

GitLab 自动触发 Jenkins 构建

yijian2595 关注

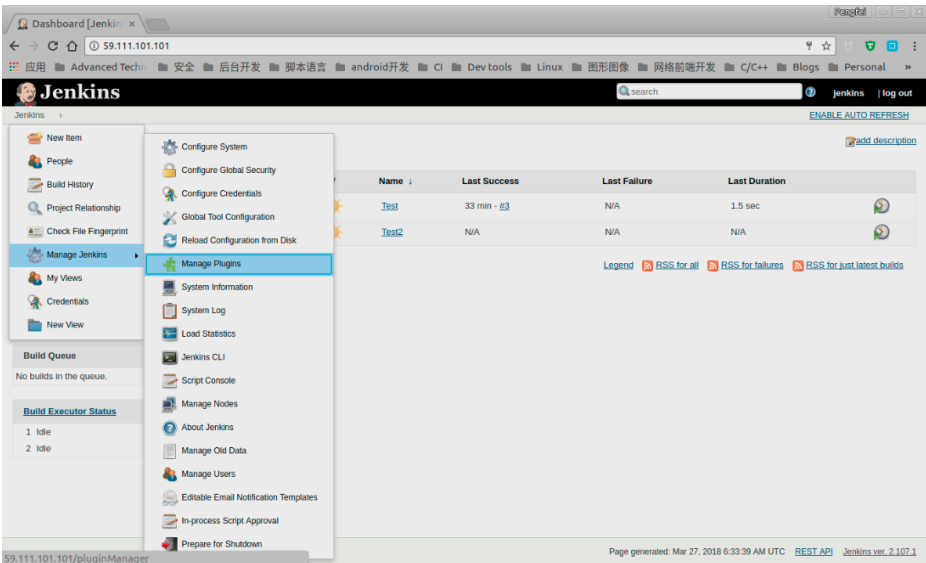
1 2018.08.17 15:03:21 字数 1,481 阅读 43,821

GitLab 是当前应用非常广泛的 Git Hosting 工具，Jenkins 是非常牛逼的持续集成工具。尽管 GitLab 有内建的 GitLab CI，但它远没有 Jenkins 那么强大好用。Jenkins 和 GitLab 在两者的结合上，都提供了非常方便的工具。在我们向 GitLab push 代码，或执行其它一些操作时，GitLab 可以将这些时间通知给 Jenkins，trigger Jenkins 工程的构建自动执行。

要实现在向 GitLab push 代码时，自动 trigger Jenkins 工程执行构建动作，需要在 GitLab 和 Jenkins 的多个地方做配置：(1)、在 Jenkins 中安装插件；(2)、配置 GitLab 用户；(3)、配置 Jenkins 服务器；(4)、配置 Jenkins 工程；(5)、配置 GitLab 工程。

在 Jenkins 中安装插件

选择 系统管理 -> 管理插件 打开插件管理也页面，如下图：



在 可选插件 中选择 Gitlab Hook Plugin 和 GitLab Plugin 等插件，然后点击下方的 直接安装 按钮安装插件：



程序员外包公司

yijian2595 关注

总资产15

微服务应用新趋势：Service Mesh、AIOps和中台化
阅读 549

JVM调优推荐
阅读 1,025

推荐阅读

11. Jenkins 触发器及视图
阅读 498

二、Jenkins Pipeline入门
阅读 566

Papercups 使用指南：如何快速自建客服平台
阅读 446

在 k8s 中安装 jenkins 并配置实现 CI/CD
阅读 456

七、Jenkins报警机制
阅读 311

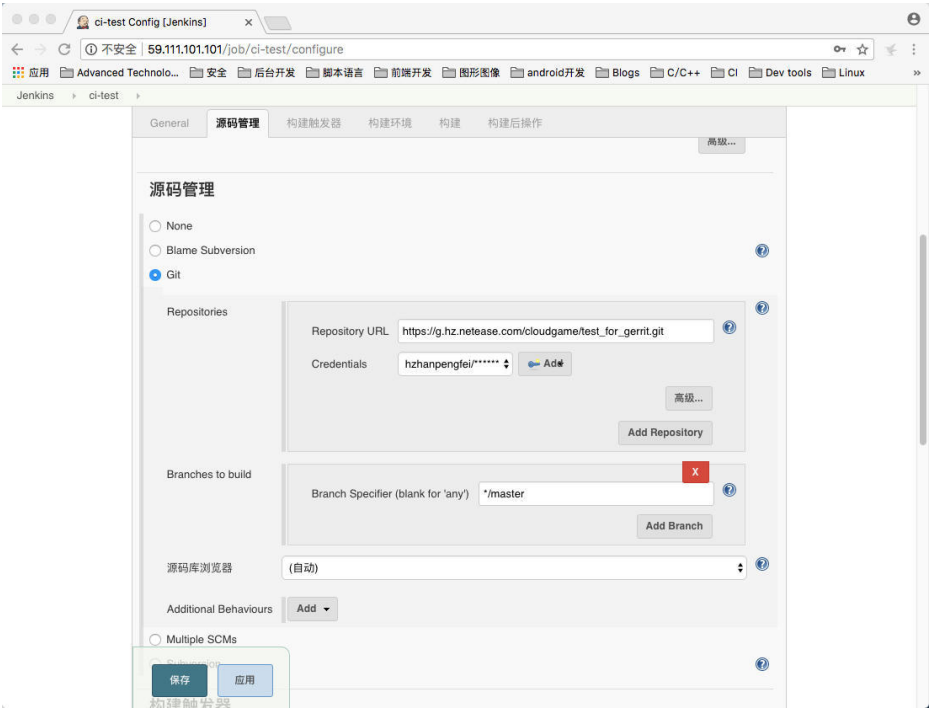


创建测试工程

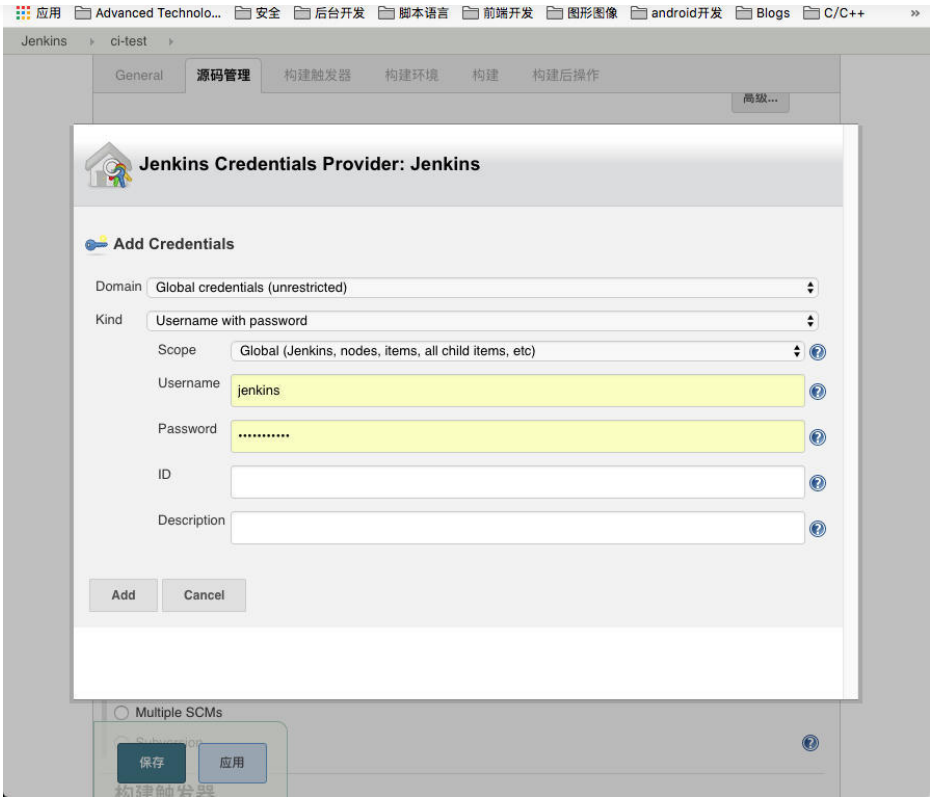
为了便于测试，这里分别先在 Jenkins 和 GitLab 上创建测试工程。在 Jenkins Dashboard 主页点击 **新建任务**，进入新建任务页面：



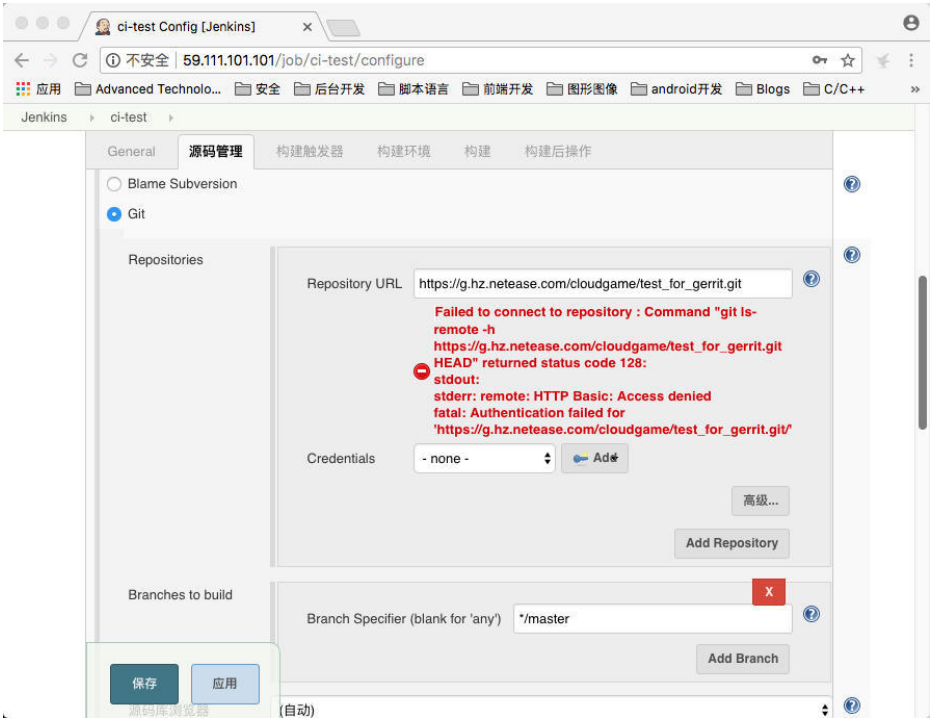
在工程配置页面的 **源码管理部分**，输入 GitLab repo 的 URL，如下图：



如果是 https 形式的 URL，记得配置登录 GitLab 的用户名和密码，通过点击 Credentials 行最后的 Add -> Jenkins 按钮，在弹出的如下对话框中输入用户名和密码：



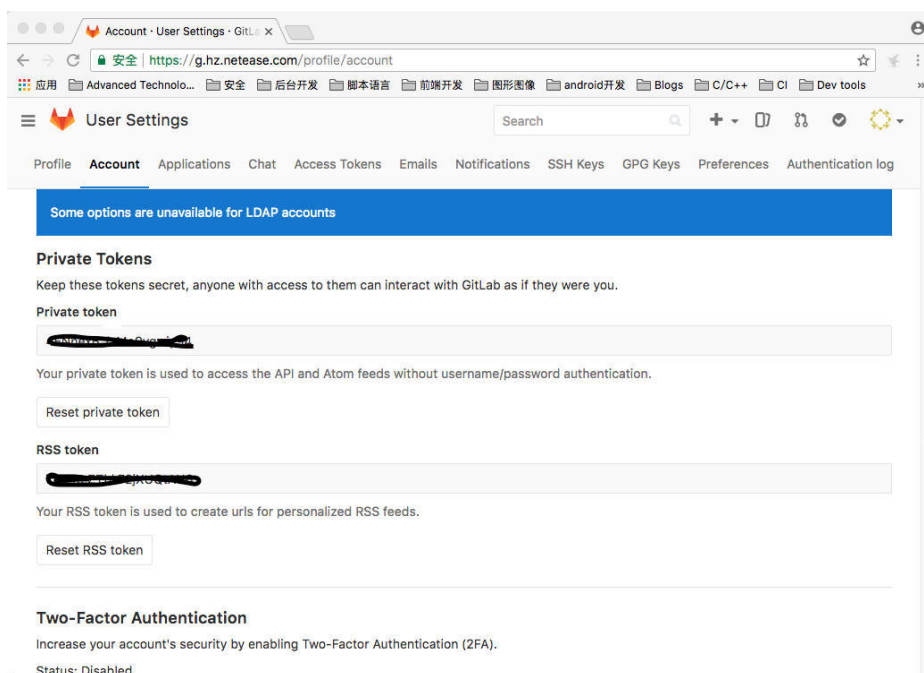
Add 之后，在 Credentials 的下拉框中选择这组用户名和密码。没有添加 GitLab 的用户密码的话，Jenkins 报错 —— 无法访问 repo，如下图：



配置 GitLab 用户

创建一个用户或选择一个已有用户，用来让 Jenkins 和 GitLab API 交互。这个用户将需要是全局的管理员或添加进每个组 / 工程，并作为成员。需要开发者权限来报告构建状态。这是由于，当使用了 'Merge when pipeline succeeds' 功能时，成功的构建状态可以触发合并。GitLab Plugin 的一些功能可能需要其它的一些权限。比如，有一个选项用于在构建成功时，接受 merge request。使用这一功能需要 developer，master 或 owner 级的权限。

选择 Settings -> Account :

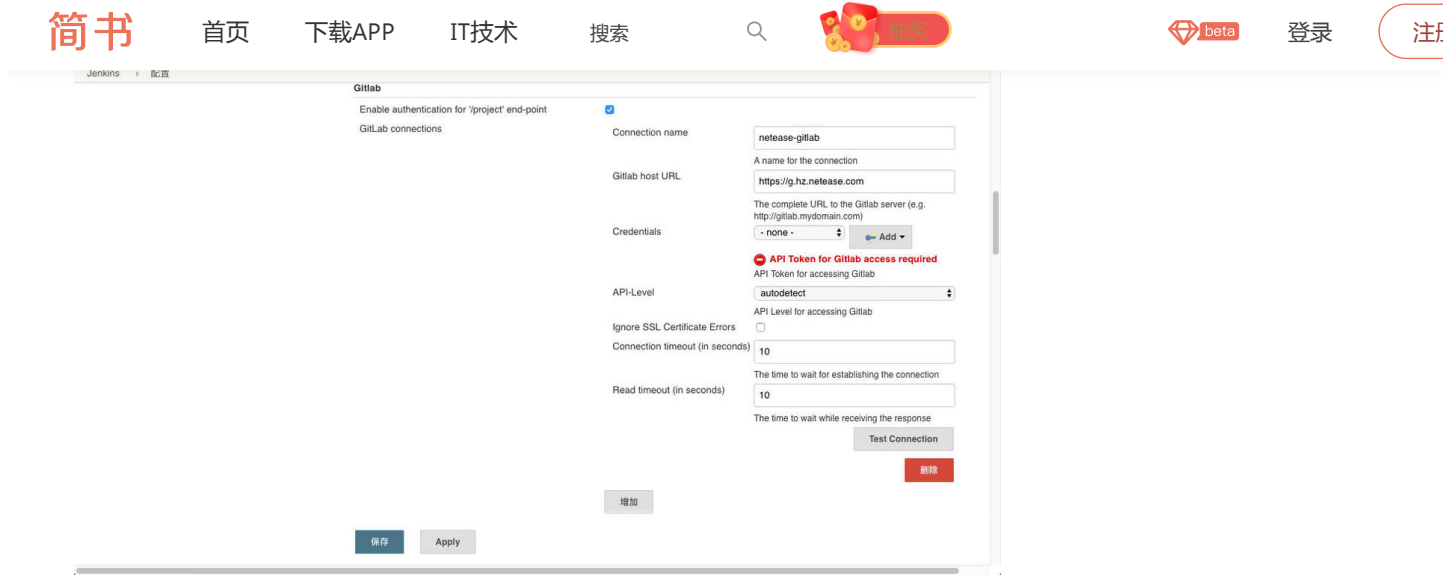


拷贝其中的 Private token，稍后在配置 Jenkins 服务器时会用到。

配置 Jenkins 服务器

需要配置 Jenkins 服务器来与 GitLab 服务器通信。

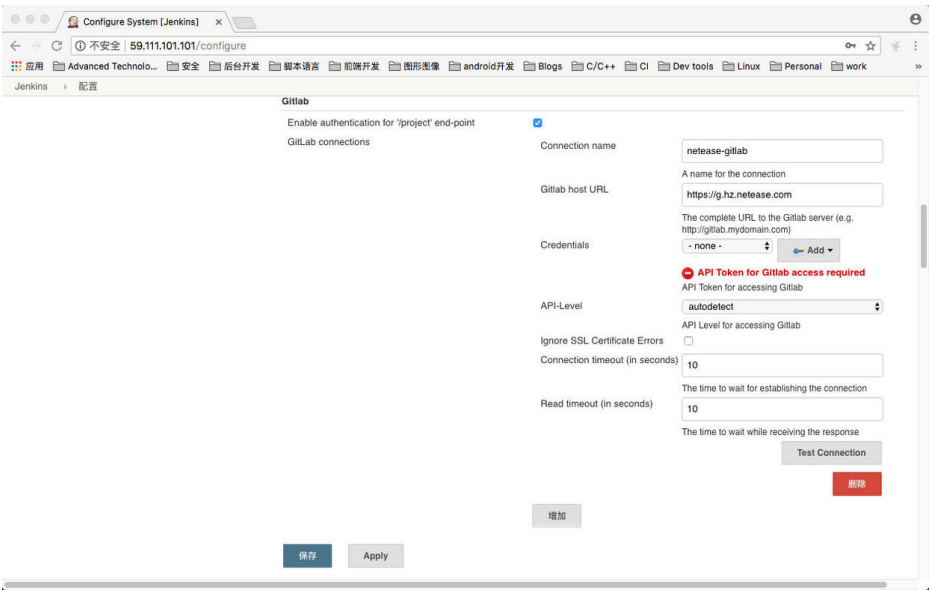
在 Jenkins 中，选择 **系统管理** -> **系统设置**，在系统设置中找到 GitLab 的部分：



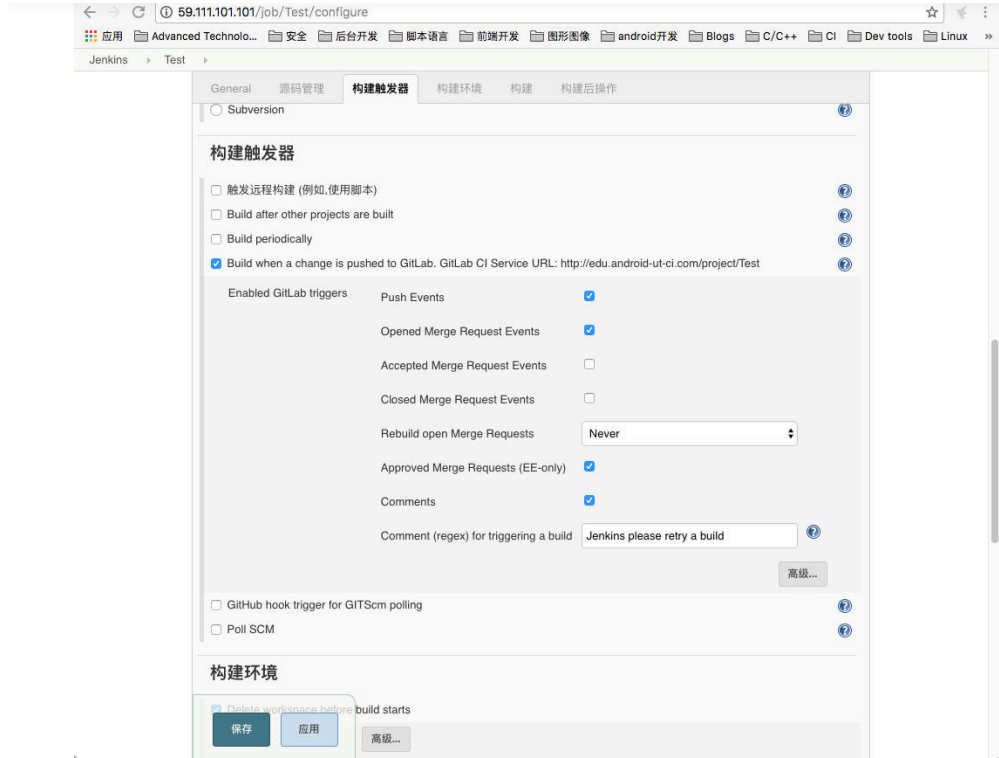
在 Connection name 后的输入框中输入连接名称，在 Gitlab host URL 后的输入框中输入 GitLab 服务器的 URL 地址。点击 Credentials 行最后面的 Add -> Jenkins 按钮，弹出如下对话框，在 Kind 后的下拉列表中选择 GitLab API token，并把上一步拷贝的 Private token 粘贴到 API token 后面的输入框中。随后在 Credentials 的下拉框中选择 GitLab API token。

配置 Jenkins 工程

打开 Jenkins 工程的配置页面，找到 **构建触发器** 的部分，勾选 Build when a change is pushed to GitLab 那一行：



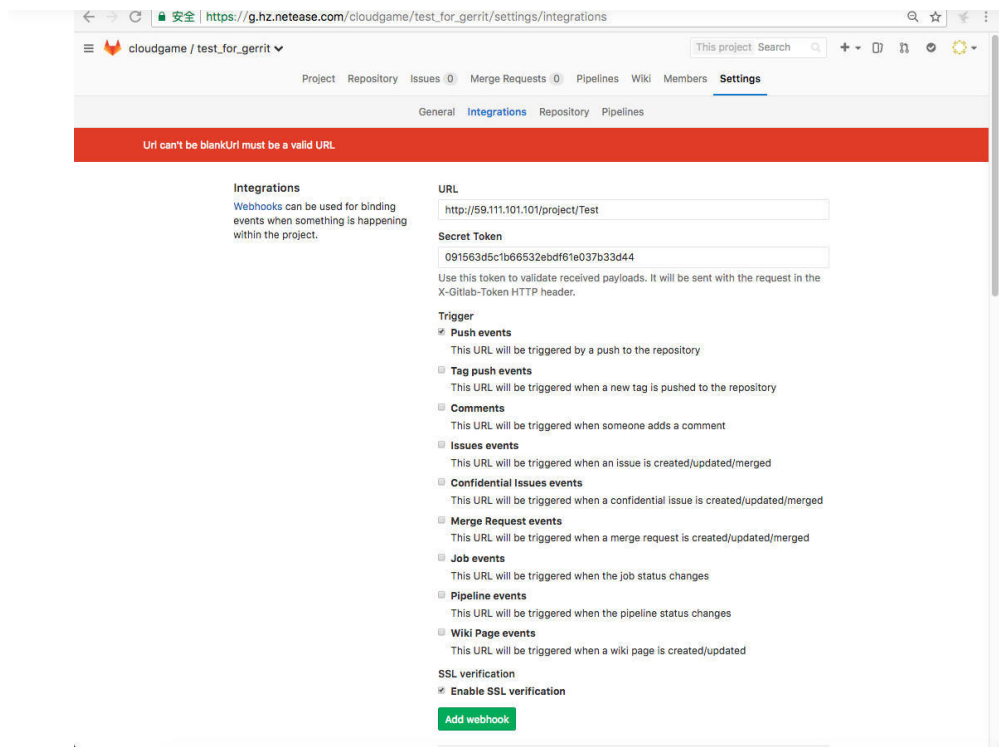
需要记下 Build when a change is pushed to GitLab 那一行中，GitLab CI Service URL: 后面的 URL，后面在配置 GitLab 工程时需要用到。



随后点击右下角的 Generate 按钮，生成 Secret token，保存这里生成的 Secret token，它同样将用于后面 GitLab 的配置。随后点击左下角的 保存 按钮，保存前面所做的配置。

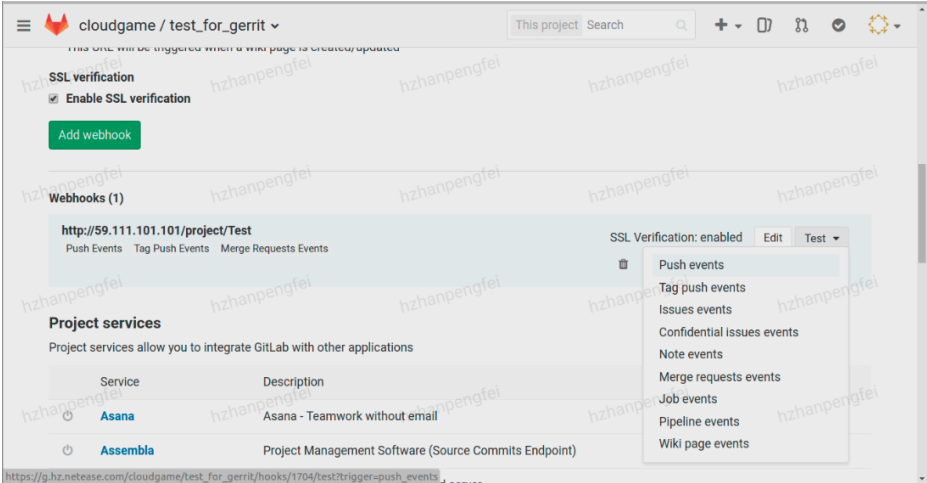
配置 GitLab 工程

创建一个新的或选择一个已有的 GitLab 工程。然后选择 Settings -> Integrations，在 URL 一栏中输入前面保存的 GitLab CI Service URL，在 Secret Token 一栏中输入前面保存的



点击绿色的 Add webhook 按钮，完成 webhook 的创建。

创建好了 webhook 之后，点击 Test 下拉框中的 Push events，如下图：



可以手动产生事件，触发 Jenkins 工程。点击 Edit，在 webhook 的编辑页面，拉到页面底部，还可以看到，该 webhook 最近的调用情况，如下图：

cloudgame / test_for_gerrit

This project Search

Save changes Test Remove

Recent Deliveries

When an event in GitLab triggers a webhook, you can use the request details to figure out if something went wrong.

Status	Trigger	URL	Elapsed time	Request time	
200	Push Hook	http://59.111.101.101/project/Test	0.02 ms	about 2 hours ago	View details
200	Push Hook	http://59.111.101.101/project/Test	0.09 ms	about 2 hours ago	View details
403	Push Hook	http://59.111.101.101/project/Test	0.01 ms	about 2 hours ago	View details
403	Push Hook	http://59.111.101.101/project/Test	0.01 ms	about 2 hours ago	View details
403	Push Hook	http://59.111.101.101/project/Test	0.02 ms	about 2 hours ago	View details
403	Push Hook	http://59.111.101.101/project/Test	0.01 ms	about 2 hours ago	View details
403	Tag Push Hook	http://59.111.101.101/project/Test	0.01 ms	about 2 hours ago	View details
200	Tag Push Hook	http://59.111.101.101/project/Test	0.01 ms	about 2 hours ago	View details

点开特定调用的 View details，还可以看到这次调用的详细情况，如下图：

Settings - cloudgame

https://ghz.netease.com/cloudgame/test_for_gerrit/hooks/1704/hook_logs/510295

cloudgame / test_for_gerrit

Project Repository Issues Merge Requests Pipelines Wiki Members Settings

General Integrations Repository Pipelines

Request details

Request URL: POST http://59.111.101.101/project/Test 200 [Resend Request](#)

Trigger: Push Hook

Elapsed time: 0.02 ms

Request time: about 2 hours ago

Request headers:

Content-Type: application/json
X-Gitlab-Event: Push Hook
X-Gitlab-Token: 091563d5c1b66532ebdf61e037b33d44

Request body:

```
{
  "object_kind": "push",
  "event_name": "push",
  "before": "e0ae206625b4e15783fb4688b70557a502719e3",
  "after": "23660c1fc1d02e125e41be46f80c8a5b23a71f7",
  "ref": "refs/heads/master",
  "checkout_sha": "23660c1fc1d02e125e41be46f80c8a5b23a71f7",
  "message": null,
  "user_id": 1335,
  "user_name": "hzhanpengfei"
}
```

由此不难理解，GitLab trigger Jenkins 工程，主要是通过向 Jenkins 服务器发送一个 POST 请求实现的。

验证测试

修改我们的 GitLab 测试工程中的文件，并 push 到 GitLab 服务器上：

```
test_for_gerrit git:(master) echo "Hello world" > README.md
```



```
(use "git add <file>..." to update what will be committed)
(use "git checkout -- <file>..." to discard changes in working directory)
modified:   re
→ test_for_gerrit git:(master) X git add .
→ test_for_gerrit git:(master) X git commit -m "Add test data."→ test_for_gerrit git:(mast

git config --global push.default matching

To squelch this message and adopt the new behavior now, use:

git config --global push.default simple

When push.default is set to 'matching', git will push local branches
to the remote branches that already exist with the same name.

Since Git 2.0, Git defaults to the more conservative 'simple'behavior, which only pushes the c
remote branch that 'git pull' uses to update the current branch.

See 'git help config' and search for 'push.default' for further information.
(the 'simple' mode was introduced in Git 1.7.11. Use the similar mode'current' instead of 'sir

Counting objects: 3, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 312 bytes | 0 bytes/s, done.
Total 3 (delta 1), reused 0 (delta 0)
To ssh://git@g.hz.netease.com:22222/cloudgame/test_for_gerrit.git 23660c1..e86152a master .
→ test_for_gerrit git:(master)
```

随后迅速地就能在 Jenkins 中，测试工程主页面的左下方，看到由 GitLab push 所 trigger 起来的构建任务：

Project Test

添加说明

禁用项目

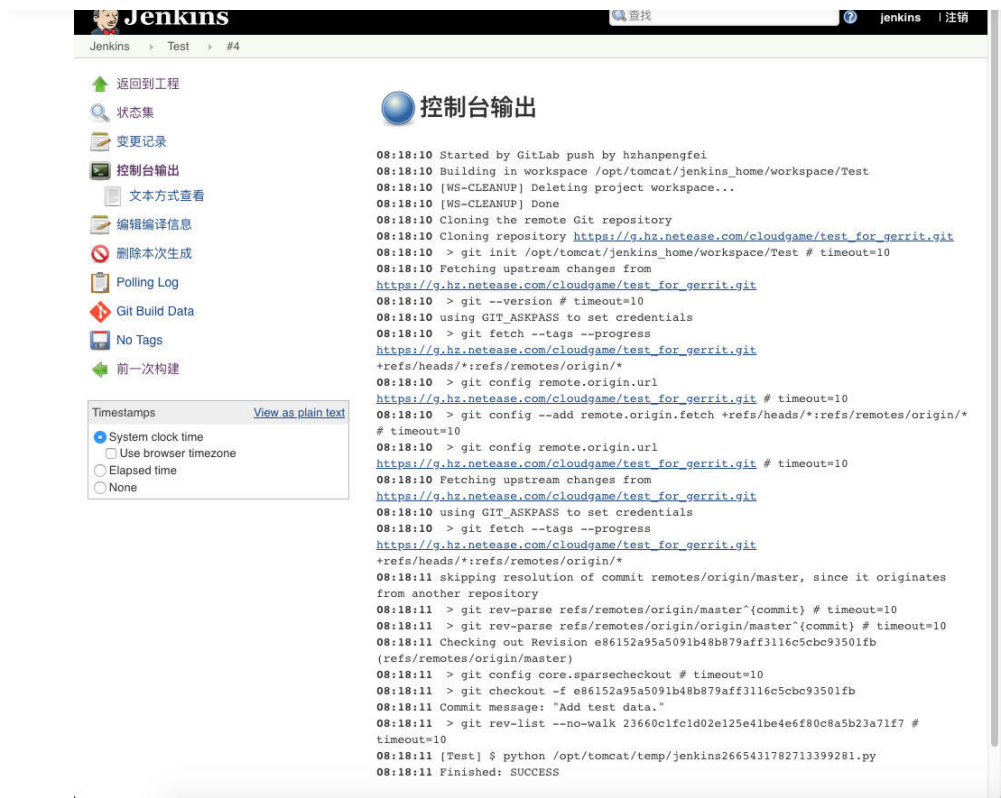
Build History 构建历史

Build	Time	Status
#4	2018-3-27 上午8:18	Started by GitLab push by hzhanpengfei
#3	2018-3-27 上午5:59	Started by GitLab push by hzhanpengfei
#2	2018-3-27 上午5:58	Started by GitLab push by hzhanpengfei
#1	2018-3-27 上午2:22	

RSS 全部 RSS 失败

生成 页面: 2018-3-27 上午08时21分53秒 REST API Jenkins ver. 2.107.1

在由 GitLab push 所 trigger 起来的构建任务的下方，会显示构建任务是由谁 push 的代码所 trigger 起来的。打开特定构建任务的 **控制台输出** 可以看到构建的详细过程：



参考文档：

Jenkins CI service Gitlab自动触发Jenkins构建打包

网易云新用户大礼包：<https://www.163yun.com/gift>

本文来自网易实践者社区，经作者韩鹏飞授权发布。

 18人点赞 >



 日记本



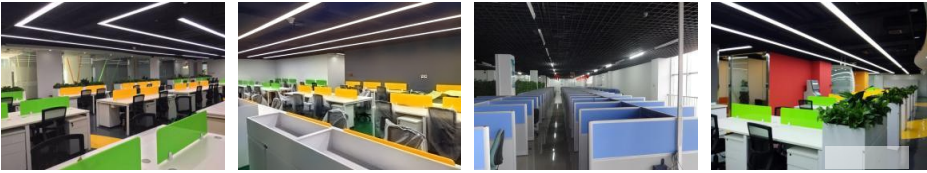
更多精彩内容，就在简书APP



"小礼物走一走，来简书关注我"

赞赏支持

工位出租600元/月



被以下专题收入，发现更多相似内容



Jenkins

推荐阅读

更多精彩内容>

GitLab 自动触发 Jenkins 构建 (2)

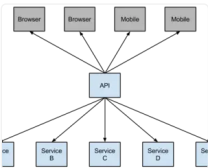
GitLab 是当前应用非常广泛的 Git Hosting 工具，Jenkins 是非常牛逼的持续集成工具。尽管 ...

yijian2595 阅读 723 评论 0 赞 2

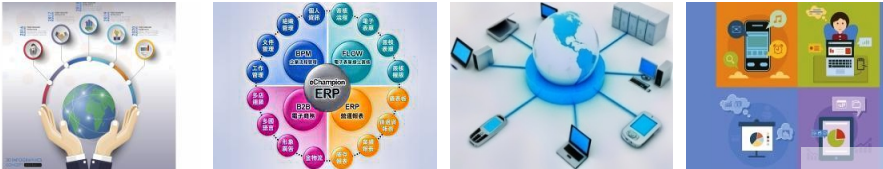
Spring Cloud

Spring Cloud为开发人员提供了快速构建分布式系统中一些常见模式的工具（例如配置管理，服务发现，断路器，智...

卡卡罗2017 阅读 121,251 评论 16 赞 134



大数据可视化5个必须知道的工具



GitLab+Jenkins+Rsync+PM2实现Node项目的持续集成与自动部署

前言 最原始的软件开发流程是，在本地搭建好环境，进行开发测试，然后去服务器上搭建环境，手动上传代码，运行测试，然后...

MageekChiu 阅读 1,047 评论 0 赞 2

设计工匠——让自己无可替代！

——《优秀到不能被忽视》一书引发的职业思考 到底是因为激情，引发了热爱和投入事业？还是因为投入？而收获了激情？一...

林青澜 阅读 402 评论 4 赞 5



android studio使用定制化android sdk

找到 中的一个classes.jar导入android studio中由于android studio配置中会使用...

傀儡世界 阅读 470 评论 0 赞 0

简书

首页

下载APP

IT技术

搜索



登录

注册

写下你的评论...

评论3

赞18