



链滴

gitlab 根据 sonar 扫描状态决定是否允许 merge

作者: [fish2018](#)

原文链接: <https://ld246.com/article/1565068821651>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



有这样一个应用场景：

当开发提交merge request时，gitlab会自动触发jenkins任务去跑sonar扫描，如果扫描状态成功则允许merge，否则拒绝。

这里有几个点需要解决：

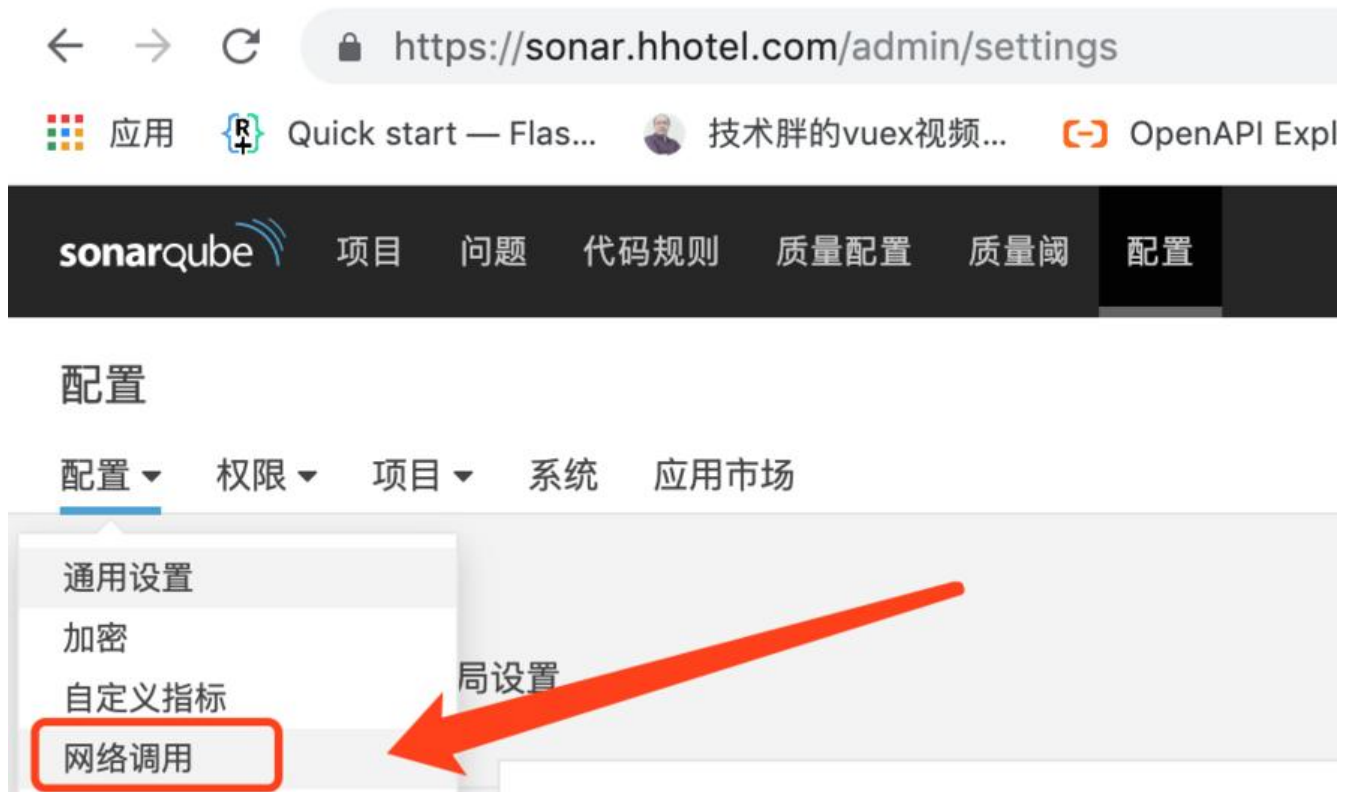
- 1、jenkins要拿到sonar扫描的结果状态，并把结果反馈给gitlab
- 2、gitlab要能够根据pipeline反馈的状态来决定是否允许merge

gitlab通过webhook触发jenkins任务的配置略

设置gitlab，只有pipeline成功才允许merge

The screenshot shows the GitLab interface for a project named 'bzy-hotel'. The left sidebar contains navigation options: Project, Repository, Issues (0), Merge Requests (0), CI / CD, Operations, Wiki, Snippets, Settings (highlighted with a red arrow), General, Members, and Badges. The main content area is titled '合并请求设置' (Merge Request Settings) and includes a sub-header '自定义您的合并请求限制。' (Customize your merge request restrictions). Under the 'Merge method' section, three options are listed: 'Merge commit' (selected), 'Merge commit with semi-linear history', and 'Fast-forward merge'. Below these, there are four checkboxes: '只允许合并流水线成功的合并请求。' (highlighted with a red box), '只允许合并所有讨论都已解决的合并请求。', '当它们过时的时候，自动解决合并请求差异讨论。', and '从命令行推送时显示创建/查看合并请求的链接'. A green '保存修改' (Save changes) button is at the bottom.

部署sonar略, 直接看webhook配置



点击创建, url填"jenkins地址/sonarqube-webhook/"

创建网络调用

名称*



URL*



接收网络调用负载的服务器地址，比如："http://my_server/foo"。如果要使用HTTP Basic认证，推荐使用HTTPS防御中间人攻击。比如："https://myLogin:myPassword@my_server/foo"

密码



如果提供了密码，会用来生成16进制（小写）HMAC摘要，对应值会包含在'X-Sonar-Webhook-HMAC-SHA256'头部。

创建

取消

配置

配置 ▾ 权限 ▾ 项目 ▾ 系统 应用市场

网络调用

创建

网络调用在任务分析完成后提醒外部服务。一个包含JSON负载的HTTP POST请求会发送给每个设置的URL。参考 [网络调用文档](#)。

名称	URL	密码?	最后信息
webhook	http://jenkins.hhotel.com/sonarqube-webhook/	否	2019年7月24日 上午11:21

jenkins要安装插件[Sonar Quality Gates Plugin](#)

这样pipeline中就可以获取sonar执行后的状态了

[updateGitlabCommitStatus](#)用来更新状态到gitlab,只有成功gitlab才会允许merge就实现了

```

stage ('静态扫描') {
  steps {
    updateGitlabCommitStatus name: 'build', state: 'running'

    script {
      withSonarQubeEnv('sonar') {
        sh "mvn validate sonar:sonar -Dsonar.java.binaries=target/sonar"
      }
      def qg = waitForQualityGate()

      if (qg.status != 'OK') {
        error "未通过Sonarqube的代码质量阈检查, 请及时修改! failure: ${qg.status}"
        updateGitlabCommitStatus name: 'build', state: 'failed'
      }
      if (qg.status == 'OK') {
        echo "通过Sonarqube的代码质量检测"
        updateGitlabCommitStatus name: 'build', state: 'success'
      }
    }
  }
}

```

如果想拿到sonar扫描后生成的url发邮件，可以通过一个脚本实现

```

#!/bin/sh
# 获取sonar扫描后返回的url, jenkins中使用方法: /data/tools/sonarurl ${JOB_URL}
JOB_URL=${1/jenkins.hhotel.com/127.0.0.1:8080}
id=`wget -qO- --content-on-error --no-proxy --auth-no-challenge --http-user=admin --http-
assword=297UVZU0u*5*1KNQ "${JOB_URL}/lastBuild/consoleText" | grep "More about the r
port processing" | head -n1 | awk -F "=" '{print $2}`
projectkey=`wget -qO- "http://sonar.hhotel.com/api/ce/task?id=${id}" --no-proxy --content-
n-error | jq -r '.task' | jq -r '.componentKey`
sonarurl=http://sonar.hhotel.com/dashboard?id=${projectkey}
echo ${sonarurl}

```

令牌获取方式

令牌

如果想强化安全，不想在执行代码扫描或调用Web Service时使用真实SonarQube用户的密码，可以使用用户令牌来代替用户登录。这样可以通过避免把分析用户的密码在网络传输，从而提升安全性。

生成令牌

sonar

名称	最后使用	已创建	
jenkins	13天前	2019年7月11日	<input type="button" value="回收"/>

maven配置sonar, 修改settings.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<settings xmlns="http://maven.apache.org/SETTINGS/1.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/SETTINGS/1.0.0 http://maven.apache.org/xsd
  settings-1.0.0.xsd">

<servers>
  <server>
    <id>maven-release</id>
    <username>devOps</username>
    <password>devOps20190717</password>
  </server>

  <server>
    <id>maven-snapshots</id>
    <username>devOps</username>
    <password>devOps20190717</password>
  </server>
</servers>

<pluginGroups>
  <pluginGroup>org.sonarsource.scanner.maven</pluginGroup>
</pluginGroups>

<profiles>
  <profile>
    <id>sonar</id>
    <activation>
      <activeByDefault>true</activeByDefault>
    </activation>
    <properties>
      <sonar.host.url>
        http://sonar.hhotel.com
      </sonar.host.url>
    </properties>
  </profile>

  <profile>
    <id>NexusRepo</id>
    <repositories>
      <repository>
        <id>nexus</id>
        <name>Nexus3 Repository</name>
        <url>http://172.19.151.229:8082/nexus/repository/maven-public/</url>
        <releases>
          <enabled>true</enabled>
        </releases>

        <snapshots>
          <enabled>true</enabled>
        </snapshots>
      </repository>
```

```
</repositories>
</profile>
</profiles>

<activeProfiles>
  <activeProfile>sonar</activeProfile>
  <activeProfile>NexusRepo</activeProfile>
</activeProfiles>

<mirrors>
  <mirror>
    <id>NexusRepo</id>
    <!-- *号代表所有仓库，此处也可以单独设置，以逗号隔开 -->
    <mirrorOf>*</mirrorOf>
    <name>NexusRepo</name>
    <url>http://172.19.151.229:8082/nexus/repository/maven-public/</url>
  </mirror>
</mirrors>
</settings>
```

这样就可以通过mvn命令执行sonar扫描了

```
mvn sonar:sonar
```