

jenkins发布java项目

过程参考发布vue项目。<https://www.cnblogs.com/jxd283465/p/11543431.html> 大同小异。

vue建立的是Freestyle project，构建环境选择node

java建立的是maven project，构建环境使用maven

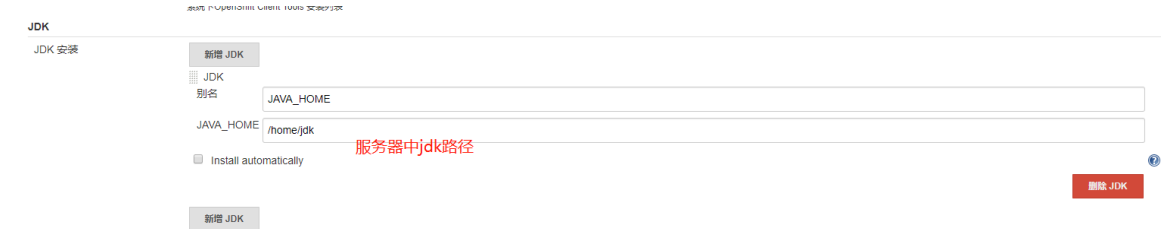
1. 配置全局工具中maven

maven及maven私服安装参考：<https://www.cnblogs.com/jxd283465/p/11548593.html>

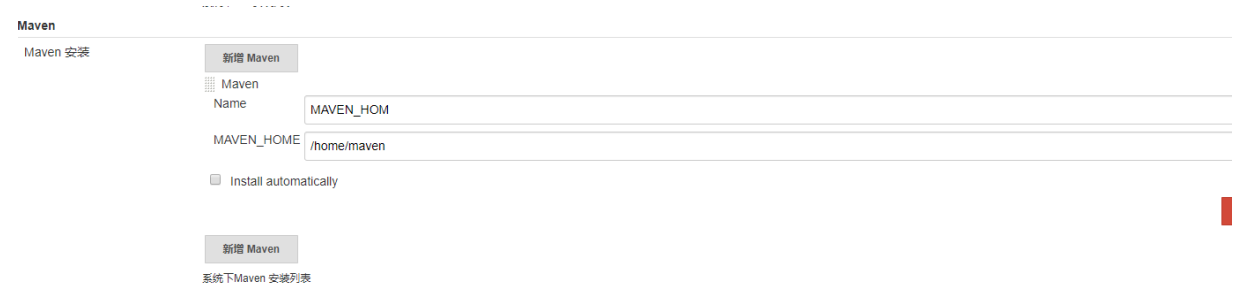
配置maven的配置文件



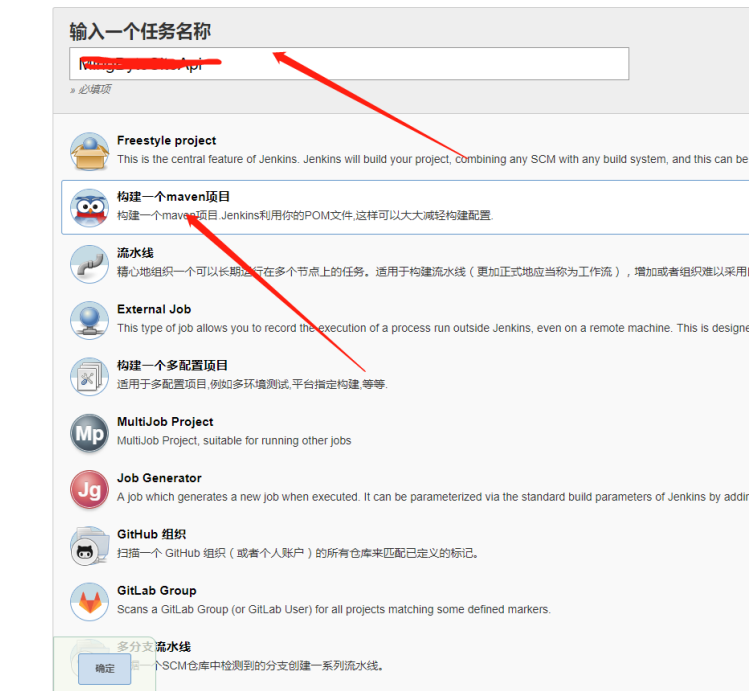
配置jdk安装路径



配置maven安装路径



2. 新建maven项目



3. 配置项目描述、git参数

General

源码管理

构建触发器

构建环境

Pre Steps

Build

Post Steps

构建设置

构建后操作

描述

项目描述

Discard old builds

策略

Log Rotation

保持构建的天数

5

如果非空，构建记录将保存此天数

保持构建的最大个数

5

如果非空，最多此数目的构建记录将被保存

丢弃旧的构建

高级...

This project is parameterized

Git Parameter

Name

Tag

Description

如需发布新版本则点击'开始构建'。如需回退版本，请选择需要回退的Tag

Parameter Type

Tag

Default Value

Default Value is required. Example origin/master

添加参数

高级...

4. 配置源码管理git

如果使用sonar插件一定要配置Name

Branch Specifier为上方参数化构建的name，\$表示调用参数

源码管理

无

Git

Repositories

Repository URL

Credentials

jenkins

Name

MingByteSiteApi

Refspec

下方配置sonar需要用

Add Repository

Branches to build

Branch Specifier (blank for 'any')

\$Tag

上方参数化构建的Name

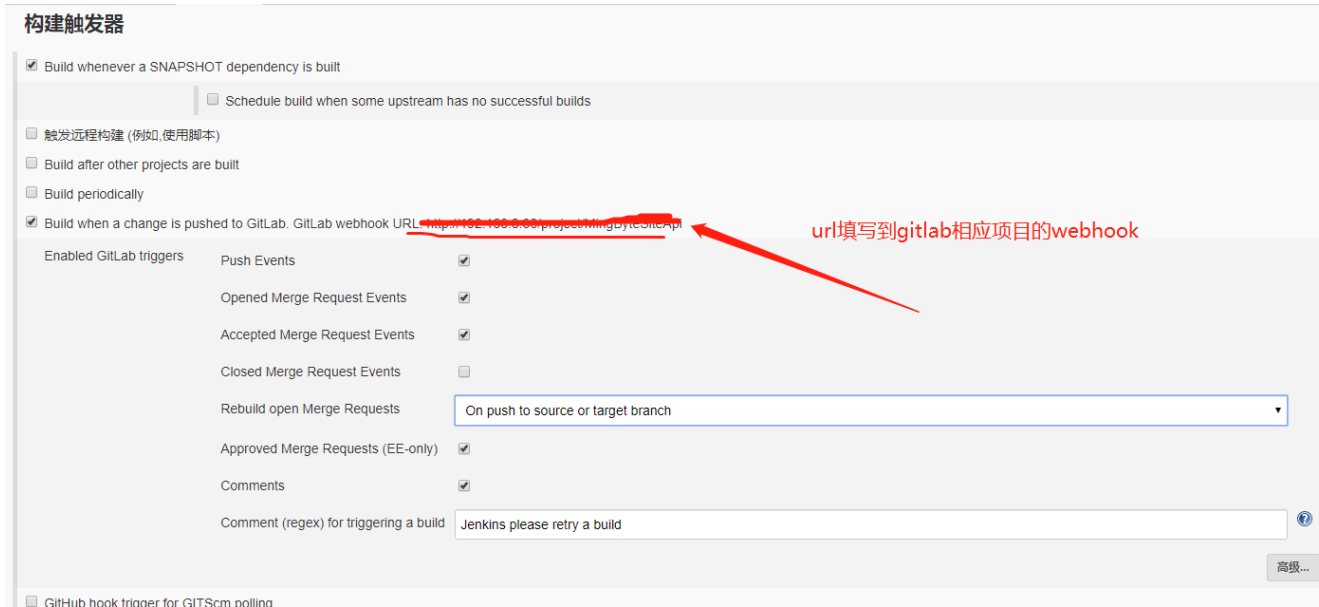
Add Branch

源码库浏览器

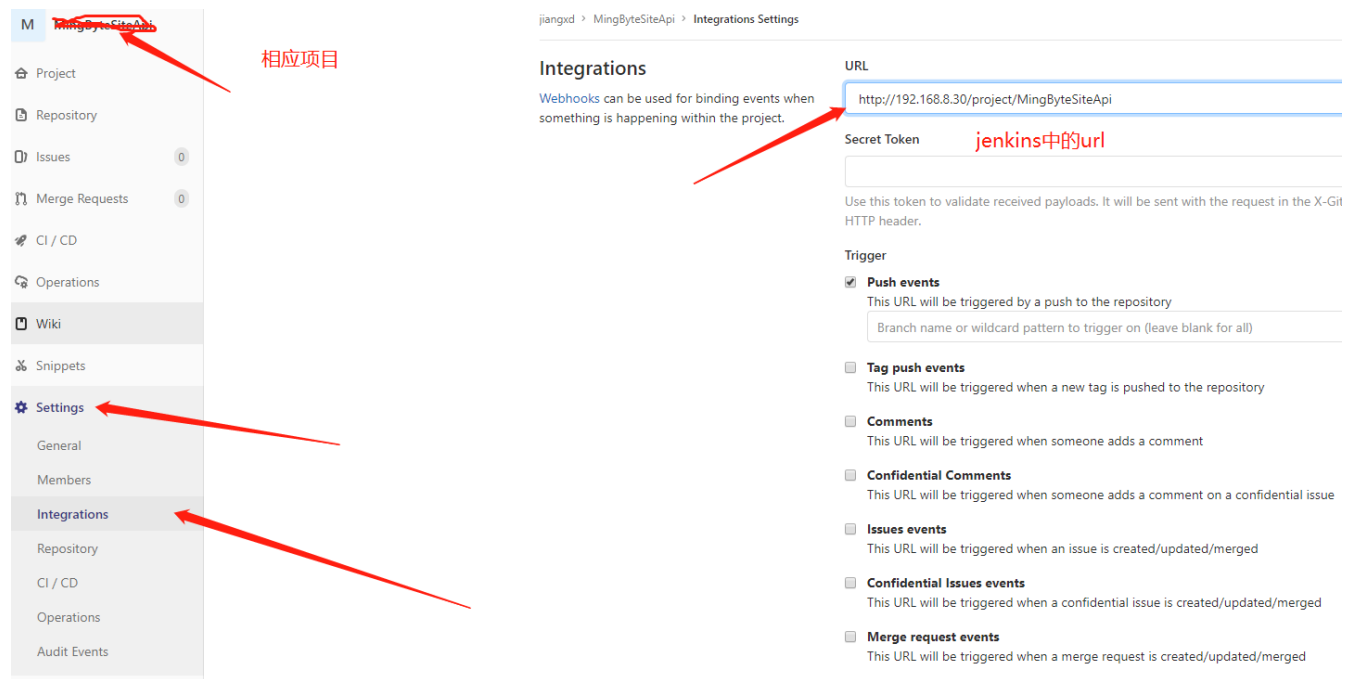
(自动)

5. 配置触发器

当提交代码到gitlab时，自动触发jenkins自动发布



gitlab webhook配置

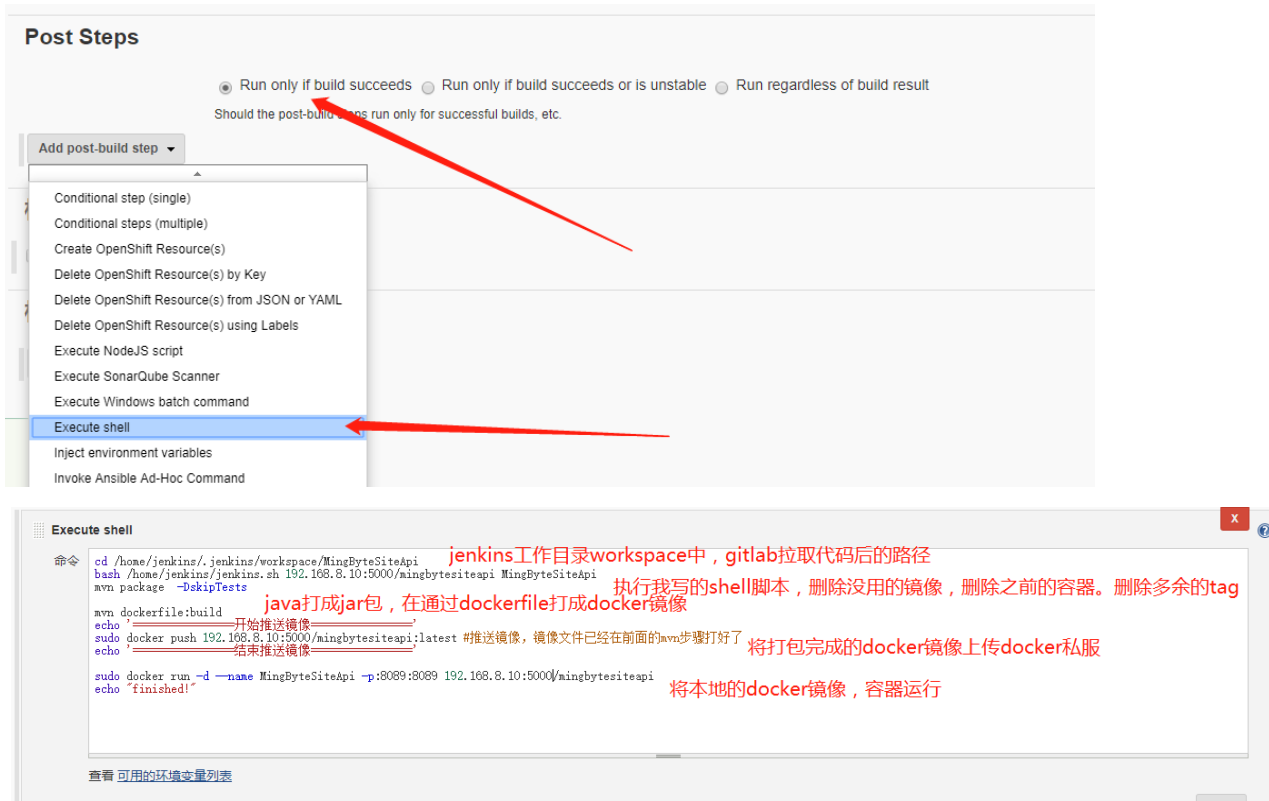


6. 配置构建环境



7. 配置启动shell

勾选 “Run only if build succeeds” ，当项目构建成功执行shell



```

cd /home/jenkins/.jenkins/workspace/MingByteSiteApi
bash /home/jenkins/jenkins.sh 192.168.8.10:5000/mingbytesiteapi MingByteSiteApi
mvn package -DskipTests

mvn dockerfile:build
echo '====='
sudo docker push 192.168.8.10:5000/mingbytesiteapi:latest #推送镜像, 镜像文件已经在前面的mvn步骤打好了
echo '====='
sudo docker run -d --name MingByteSiteApi -p:8089:8089 192.168.8.10:5000/mingbytesiteapi
echo "finished!"

```

### jenkins.sh

```

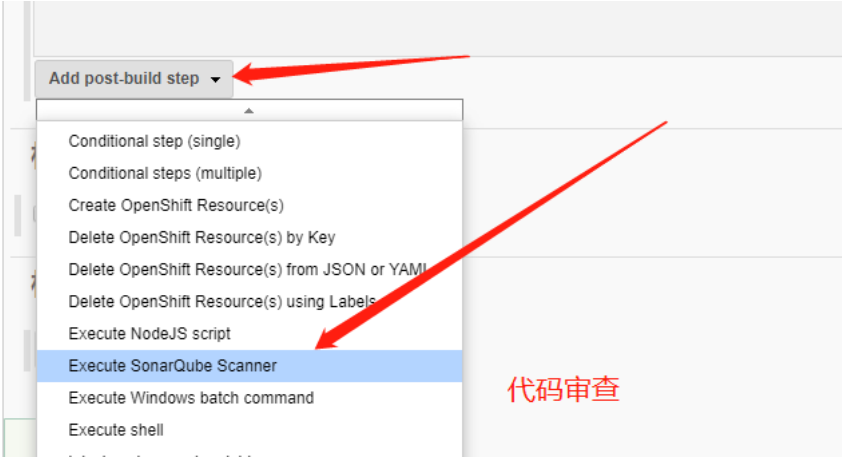
t='sudo docker ps | grep $1|awk '{print $1}'|sed 's/%%//g'`;
b='';
if [ $t ];
then
sudo docker stop $t
echo "停止容器成功"
sudo docker rm $t
echo "删除容器成功"
else
echo "首次部署";
fi

tagnum=`git tag | wc -l`;
tag=( git tag )
a=0;
echo $tagnum
until [ $tagnum -lt 3 ]
do
echo $a 删除tag: ${tag[$a]}
git tag -d ${tag[$a]}
git push $2 :refs/tags/${tag[$a]}
a=`expr $a + 1`
tagnum=`expr $tagnum - 1`
done

```

### 8. 集成sonar代码审查

整合sonar代码审查参考：<https://www.cnblogs.com/jxd283465/p/11548590.html>



**Execute SonarQube Scanner**

Task to run:

JDK:

JDK to be used for this SonarQube analysis:

Path to project properties:

Analysis properties: 

```
sonar.projectVersion=1.0
sonar.sources=src/main/java
sonar.java.binaries=target/classes
sonar.language=java
sonar.sourceEncoding=UTF-8
```

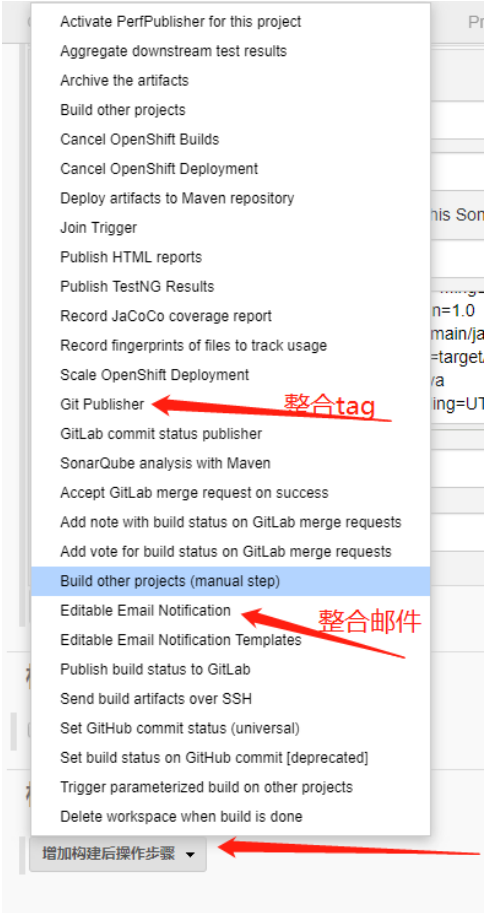
Additional arguments:

JVM Options:

9. 整合tag打包备份、整合邮件提醒

整合tag备份参考：<https://www.cnblogs.com/jxd283465/p/11548589.html>

整合邮件提醒参考：<https://www.cnblogs.com/jxd283465/p/11548580.html>



Git Publisher

Push Only If Build Succeeds

☒

Merge Results

☐

Force Push

☐

Tags

Tag to push

rc\_\${BUILD\_NUMBER}

Tag message

版本回退

Create new tag

☒

Update new tag

☐

Target remote name

MingByteSiteApi

Add Tag

Tags to push to remote repositories

Add Branch

Branches to push to remote repositories

Add Note

Notes to push to remote repositories

项目构建成功，push tag

tag名字为项目构建序号

创建一个新的Tag

上文中gitlab的Name

Editable Email Notification

Disable Extended Email Publisher

☐

Project From

Project Recipient List

收信人

Project Reply-To List

\$DEFAULT\_REPLYTO

Content Type

Default Content Type

Default Subject

\$DEFAULT\_SUBJECT

邮件标题

Default Content

\$DEFAULT\_CONTENT

这里调用的系统配置中的设置，详情参考整合邮件

邮件内容

Attachments

Can use wildcards like 'module/dist/\*\*/\*.zip'. See the @includes of Ant fileset for the exact format. The base directory is the workspace.

Attach Build Log

Content Token Reference

Pre-send Script

Post-send Script

Additional groovy classpath

Save to Workspace

Triggers

Can use wildcards like 'module/dist/\*\*/\*'.zip'. See the [@includes of Ant fileset](#) for the exact format. The base directory is [the workspace](#).

Attach Build Log

\$DEFAULT\_PRESEND\_SCRIPT

\$DEFAULT\_POSTSEND\_SCRIPT

新增

☐

Always

Send To

Developers

Recipient List

新增

Add Trigger

每次项目构建都发送邮件

设置邮件发送的时机

X

?

X

?

高级...