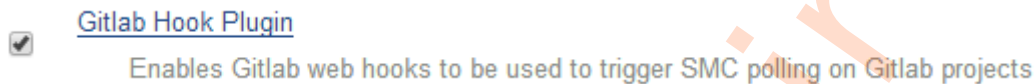


之前部署了Gitlab的代码托管平台和Jenkins的代码发布平台。通常是开发后的代码先推到Gitlab上管理，然后在Jenkins里通过脚本构建代码发布。这种方式每次在发版的时候，需要人工去执行jenkins上的构建动作，有时显得过于繁琐。于是就想到了Gitlab的Webhook功能，通过Webhook的相关设置，可以实现代码Push后，自动去触发jenkins上的构建动作，这样就不需要人工干预去执行发版操作了。设置如下：

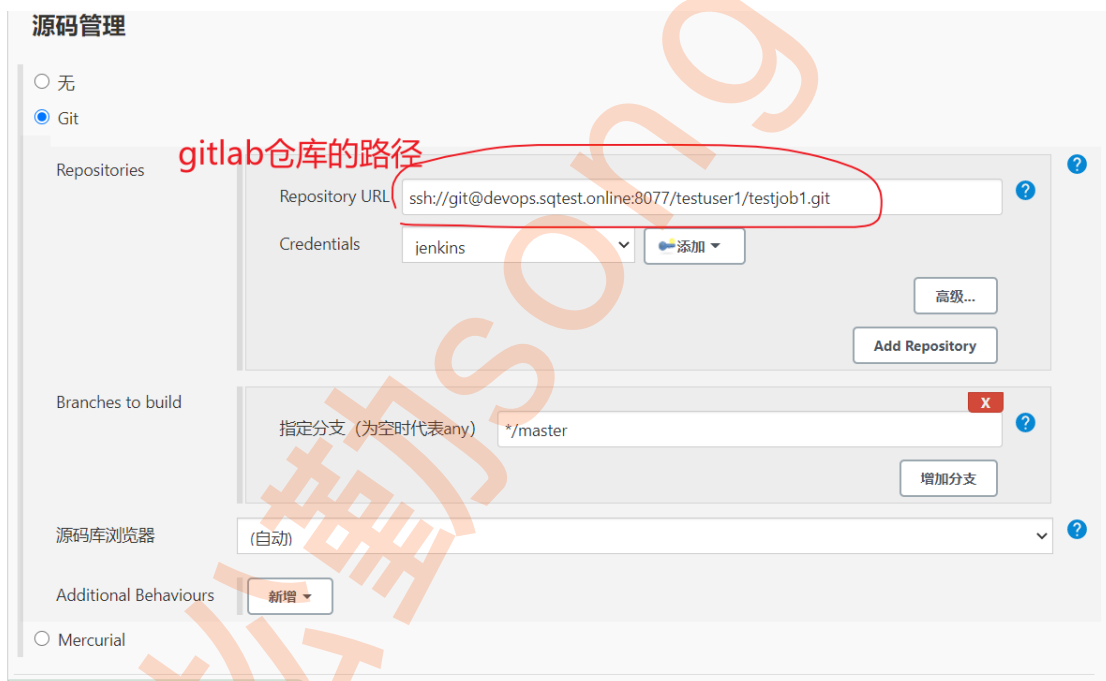
1) Jenkins上的操作

通常，jenkins在项目配置时是可以配置远程构建触发器，设置好jenkins的回调url后就可以让jenkins进行自动构建。这就需要先下载Gitlab Hook Plugin 插件。登陆Jenkins，分别点击"系统管理"->"插件管理"->"可选插件"，选择Gitlab Hook Plugin，(公网环境<http://devops.sqtest.online:7073/> 已经提前安装好，可以忽略这步)



然后构建相应的项目工程，比如下面的jenkins-demo-api-dev（下面最关键的是第2步中查看url，然后用这个url回调地址去Gitlab的webhook上添加）

a) 先设置代码的git下载路径，这里通过ssh方式（需要提前将Jenkins本机的key添加到Gitlab上）



b) 查看jenkin生成回调地址。在任务重建构建触发器下获取回调URL。下面的URL那一行只有Gitlab Hook Plugin插件下载成功后才能显示。

General 源码管理 **构建触发器** 构建环境 构建 构建后操作

构建触发器

☐ 触发远程构建 (例如,使用脚本) ?
☐ 其他工程构建后触发 ?
☐ 定时构建 ?
☒ Build when a change is pushed to GitLab. GitLab webhook URL: http://devops.sqtest.online:7073/project/jenkins-demo-api-dev ?

Enabled GitLab triggers	Push Events
	<input checked="" type="checkbox"/>
	<input checked="" type="checkbox"/>
	<input type="checkbox"/>
	<input type="checkbox"/>
	Rebuild open Merge Requests Never
	<input checked="" type="checkbox"/>
	<input checked="" type="checkbox"/>
	Comment (regex) for triggering a build Jenkins please retry a build ?

高级...

c) 点开下面的高级，生成密钥并保存

Enable [ci-skip] ☒
Ignore WIP Merge Requests ☒
Set build description to build cause (eg. Merge request or Git Push) ☒
Build on successful pipeline events ☐
Pending build name for pipeline
Cancel pending merge request builds on update ☐
Allowed branches
☒ Allow all branches to trigger this job ?
☐ Filter branches by name ?
☐ Filter branches by regex ?
☐ Filter merge request by label
Secret token e570324762...56d8eadc8b0dfd
点击生成密钥，复制保存 Generate Clear

d) 构建命令随便写一个用于测试

构建

执行 shell X ?

命令 `echo "start job"`
`ls`

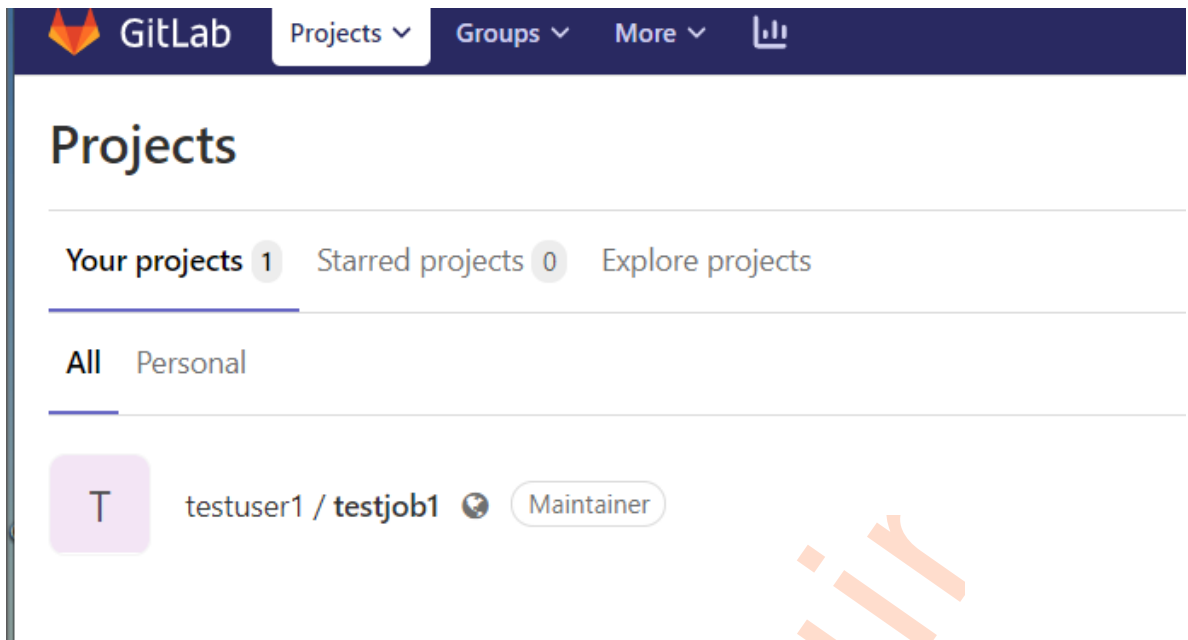
查看 [可用的环境变量列表](#)

高级...

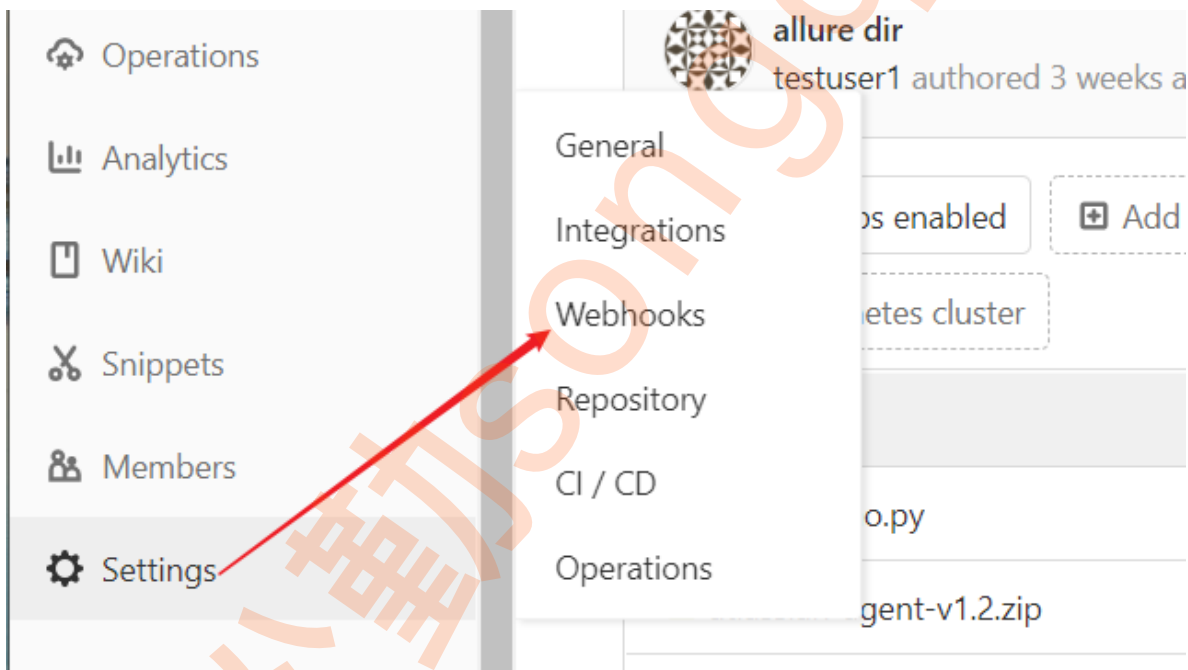
增加构建步骤 ▾

2) Gitlab上的操作

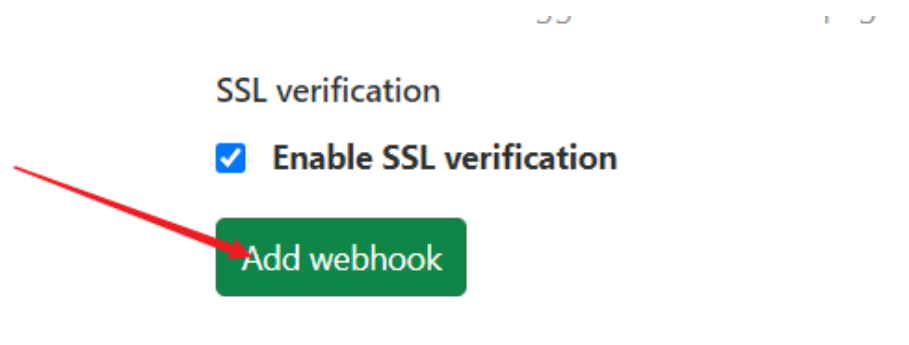
登陆Gitlab,点击testjob1进入项目中, 添加webhook (用上面Jenkins中生成的回调url)



然后点击这个项目的"Settings",选择webhooks



点击添加webhooks



填写相关信息, 注意gitlab和Jenkins必须处于同一个网络, 如果使用公网环境的, 注意配置jenkins-job地址需要做下IP和端口转换, 和截图上保持一致

Webhook

填写Jenkins-job地址, 这里host和port注意替换一下, 非公网地址

Webhooks enable you to send notifications to web applications in response to events in a group or project. We recommend using an [integration](#) in preference to a webhook.

Secret Token

把之前复制的密钥粘贴在这里

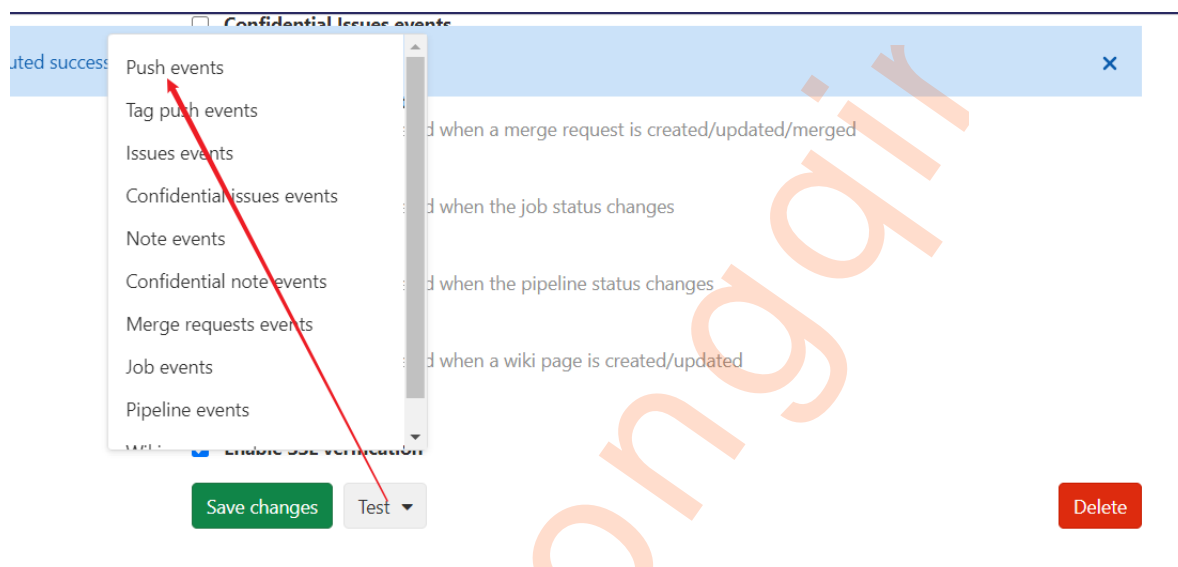
Use this token to validate received payloads. It will be sent with the request in the X-Gitlab-Token HTTP header.

Trigger

☒ Push events

This URL will be triggered by a push to the repository

添加成功后, 点击此webhook后面的test进行测试



Verifies

GitLab triggers a request to use the request URL if something

Status	Trigger	URL	Elapsed time	Request time	
200	Push Hook	http://192.168.122.41:9002/project/jenkins-demo...	0.04 sec	just now	View details

测试成功

如果返回Hook successfully executed.表示配置成功。

这样, 下次push代码后, 就会自动触发jenkins上相关的构建工程进行自动发布了! 无需人工干预~

如上设置后, 比如在gitlab的jenkins-demo-api-dev项目代码库中push一个py文件, 然后测试下是否自动发布了

T

testjob1

Project ID: 1

🔔

☆ Star

0

🍴 Fork

0

🔗 5 Commits

🌿 1 Branch

🏷️ 0 Tags

📁 1.1 MB Files

💾 1.1 MB Storage

master

testjob1 / +


History

Find file

Web IDE

📄

Clone

 allure dir

testuser1 authored 3 weeks ago

3ea7eed1

⚙️ Auto DevOps enabled




📖 Add README

📖 Add LICENSE

📖 Add CHANGELOG

📖 Add CONTRIBUTING

📖 Add Kubernetes cluster

Name	Last commit	Last update
 allure_demo.py	allure dir	3 weeks ago
 atlassian-agent-v1.2.zip	jar	1 month ago
 case001.txt	123	1 month ago

然后去Jenkins上查看，如下发现已经自动触发构建了！

Jenkinsjenkins-demo-api-dev

🏠 返回面板

🔍 状态

📝 修改记录

📁 工作空间

🚀 立即构建

🚫 删除工程

⚙️ 配置

★ 收藏夹

🌐 打开 Blue Ocean

📁 重命名

📁 工作区

📝 最新修改记录

相关链接

- 最近一次构建(#11),5分43秒之前
- 最近稳定构建(#11),5分43秒之前
- 最近成功的构建(#11),5分43秒之前
- 最近失败的构建(#3),1月3天之前
- 最近未成功的构建(#3),1月3天之前
- 最近完成的构建(#11),5分43秒之前

Build History构建历史

find

#112020-9-8 上午3:41

Started by GitLab push by testuser1

#102020-9-3 上午5:31

Started by GitLab push by testuser1