**Jenkins基础**

Jenkins简介

Jenkins概述

Jenkins是由java编写的一款开源软件

作为一款非常流行的CI（持续集成）工作，用于构建和测试各种项目

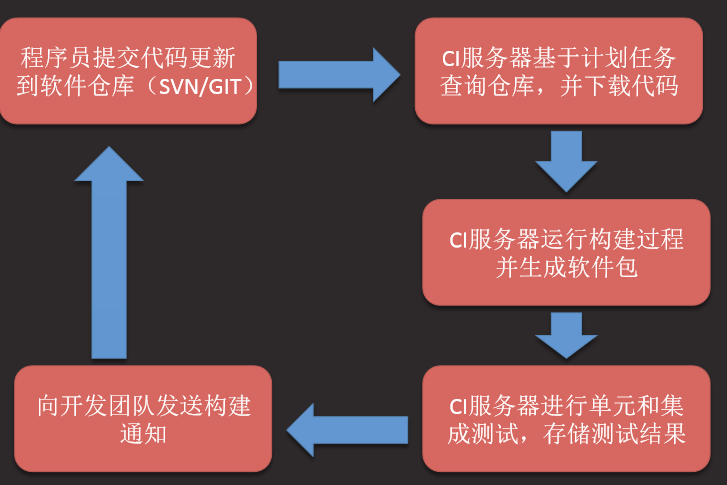
Jenkins的主要功能是监视重复工作的执行，例如软件工程的构建或在cron下设置的jobs

持续集成

持续集成（CI）是当下最为流行的应用程序开发实践方式

程序员在代码仓库中集成了修复bug、新特性开发或是功能革新

CI工具通过自动构建和自动测试来验证结果。这可以检测到当前程序代码的问题，迅速提供反馈



jenkins特点

简单、可扩展、用户界面好

支持各种SCM（软件配置管理）工具，如SVN、GIT、CVS等

能够构建各种风格的项目

可以选择安装多种插件

跨平台，几乎可以支持所有的平台

安装jenkins

下载jenkins



安装jenkins

rpm -ivh jenkins-2.121-1.1.noarch.rpm

启动服务

systemctl start jenkins

systemctl enable jenkins

初始化jenkins

jenkins默认运行在8080端口



安装插件

如果网速较快，选择推荐插件，否则选择自定义，只选中GIT即可



管理用户

可以直接使用admin登录



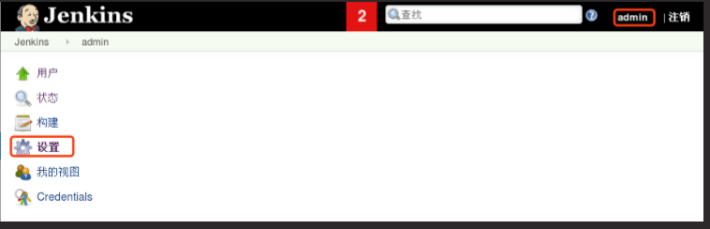
完成安装

安装成功如下图所示：



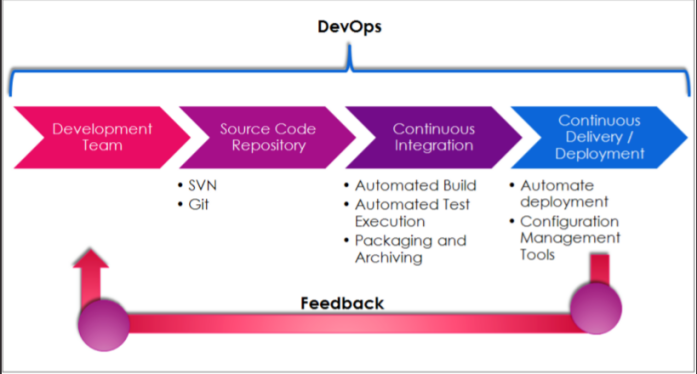
修改管理员密码

修改管理员密码：admin->设置



CI/CD流程

程序开发的生命周期内，需要程序员和运维共同协作



准备git仓库

本地仓库

初始化wordpress

解压wordpress4.8版本

unzip wordpress-4.8-zh\_CN.zip

初始化git仓库

cd wordpress

git init

git add .

git commit -m “wordpress init”

git status

tag标签

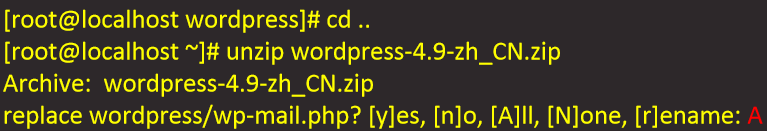
如果达到一个重要的阶段，并希望永远记住那个特别的提交快照，可以使用git tag给它打上标签

将初始化完毕的wordpress打标签1.0

git tag v1.0

升级wordpress

将wordpress新版本解压到项目中



更新git仓库

检查git状态

cd wordpress/

git status

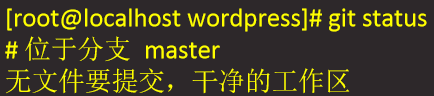
将更新/增加的文件确认至仓库

git add .

git commit -m “upgrade to new version”

添加标签

确认所有项目已提交



为升级后的wordpress项目增加v2.0标签

git tag v2.0

管理标签

查看所有标签git tag

切换到v1.0标签git checkout v1.0

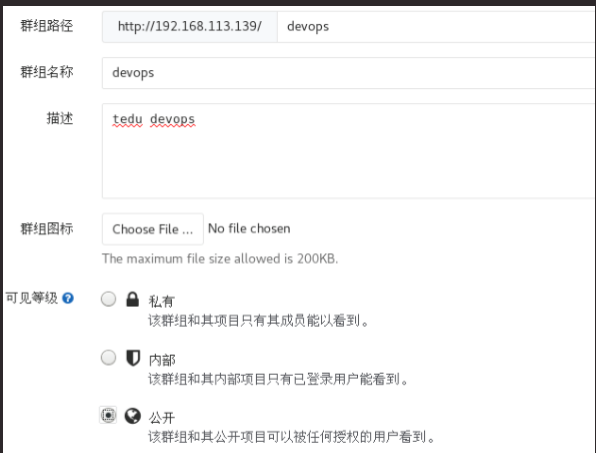
切换到最新状态git checkout master

远程仓库

创建群组

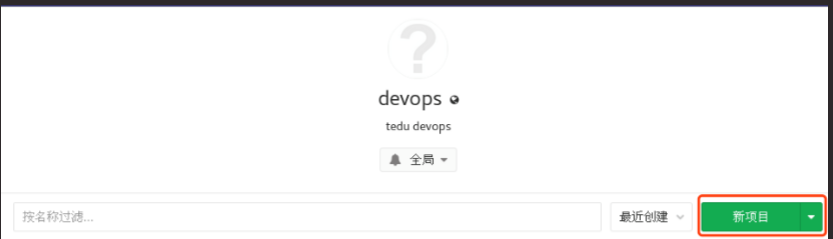
创建名为devops的群组





创建wordpress项目

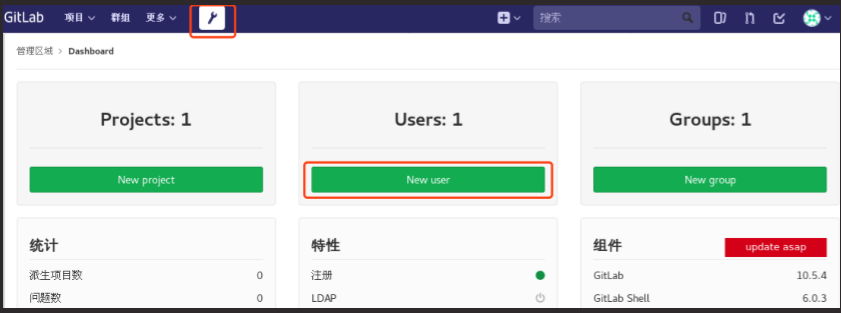
创建名为wordpress的项目

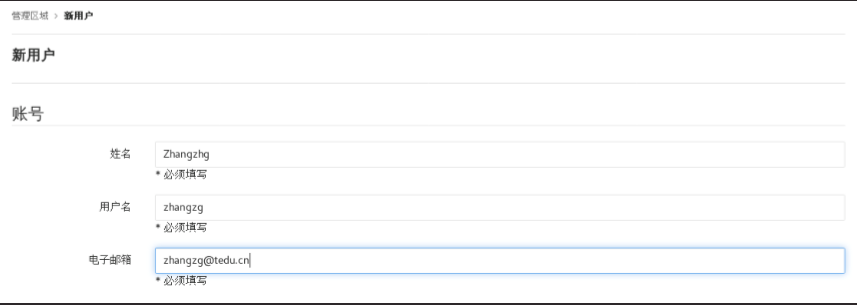


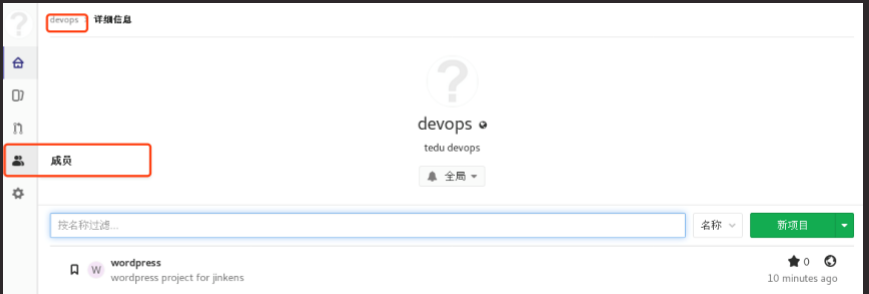


创建用户

为wrodpress项目创建主程序员用户

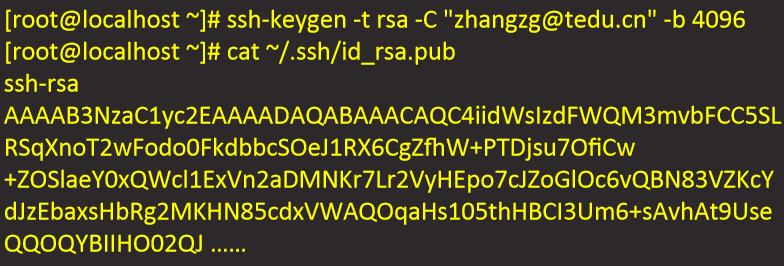




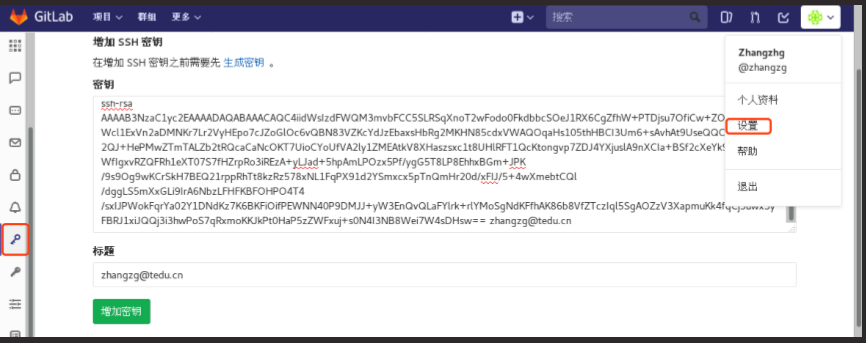




用户生成ssh密钥

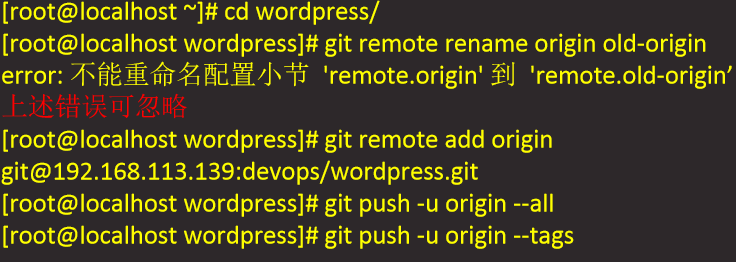


新用户登录重置密码，然后设置ssh密钥



上传wordpress代码

因为本地wordpress已经是git版本库了，所以采用以下方式进行上传



在网页中查看项目

所有tag可以通过http://192.168.113.139/devops/wordpress/tags访问

**应用jenkins**

构建项目

下载git插件

为了使得jenkins可以使用git的tag，需要下载git parameter插件

点击“系统管理”->“管理插件”

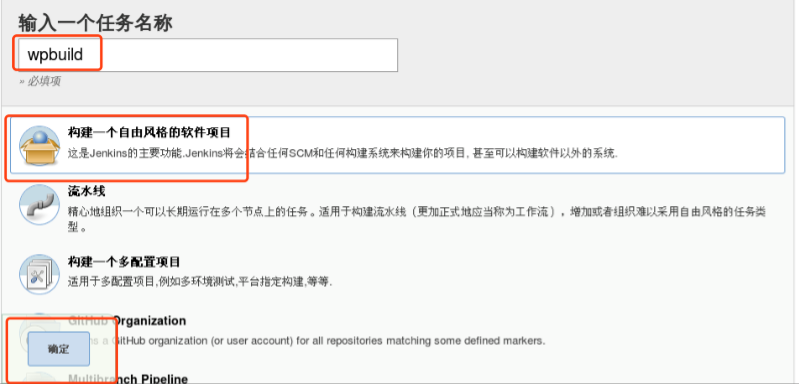


创建自由风格项目

新建项目

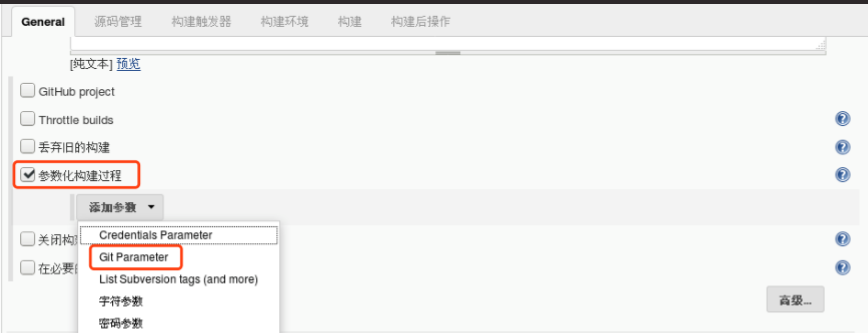


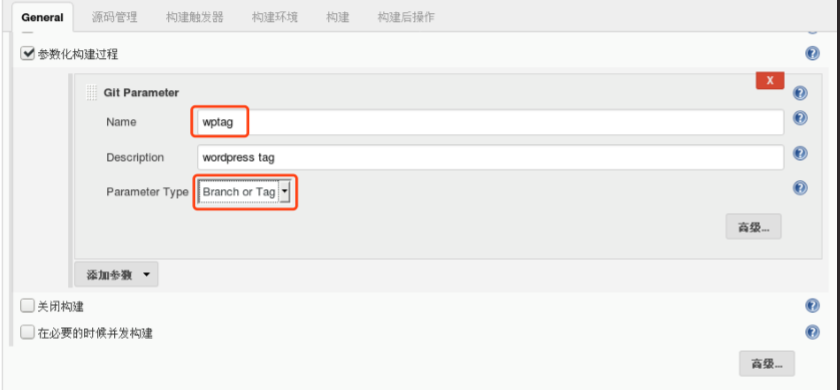
选择自由风格



设置参数

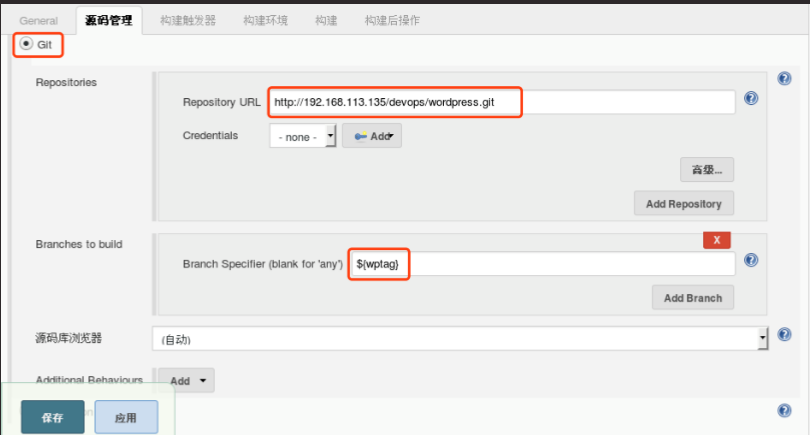
添加git parameter参数



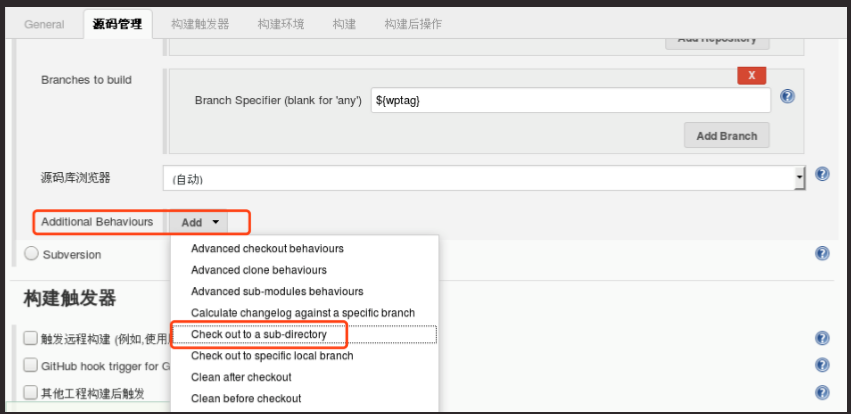


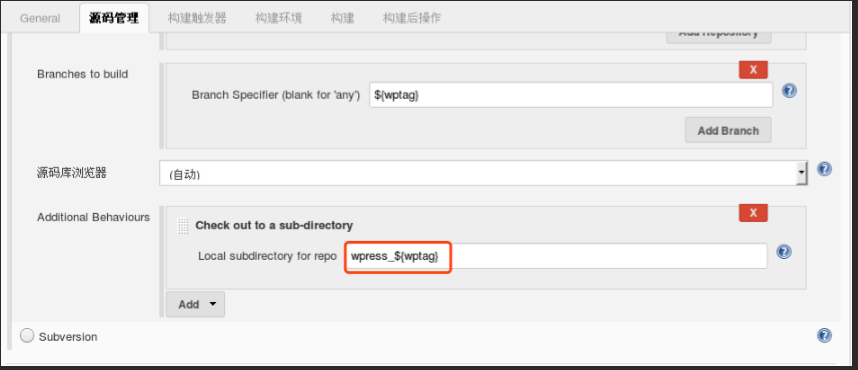
源码管理

源码采用git



将源码checkout到子目录





构建工程

构建工程





检验结果

选择指定的标签



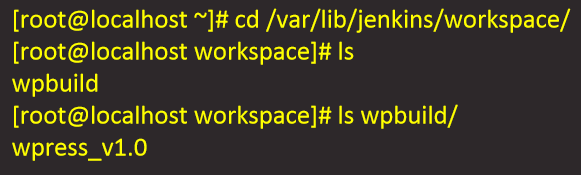
查看日志输出





查看本地结果

构建好的项目出现在/var/lib/jinkens目录下



分发服务器管理

优化构建工程

在jenkins服务器上安装apache，用于分发应用程序

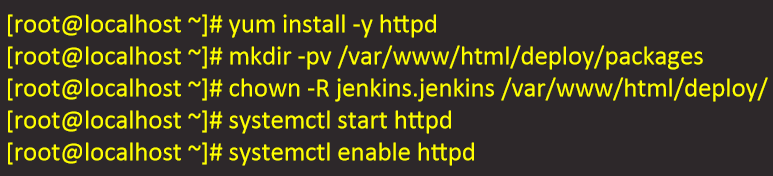
为了方便应用服务器下载，jenkins构建的工程应该打包为一个文件

为了应用程序可以获知下载的程序文件是没有损坏的，应该为其生成md5值

配置分发服务器

通过web服务为应用服务器提供应用程序

下载目录为/var/www/deploy/packages



修改工程构建工程

为下载的应用打包，以及生成md5可以能过在工程中增加构建步骤完成





构建测试

执行构建工程



检查分发服务器的相关目录



创建版本文件

创建两个版本文件

live\_version：表示当前使用版本

last\_version：表示上一个版本

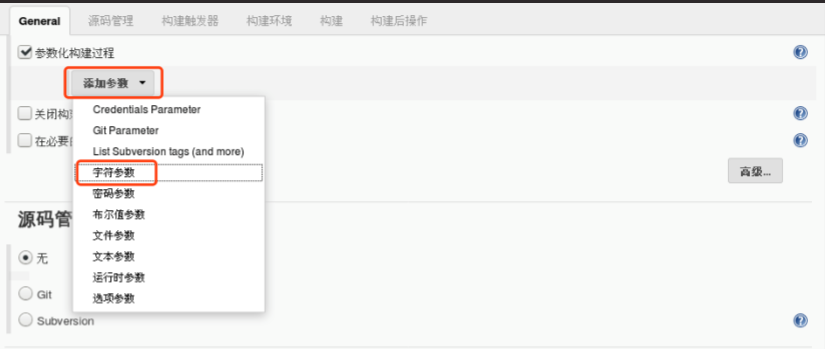
应用服务器可以查看live\_version决定是不是要发布新版本

如果新版本有问题，应用服务器可以根据last\_version回滚到前一版本

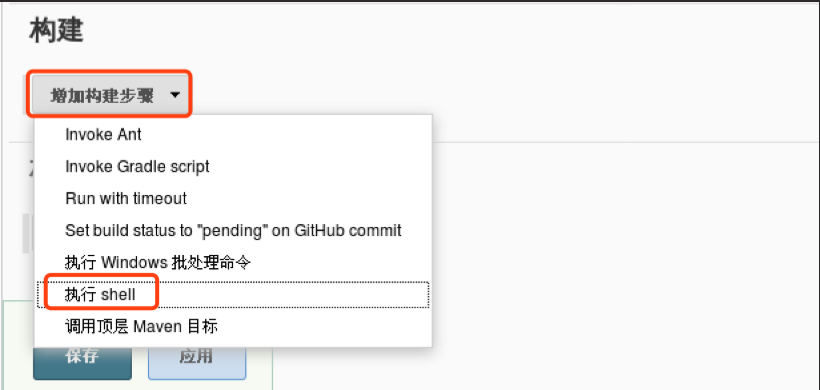
创建live\_version文件

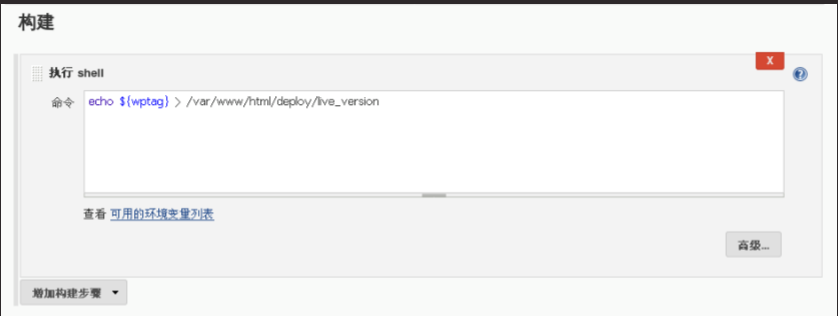
为了或所正确的版本，通过jenkins工程实现











执行构建版本工程

创建主机操作与获取信息操作完全一样，只是传递的请求参数不一样而已



查看构建结果



**管理应用服务器**

自动化部署

服务器规划

为了方便版本的切换，可以规划如下目录

/var/www/downliad用于存储下载的应用

/var/www/deploy用于存储解压的应用

创建/var/www/html/current软连接，指向需要部署的应用版本

下载应用

编写下载应用的功能代码

通过位置参数指定要下载的版本

位置参数是live下载当前版本

位置参数是last下载前一个版本

如果已经下载，则不要重复下载

检验文件

编写校验文件代码

计算指定文件的md5值

将md5值与发布服务器提供的md5值进行比较，以确认下载的文件无误

发布应用

编写应用发布代码

根据指定的版本，创建/var/www/html/current链接，指向到不同的发布版本