jenkins 可以和 gitlab 实现持续集成。也就是 jenkins 可以关联上 gitlab 的 project,当 gitlab 的 project,当 gitlab 的 project 有 push 请求后,会自动的触发 jenkins,执行相关操作。

gitlab 新建工程

在 gitlab 上,新建工程 test-jenkins,这个工程将来会和 jenkins 协同工作。

e or Import your project from popular Git ces	http://	/10.1.151.151/jun.le	J/	test-j	jenkins			
	Want to house several depen Import project from	dent projects unde	r the same namespa	ce? Create a gro	up			
	G GitHub	😽 GitLab.com	G Google Code	🕸 Fogbugz	🕅 Gitea	git Repo by URL	₩ GitLab export	
	Project description (optional)						
	test-jenkins							
	Visibility Level 🕑							
	Private Project access must b	e granted explicitly	/ to each user.					
	Internal The project can be clo	oned by any logge	d in user.					
	Public The project can be closed	oned without any a	uthentication.					

新建一个 README.md 文件。这个 README.md 文件需要使用 markdown 格式编写。 这里的 http 地址要记一下,因为后面的 jenkins 要使用。

test-jenkins 🖷			
test-jenkins			
☆ Star 0 HTTP ▼ http://jun.lu@sh06753pcu.spreadt ► ▲ Global ▼			
If you already have fact you can use them using commend line instructions below.			
Otherwise you can start with adding a README a LICENSE, or a .uitignore to this project.			
You will need to be owner or have the master permission level for the initial push, as the master branch is automatically protected.			
Command line instructions			
Git global setup			
git configglobal user.name "lujun" git configglobal user.email "jun.lu@spreadtrum.com"			
Create a new repository			
创建好 README.md 文件。			
The file has been successfully created.			
master v test-jenkins / README.md			
Add readme.md lujun committed less than a minute ago 64c43dba			
README.md 22 Bytes Blame History Permalink Edit Replace Delete			
this is a jenkins test			
新建一个 c 文件。			

http://www.lujun.org.cn

	Files Commits	Branches Tags Contributors Graph Compare Charts
master v test-jenkins /	+	
Name Last co	New file	Add readme.md History
README.md Add re	New directory	
README.md	P New branch New tag	
this is a jenkins test	<i>"</i> ····································	

该c文件内容:

New File

¥ master /	test_jenkins.c		
1 #include <s 2 3 int main() 4 @ 5 printf(6 return</s 	;tdio.h> ("hello jenkis\n"); 0;		
7 }			

然后创建一个 Makefile,用来将来 jenkins 执行。其实就是编译 c 文件,然后执行生成 的 a.out 可执行程序。注意,命令前面,要两个 tab。

Edit File	Preview Changes
₽ master	Makefile
1 - all:	
2	gcc test_jenkins.c && ./a.out

自此,gitlab 创建工程,以及创建文件就完成了。以上创建文件,是在 gitlab 网页上操作的,也可以使用 ssh, http 方式将该仓库通过 git 下载到本地,然后通过 push 上传文件。

Jenkins 创建任务

需要创建一个 jenkins 任务,和 gitlab 中的工程关联,实现持续集成。

http://www.lujun.org.cn

😥 Jenkins	
Jenkins → test →	
🝧 新建	jenkins
🗞 用户	All pipelint project test test-jenkins +
🥏 任务历史	
编辑视图	该视图下目前没有相关的任务。你可以添加现有的任务到该视图中或者在该该视图中创建一个新的任务。
◎ 删除视图	
🔅 系统管理	
les My Views	
Mainter Status Monitor	
Q Credentials	

选择 构建一个自由风格的软件 项目。

Enter an item name	
test-jenkins	
» Required field	,
构建一个自由风格的软件项目 这是Jenkins的主要功能 Jenkins将会结合任何SCM和任何构建系统来构建你的项目, 甚至可以构建软件	以外的系统.
构建一个maven项目 构建一个maven项目.Jenkins利用你的POM文件,这样可以大大减轻构建配置.	
Pipeline Orchestrates long-running activities that can span multiple build slaves. Suitable for building pipelines	s (formerly known as workflows) and/or organizing com

进入到配置界面。

在源码管理中,选择 git,然后输入 gitlab 工程的 http 地址。选择认证中的 Add,增加认证。

•				
ì	原码管理			
	None None			
-	9 Git			
	Repositories	Repository URL_Failed to connect to repository : Command "/usr/bin/git Is-remote -h http://jun.lu@sh06753pcu.spreadtrum.com/jun.lu/test-jenkins.git HEAD" • Future status code 128: stdout: stdout: stdout: • Teture tail: Authentication failed Credentials - none - V • Ad	0	Ø
	Branches to build	高级 Add Repository X		
		Branch Specifier (blank for 'any') "/master	0	

输入用户名和密码,也就是 gitlab 的账号和密码。

💕 Add	Credentials		
Domain	Global cred	entials (unrestricted)	•
Kind	Username v	/th password	•
	Scope	Global (Jenkins, nodes, items, all child items, etc)	0
	Username	jun.lu	Ø
	Password		0
	ID		0
	Description		0
Add	Cancel		

在红色框,选择之前创建的认证。这样, jenkins 就可以连接到 gitlab,将该工程给 clone 下来。

Git				
Repositories	Repository URL http://jun.kg/		0	0
		高级 Add Repository		
Branches to build	Branch Specifier (blank for 'any') */master	X	0	

在 构建触发器 选择中,选择当 gitlab 因为 push 变化时,自动 build。把红色部分的 URL 地址记录下来。

构建触发器			
触发远程49編(940.使用脚本)			•
Build after other projects are) built		0
Build periodically			Ø
Build when a change is push	ned to GitLab. GitLab CI Service U	RL: http://10.1.151.151:8082/project/lest-jenkins	Ø
Enabled GitLab triggers	Push Events	8	
	Merge Request Events	8	
	Rebuild open Merge Requests	Never	
	Comments	8	
	Comment for triggering a build	Jenkins please retry a build	•
	Merge Request Events Rebuild open Merge Requests Comments Comment for triggering a build	Never Jenkins please retry a build	Ø

在 gitlab 工程的 settings 下的 integrations 选项下,将 jenkins 中的上述 URL 填入到 URL 中。这个当 gitlab 工程因为 push 发生变化时,会通过这个 URL 通知 jenkins,工程发生变化,从而触发 jenkins 的相关操作。

	Project Repository Issues 0 Merge Requests 0 Pipelines Wiki Settings
	General Members Integrations Repository CI/CD Pipelines Pages
Integrations	URL
Webhooks can be used for binding events	http://10.1.151.151:8082/project/test-jenkins
project.	Secret Token
	Has this taken to validate received poulands It will be cant with the request in the V Citlah Taken HTTD bander

在 构建 选项,选择 execute shell,执行的命令就是 make。

Execute shell			
Command	nake		
	See the list of available environment variables		

这样,一个 jenkins 任务就创建好了。

Jenkins > test >	test-jenkins >				
摿 返回面板	Project test jonking				
🔍 状态	Project test-jenkins				
🗾 修改记录					
📄 工作空间					
🔊 立即构建					
🚫 删除 Project					
🐡 配置	量新修改记录				
🖤 Embeddable Build S	tatus				
	相关连接				
Build History					
点击 立即构建	, 检查 jenkins 和 gitlab 的连接情况。会出现 build 结果。				
💧 返回面板	Project test jonking				
🔍 状态	Project lest-jenkins				
🗾 修改记录					
📄 工作空间					
シ 立即构建					
🚫 删除 Project	(0000000)				
🔅 配置	● 量新修改记录				
Embeddable Build State					
Build History					
find	· Last stable build(#1).8.1 秒之前				
11110 0 #1 2017 / 0 下ケイ・・・	▲ Last successful build(#1).8.1 秒之前 ▲ Last completed build(#1).8.1 秒之前				
<u>₩</u> 2017-4-8 N+1.4					
5 R	ISS 全部 💦 RSS 失败				

查看该 build 结果的终端输出,执行了 make 命令,编译 c 代码,并执行 a.out。因为 a.out 返回值是 0,所以结果是成功。

对于 jenkins,如果执行的 shell 命令,最终是返回非 0,那就认为是失败,如果返回 0,认为是成功。

摿 返回到工程			
🔍 状态集	● 控制台输出		
🔁 变更记录	-		
Console Output	Started by user <u>ium lu</u> Building in workspace /war/lib/jenkins/jobs/test-jenkins/workspace Cloning repository <u>http://ium.lu@sh/6755pcu.spreadtrum.com/ium.lu/test-jenkins.rit</u>		
🧾 编辑编译信息	>/usr/bin/rit init /var/lib/jenkins/jobs/test-jenkins/workspace # timeout=10		
🚫 删除本次生成	Fetching upstream changes from http://jun.lugendo/bbjpcu.spreadtrum.com/jun.lu/test_tenkins.git >/uur/bin/git=-wersion # timeout=10		
🚸 Git Build Data	using GIT_ASERASS to set credentials		
No Tags	>/usr/bin/git fetch — tags — progress http://um.lwghoDGTSpcu.spreadfrum.com/um.lu/testienkins.git Trefs/heads/*:refs/remotes/origin/* >/usr/bin/git config remote.origin.url http://um.lwghofSpcpu.spreadfrum.com/um.lu/testienkins.git # timeout=10		
Ima No rago	<pre>/ /urr/in/git config renote.orgin.url http://un.lub/shf05/50cu.goreat/uw.con/un.lu/rest-impline.ril # timeout=10 / /urr/in/git config renote.orgin.fetch trefs/heads/ric/fs/fs/enotes/orgin/# # timeout=10 / /urr/in/git config renote.orgin.fetch trefs/heads/ric/fs/fs/enotes/orgin/# # timeout=10 / /urr/in/git config renote.orgin.fetch trefs/heads/ric/fs/fs/enotes/orgin/un.lu/rest-impline.git / /urr/in/git fetchtagsprogress http://un.lub/hof5550cu.goreadtrum.com/un.lu/rest-impline.git / /urr/in/git rempare ofs/fsmotes/orgin/master [comail # timeout=10 / /urr/in/git config one.gorescheduut # timeout=10 / /urr/in/git config one.gorescheduut</pre>		

在工作空间, gitlab 工程的源代码,也 clone 了下来,并且生成了 a.out 文件。

🛧 返回面板	Workspace of test jonking on master		
Q. 状态	workspace of test-jenkins of fi	laster	
🗾 修改记录	▶		
📄 工作空间			
🌇 清理工作空间	国 <u>a.out</u> 8.35 KB <u>查看</u>		
▲ 立即构建	I Makefile 37 B <u>青</u>		
	I README.md 22 B <u>查看</u>		
🚫 删除 Project	interst_jenkins.c 78 B <u>音看</u>		
🔅 配置	<u>〇(打包下载全部文件)</u>		
Embeddable Build Status			

当确认 jenkins 与 gitlab 成功关联后,后面就是相当奇妙的持续集成了。 修改 gitlab 工程的 c 代码。

Edit File Preview Changes	
<pre> master test_jenkins.c </pre>	
<pre>1 #include <stdio.h> 2 3 int main() 4 ~ {</stdio.h></pre>	
<pre>5 printf("hello jenkis, this will failed n");</pre>	
6 return 1; 7 }	

当 commit 该文件后,也就是向版本库中 push 了更新, jenkins 这边,就会自动检测到该 更新,触发一个 build 操作。这一切,都是自动完成的,不需要我们干预。



最终,这个构建操作,是失败的,查看终端打印信息,可以查看详细信息。

http://www.lujun.org.cn



这样,就实现了 gitlab 和 jenkins 的持续集成。每当 gitlba 版本库,有更新时,就会自动 触发 jenkins 的构建任务。利用这个持续集成,我们可以做很多自动化工作。

Jenkins 还可以设置,构建任务完成后,再执行一些操作。这些操作统称为构建后操作。 在任务的设置界面中,可以进行设置。比如通过 ssh 登录到远程主机,做一些操作,或者发 送邮件等等。

增加构建步骤	•		
构建后操作	۲ F		
增加构建后操作	時一日 一日 一	_	
保存	Apply		