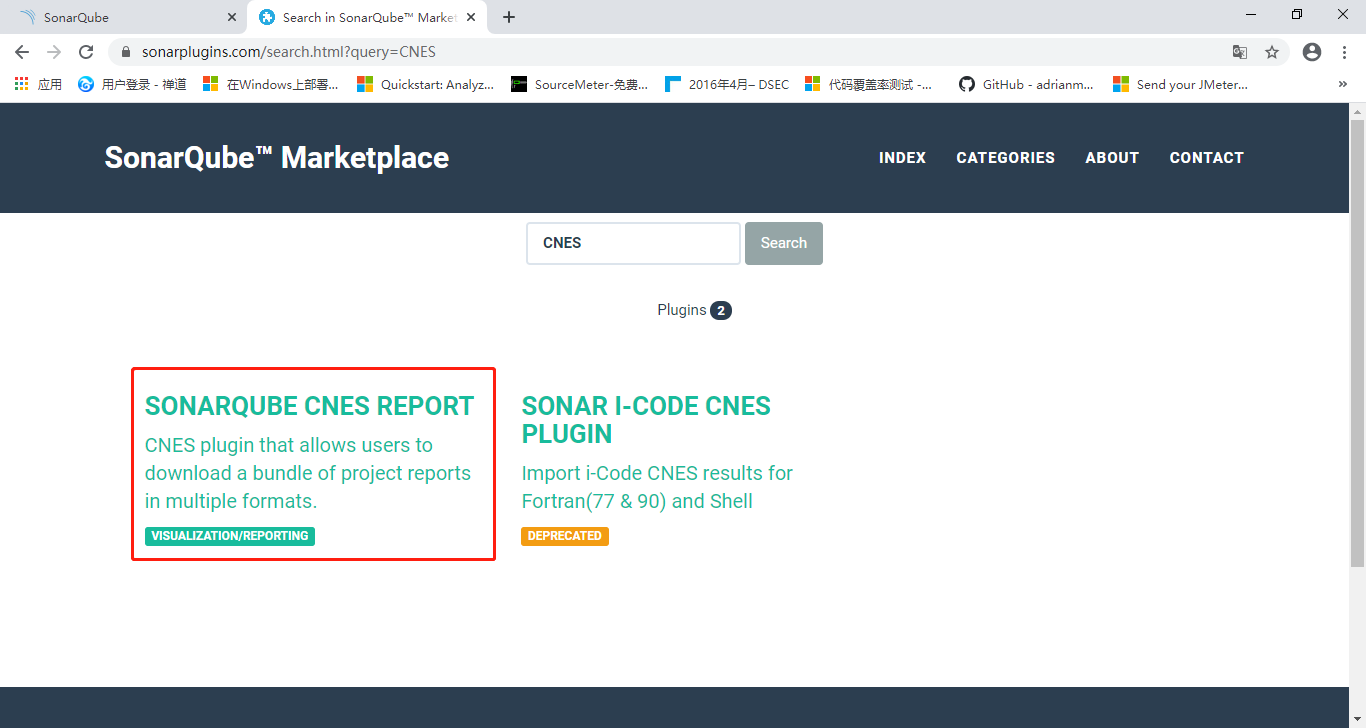


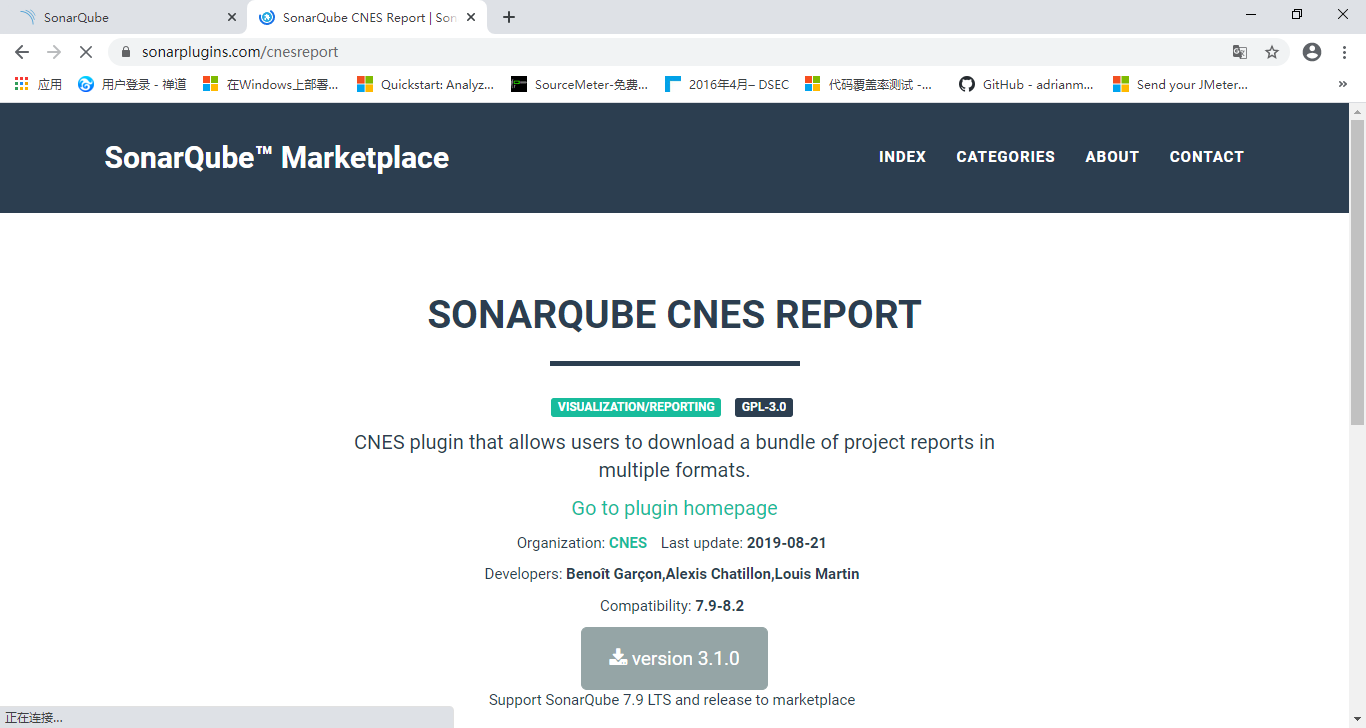
1. **CNES Report**

用于SonarQube的CNES应用程序/插件，该插件可以将SonarQube服务器中的分析报告导出为docx，xl​​sx，csv，markdown和文本文件等格式。

1. **安装SonarQube** CNES**插件**

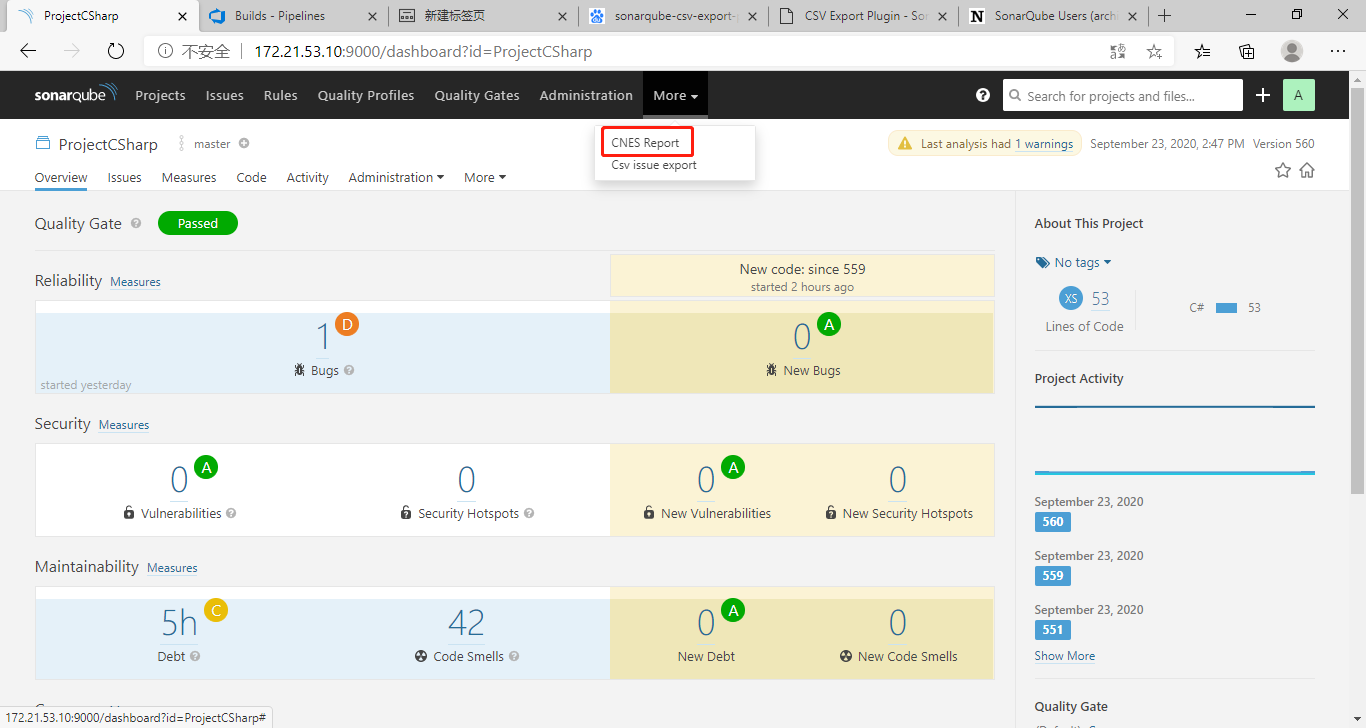
在 SonarQube Marketplace中搜索CNES插件, 将下载的插件.jar文件复制到SonarQube插件目录extensions/plugins下，重启SonarQube。



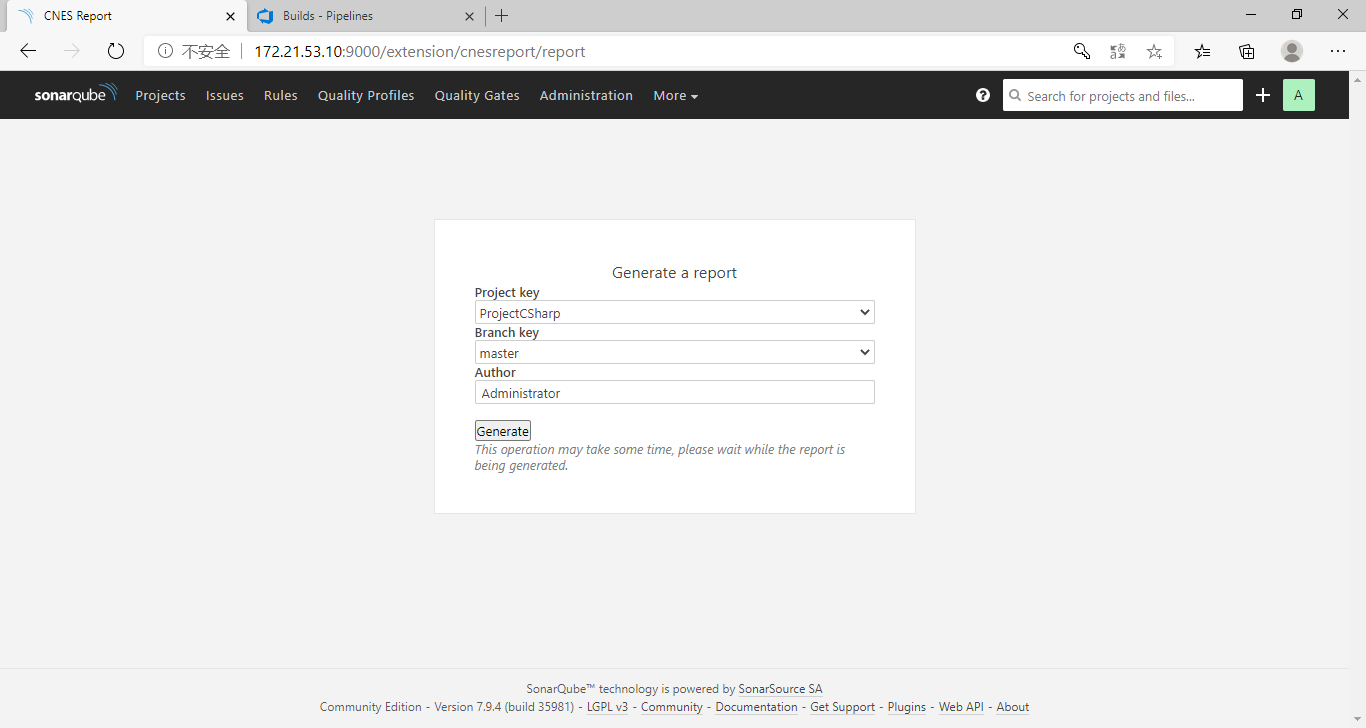


1. **导出CNES报告**

在SonarQube导航栏中点击【More】选项，点击【CNES Report】。



选择导出项目相对应的Project key，点击【Generate】，即可导出报告。



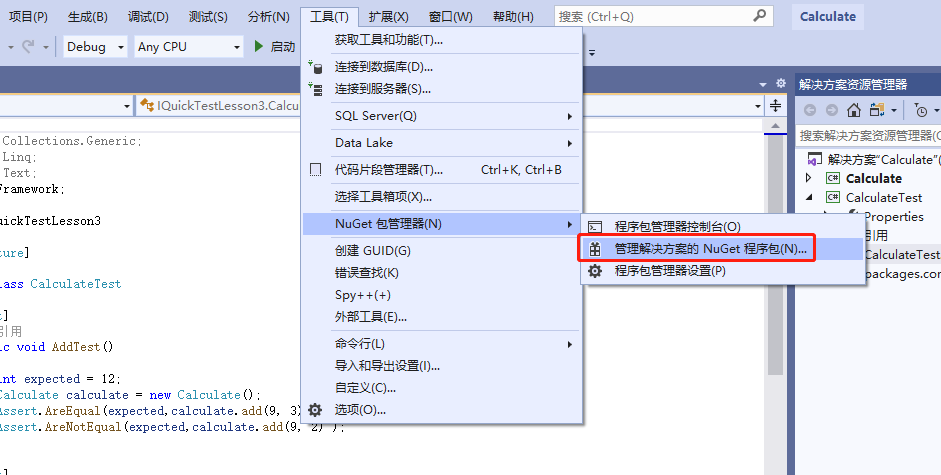
1. **Code Cracker**

Code Cracker 是C＃和VB的分析器库，它使用Roslyn进行重构，代码分析和其他功能。该工具可以应用到Visual Studio和SonarQube，在Visual Studio可以通过两种方式使用CodeCracker：作为与Nuget一起安装到项目中的分析器库，或者作为Visual Studio扩展。在SonarQube中则是通过插件的方式使用，该插件提供Code Cracker规则以分析C＃项目。

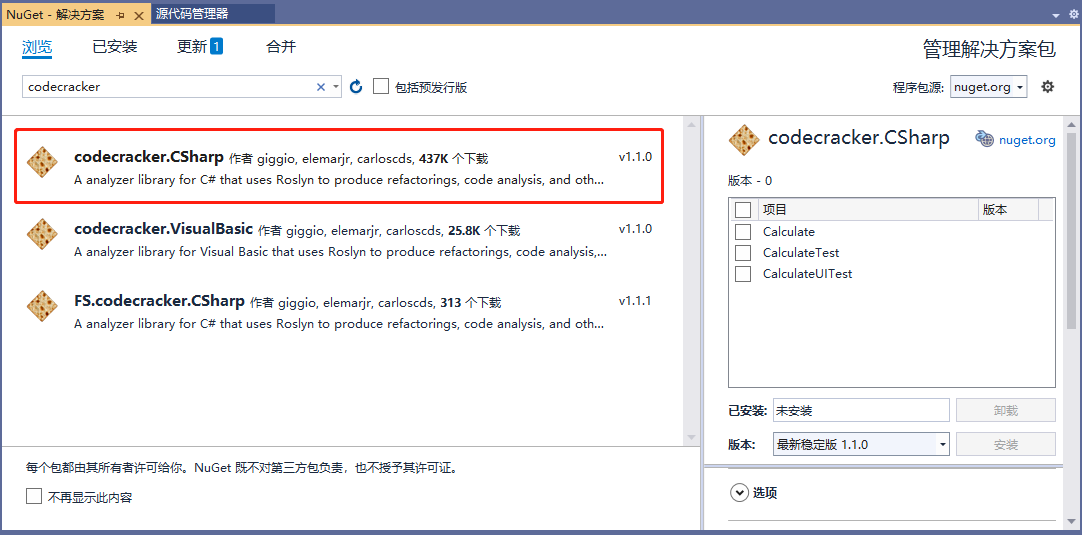
参考文档：https://github.com/code-cracker/code-cracker

1. **Visual Studio中安装Code Cracker分析器**

打开VS，在导航栏的【工具】选项中打开NuGet 包管理器（联网状态）。

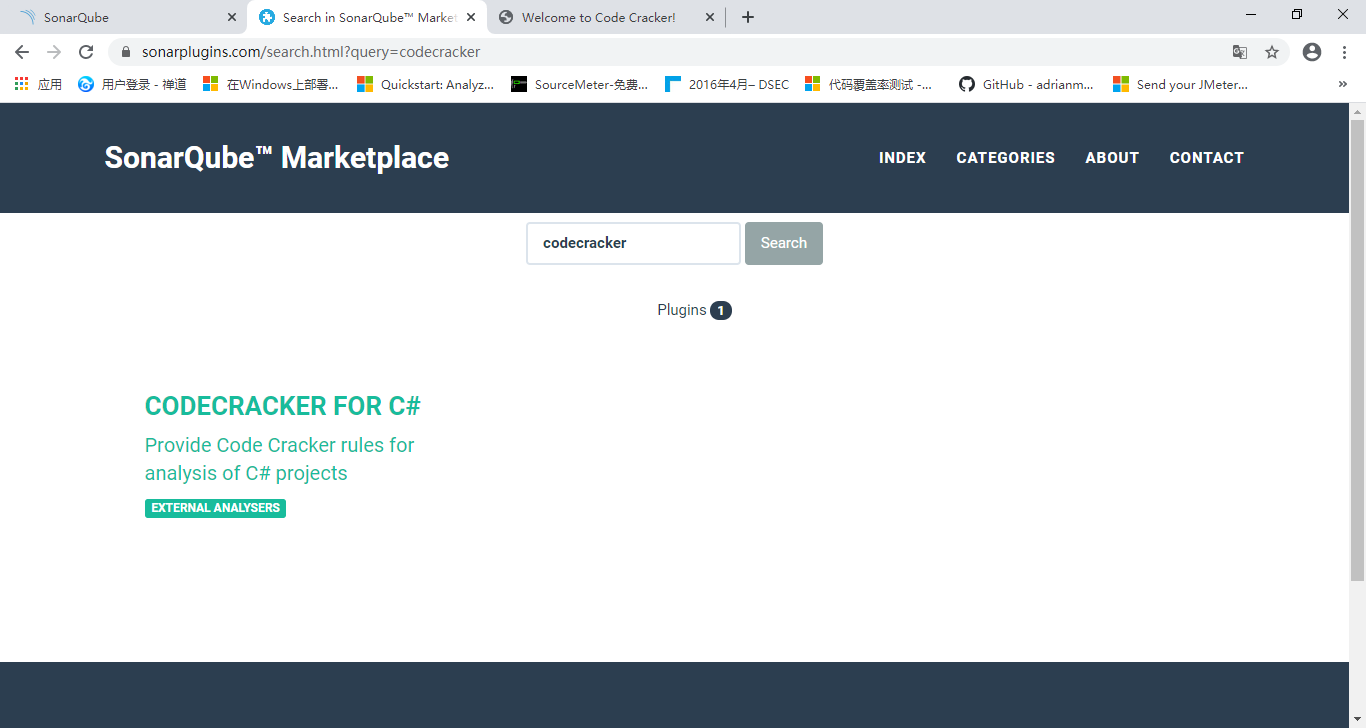


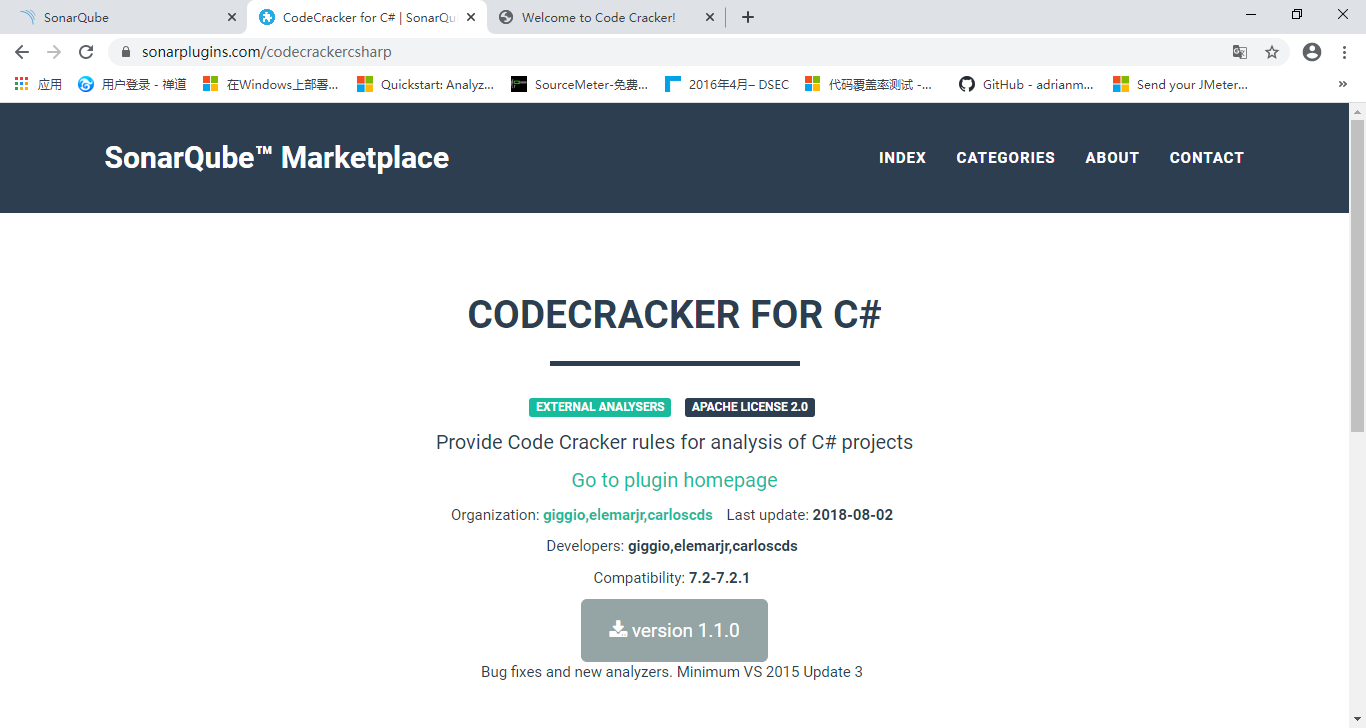
在浏览页面的文本框中输入codecracker，选择codecracker.CSharp进行安装。



1. **安装SonarQube CodeCracker插件**

在 SonarQube Marketplace中搜索codecracker插件, 将下载的SonarQube ReSharper插件.jar文件复制到SonarQube插件目录extensions/plugins下，重启SonarQube。





1. **将CodeCracker规则添加到质量配置文件**

在顶部导航工具栏中选择【Rules】，然后在左侧栏中选择【Repository】，选择【CodeCracker for C#】查看CodeCracker规则列表。

选择【Bulk Change】，然后选择激活输入。从下拉列表中，选择【Sonar Csharp】配置文件。

