# 简介

官网地址：<https://docs.sonarqube.org/latest/>

# 安装

## 快速开始

这里使用docker镜像的方式快速启动sonarqube。

1. 拉取sonar镜像并启动

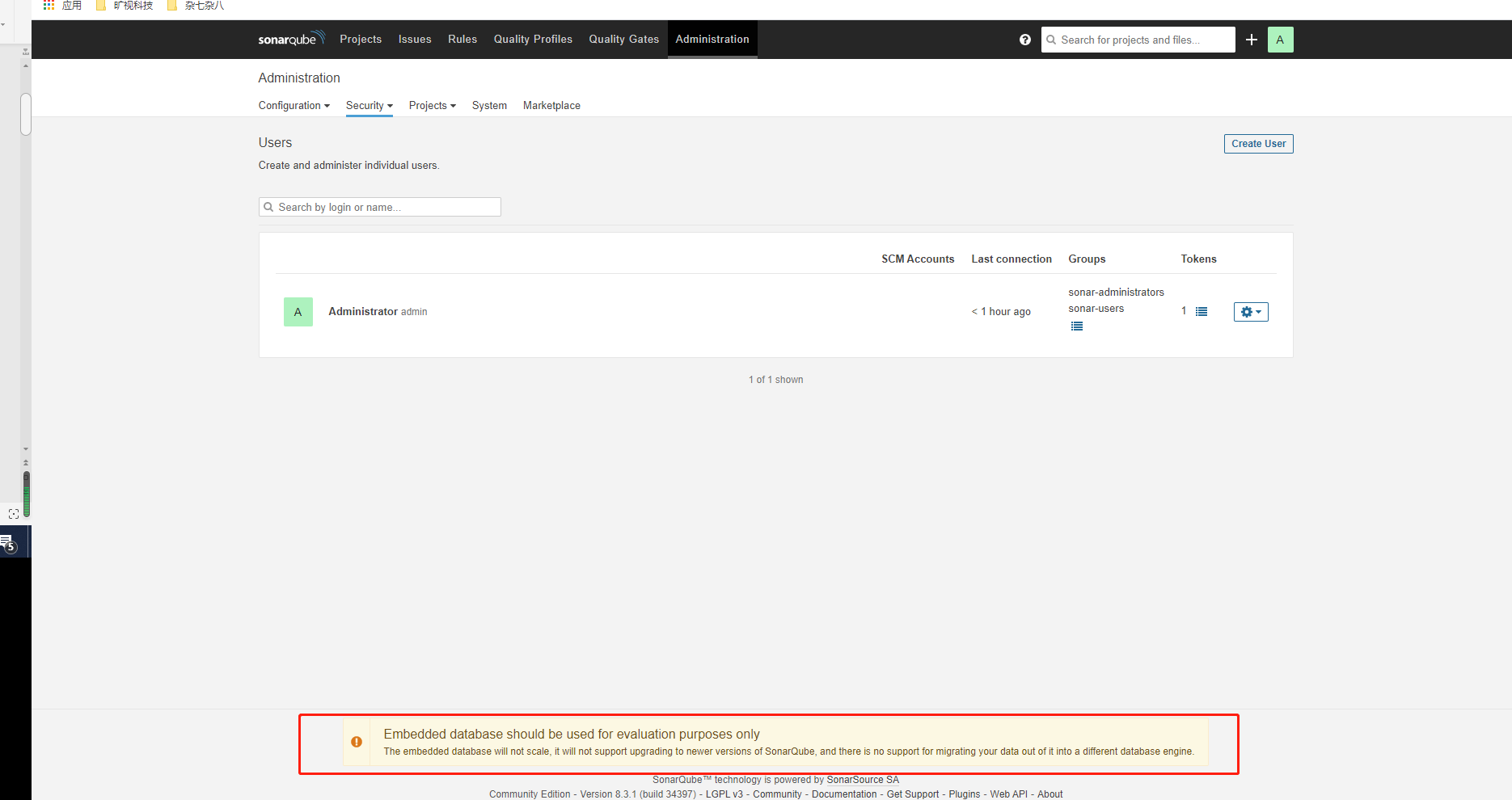
> docker run -d --name sonarqube -p 9000:9000 sonarqube:latest

1. 登录sonar服务器

浏览器输入 [http://localhost:9000](http://localhost:9000/" \t "https://docs.sonarqube.org/latest/setup/get-started-2-minutes/_blank) 进行登录，账号：admin，密码：admin

## sonarqube服务器（Docker镜像方式）

上面这种方式安装的sonarqube不能用于生产环境，sonarqube也在web页面给出了提示，如下：



根据提示，主要是因为快速开始的方式的数据库是内置的，所以我们需要用对它进行配置，当然，还有一些其他的原因。

## sonarqube服务器（压缩包方式）

# 初步检查项目代码

### 在sonar上创建项目

1. 在sonar上面创建一个项目，主要是要取名，选择token，这里输入之前创建好的token，或者重新生成；
2. 在要检查的项目下执行下面的命令

|  |
| --- |
| mvn sonar:sonar -Dsonar.projectKey=designmodel -Dsonar.host.url=http://localhost:9000 -Dsonar.login=fb3ff39628f17675af5ca48c5e3a123b516b6ca8 |

其中 projectKey 就是sonar上面创建的项目名，login是对应的token。

这种方式一般不是很推荐，操作起来太过复杂。

### maven插件方式生成

1. 配置Maven插件

在maven全局settings标签下面添加：

|  |
| --- |
| <pluginGroups>  <pluginGroup>org.sonarsource.scanner.maven</pluginGroup>  </pluginGroups>  <profiles>  <profile>  <id>sonar</id>  <activation>  <activeByDefault>true</activeByDefault>  </activation>  <properties>  <!-- Optional URL to server. Default value is http://localhost:9000 -->  <sonar.host.url>  http://localhost:9000  </sonar.host.url>  </properties>  </profile>  </profiles> |

注意更改里面的host地址

1. 执行mvn 命令

此时执行maven命令的时候，会自动的上传报告到sonar服务器

# 集成jacoco生成测试覆盖率报告

### 直接执行maven命令

|  |
| --- |
| mvn clean org.jacoco:jacoco-maven-plugin:prepare-agent package org.jacoco:jacoco-maven-plugin:report sonar:sonar |

### 简化maven命令

1. 在项目中添加maven插件

|  |
| --- |
| <plugins>  <plugin>  <groupId>org.jacoco</groupId>  <artifactId>jacoco-maven-plugin</artifactId>  <version>0.8.4</version>  <executions>  <execution>  <id>prepare-agent</id>  <goals>  <goal>prepare-agent</goal>  </goals>  </execution>  <execution>  <id>report</id>  <goals>  <goal>report</goal>  </goals>  </execution>  </executions>  </plugin>  </plugins> |

1. 执行命令

|  |
| --- |
| mvn clean jacoco:prepare-agent package jacoco:report sonar:sonar |

具体配置可以参考

<https://docs.sonarqube.org/latest/analysis/scan/sonarscanner-for-maven/>

# 集成sonar, jacoco到Jenkins

集成sonar可以进行代码质量检查， 而引入了jacoco则可以检查测试用例的覆盖率。

### 插件方式

通过给Jenkins安装sonarqube插件的方式进行代码扫描。

### maven命令方式

这种方式，通过如下两步操作即可

1. 配置maven

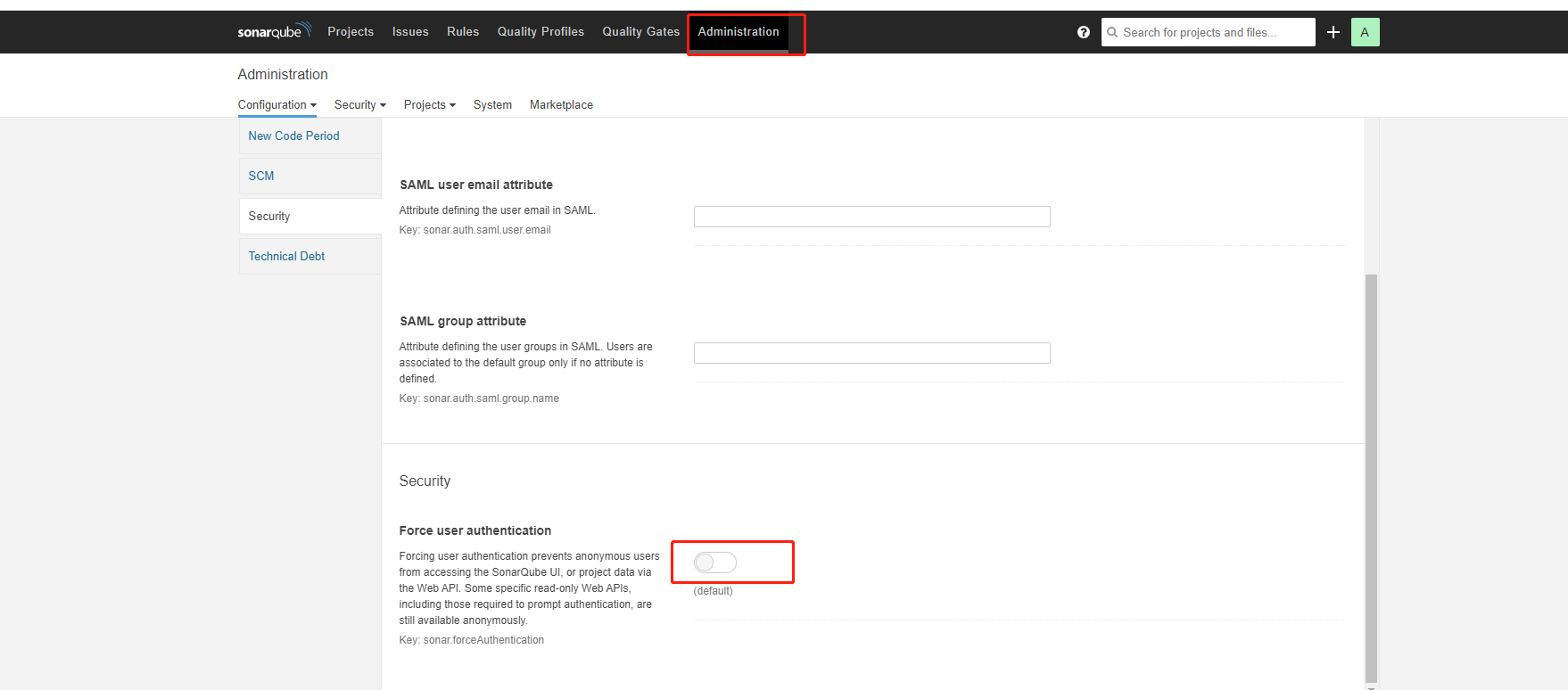
对jenkins使用的maven的settings.xml文件进行配置，配置方式和上面的一致；

1. jenkins打包的时候，指定maven命令带上jacoco agent，如下：

|  |
| --- |
| clean test org.jacoco:jacoco-maven-plugin:prepare-agent install org.jacoco:jacoco-maven-plugin:report sonar:sonar |

# 开启登录认证

默认情况下，sonar不需要做登录就可以进行代码分析，我们可以如下方式开启认证：



这样的话，需要分析的项目需要配置账号密码。

### Maven项目的配置

我们需要在maven的pom文件中配置如下内容：

|  |
| --- |
| <properties>  <sonar.login>3d6c973435a763f53d0d82b8b132db78cfca5943</sonar.login>  <sonar.password>admin</sonar.password> </properties> |

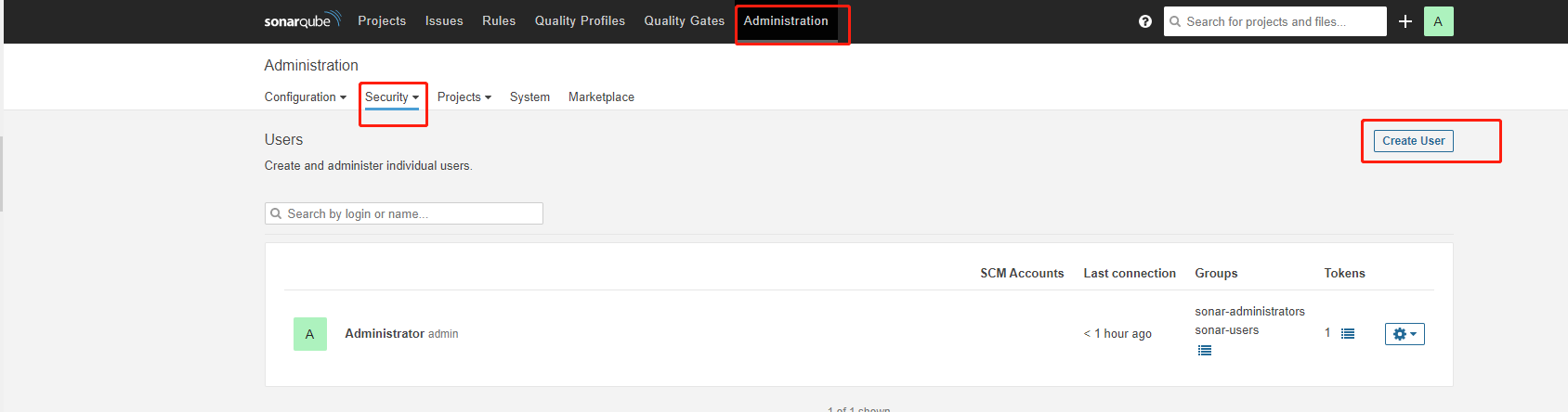
有两种配置方式

1. 使用token

如果我们使用token进行认证，那么我们只需要配置sonar.login即可，里面填写之前生成的token

1. 使用账号密码

此时需要同时填写sonar.login和sonar.password，这个账号密码可以使用默认的 admin/admin，也可以自己通过如下方式创建：



### Jenkins中配置