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

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。

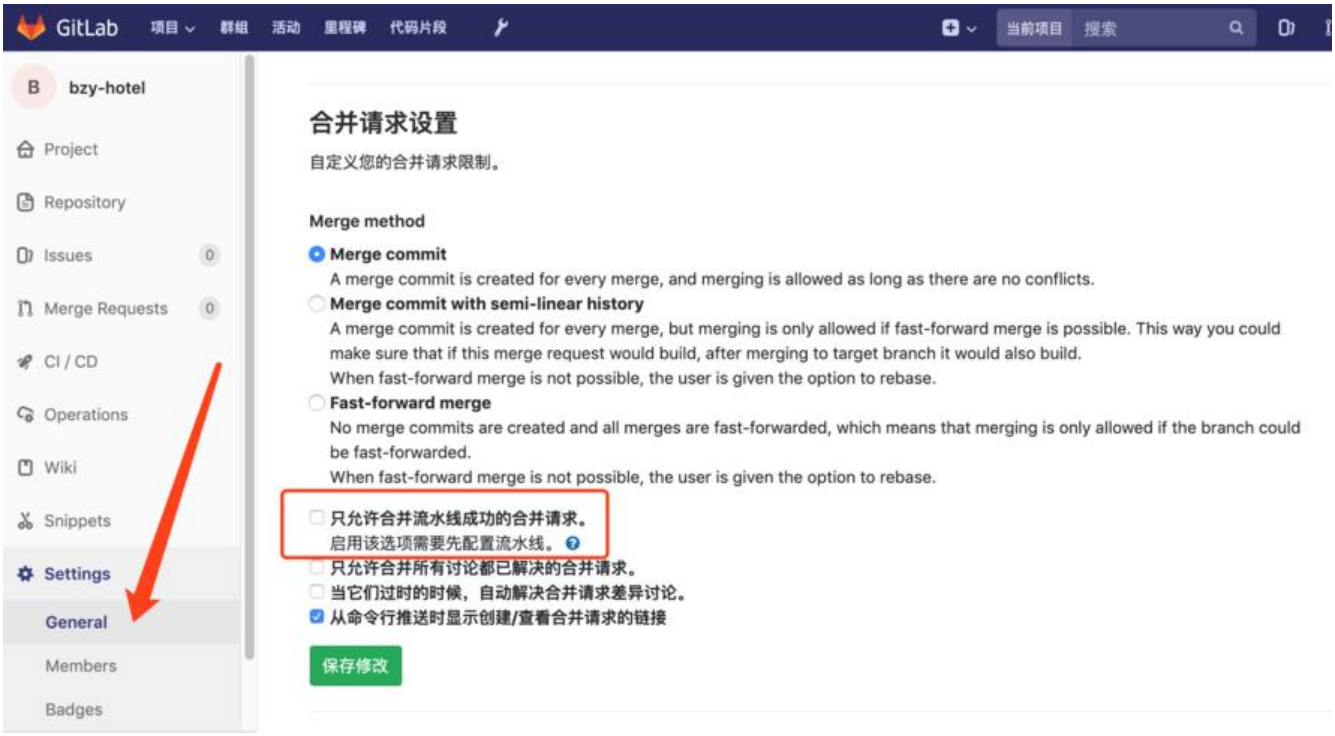
这里有几个点需要解决：

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

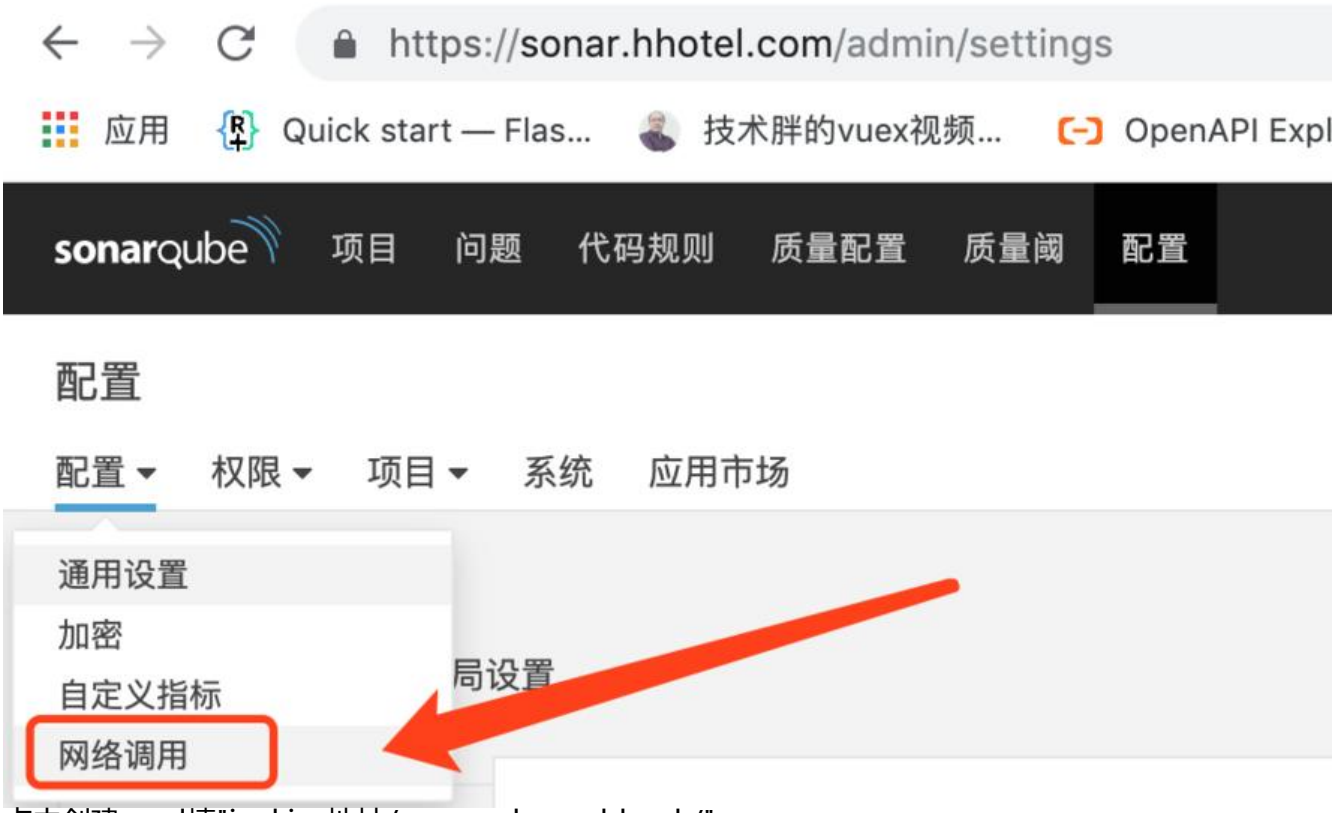
gitlab通过webhook触发jenkins任务的配置略

设置gitlab，只有pipeline成功才允许merge



原文链接：gitlab 根据 sonar 扫描状态决定是否允许 merge

部署sonar略，直接看webhook配置



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

原文链接：gitlab 根据 sonar 扫描状态决定是否允许 merge

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

原文链接：gitlab 根据 sonar 扫描状态决定是否允许 merge

```
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

原文链接：gitlab 根据 sonar 扫描状态决定是否允许 merge

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</usernam>
    <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
```