

DEVOPS DAY05



Python开发

NSD DEVOPS DAY05

内容

上午	09:00 ~ 09:30	作业讲解和回顾
	09:30 ~ 10:20	Jenkins基础
	10:30 ~ 11:20	
	11:30 ~ 12:00	准备git仓库
下午	14:00 ~ 14:50	应用jenkins
	15:00 ~ 15:50	
	16:10 ~ 17:00	管理应用服务器
	17:10 ~ 18:00	总结和答疑



Jenkins基础

Jenkins基础

Jenkins简介

Jenkins概述

持续集成

Jenkins特点

安装Jenkins

下载Jenkins

安装Jenkins

初始化Jenkins

安装插件

管理用户

完成安装

修改管理员密码

CI/CD流程

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

知识讲解



持续集成

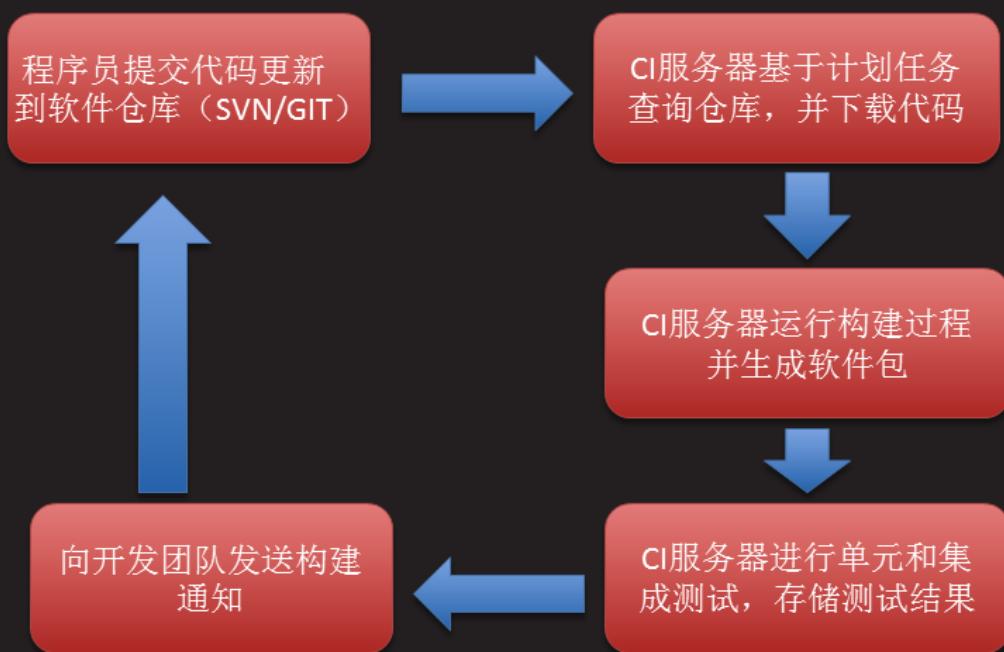
知识讲解

- 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



持续集成（续1）

知识讲解



Jenkins特点

知识讲解

- 简单、可扩展、用户界面友好
- 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

The screenshot shows the Jenkins website at jenkins.io. At the top, there's a navigation bar with links to Blog, Documentation, Plugins, Use-cases, Participate, Sub-projects, Resources, About, and a prominent Download button. Below the navigation is a large cartoon character of a man with a white beard, wearing a blue suit and red bow tie, holding a white mug. To the right of the character, the word "Jenkins" is written in a large, bold, serif font. Below it, the tagline "Build great things at any scale" is displayed. A paragraph of text explains that Jenkins is an open-source automation server with many plugins for building, deploying, and automating projects. At the bottom of the main content area, there are two buttons: "Documentation" (in a light blue box) and "Download" (in a red box). A blue banner at the bottom left says "Say ‘hello’ to Blue Ocean" and describes it as a new user experience. On the right side of the banner, there's a small screenshot of a Jenkins pipeline interface.

知识讲解



安装Jenkins

- 安装Jenkins

```
[root@localhost 下载]# rpm -ivh jenkins-2.121-1.1.noarch.rpm
```

警告: jenkins-2.121-1.1.noarch.rpm: 头V4 DSA/SHA1 Signature, 密钥 ID d50582e6: NOKEY

准备中... # ##### [100%]

正在升级/安装...

```
1:jenkins-2.121-1.1 ##### [100%]
```

- 启动服务

```
[root@localhost 下载]# systemctl start jenkins
```

```
[root@localhost 下载]# systemctl enable jenkins
```

知识讲解



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登陆

新手入门

创建第一个管理员用户

用户名:

密码:

确认密码:

全名:

电子邮件地址:

Jenkins 2.121

[使用admin账户继续](#) [保存并完成](#)

知识讲解



完成安装

- 安装成功如下图所示：

新手入门

Jenkins已就绪！

你已跳过创建admin用户的步骤。要登录请使用用户名：'admin' 及用于访问安装向导的管理员密码。

Jenkins安装已完成。

[开始使用Jenkins](#)

Jenkins 2.121

知识讲解



修改管理员密码

- 修改管理员密码：admin->设置

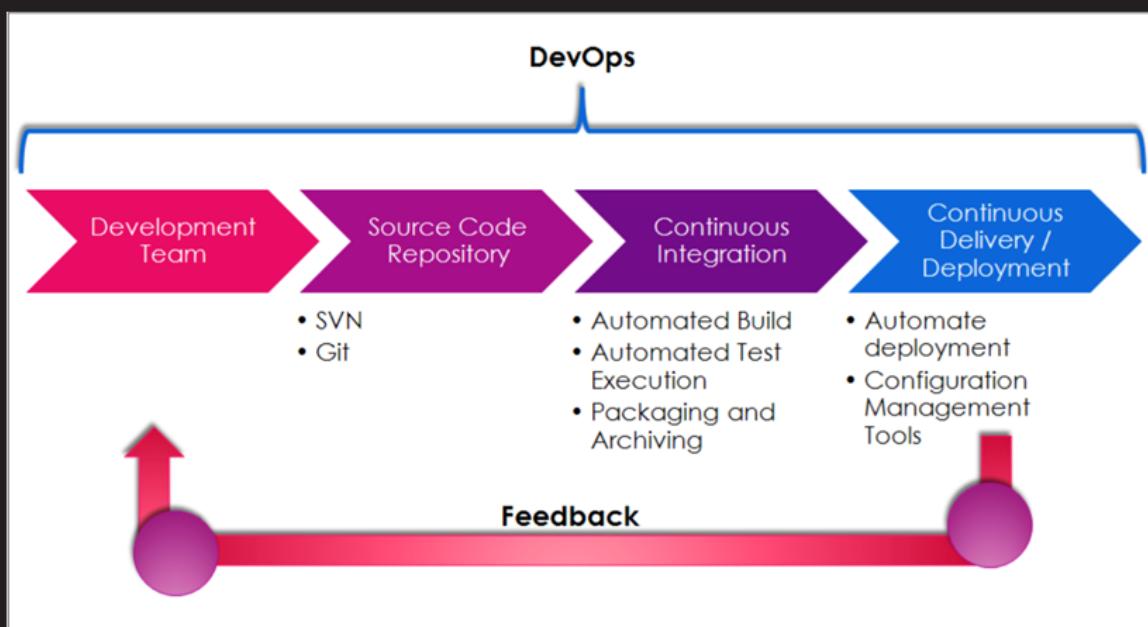


知识讲解



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



知识讲解



案例1：安装Jenkins

课堂练习

1. 运行虚拟机，将第一块网络的连接方式改为NAT
2. 安装Jenkins
3. 初始化Jenkins



准备git仓库

准备git仓库

本地仓库

初始化wordpress项目

tag标签

升级wordpress

更新git仓库

添加标签

管理标签

远程仓库

创建群组

创建wordpress项目

创建用户

上传wordpress代码

在网页中查看项目

本地仓库

初始化wordpress项目

- 解压wordpress 4.8版本

```
[root@localhost ~]# unzip wordpress-4.8-zh_CN.zip
```

- 初始化git仓库

```
[root@localhost ~]# cd wordpress/  
[root@localhost wordpress]# git init  
[root@localhost wordpress]# git add .  
[root@localhost wordpress]# git commit -m "wordpress init"  
[root@localhost wordpress]# git status  
# 位于分支 master  
无文件要提交，干净的工作区
```

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v1.0

```
[root@localhost wordpress]# git tag v1.0
```

知识讲解



升级wordpress

- 将wordpress新版本解压到项目中

```
[root@localhost wordpress]# cd ..  
[root@localhost ~]# unzip wordpress-4.9-zh_CN.zip  
Archive: wordpress-4.9-zh_CN.zip  
replace wordpress/wp-mail.php? [y]es, [n]o, [A]ll, [N]one, [r]ename: A
```

知识讲解



更新git仓库

知识讲解

- 检查git状态

```
[root@localhost ~]# cd wordpress/  
[root@localhost wordpress]# git status
```

- 将更新/增加的文件确认至仓库

```
[root@localhost wordpress]# git add .  
[root@localhost wordpress]# git commit -m "upgrade to new version"
```



添加标签

知识讲解

- 确认所有项目已提交

```
[root@localhost wordpress]# git status  
# 位于分支 master  
无文件要提交，干净的工作区
```

- 为升级后的wordpress项目增加v2.0标签

```
[root@localhost wordpress]# git tag v2.0
```



管理标签

知识讲解

- 查看所有标签

```
[root@localhost wordpress]# git tag  
v1.0  
v2.0
```

- 切换至v1.0标签

```
[root@localhost wordpress]# git checkout v1.0  
Note: checking out 'v1.0'.
```

- 切换到最新状态

```
[root@localhost wordpress]# git checkout master  
之前的 HEAD 位置是 50984b1... wordpress init  
切换到分支 'master'
```



课堂练习

案例2：设置本地仓库

1. 解压wordpress4.8
2. 将解压目录配置为git仓库
3. 为当前代码标记为v1.0
4. 更新wordpress到4.9
5. 将代码标记为v2.0



远程仓库

创建群组

- 创建名为devops的群组

知识讲解



创建群组（续1）

- 创建名为devops的群组

知识讲解

群组路径: http://192.168.113.139/ devops
群组名称: devops
描述: tedu devops
群组图标: Choose File ... No file chosen
The maximum file size allowed is 200KB.
可见等级: 私有
该群组和其项目只有其成员能以看到。
 内部
该群组和其内部项目只有已登录用户能看到。
 公开
该群组和其公开项目可以被任何授权的用户看到。

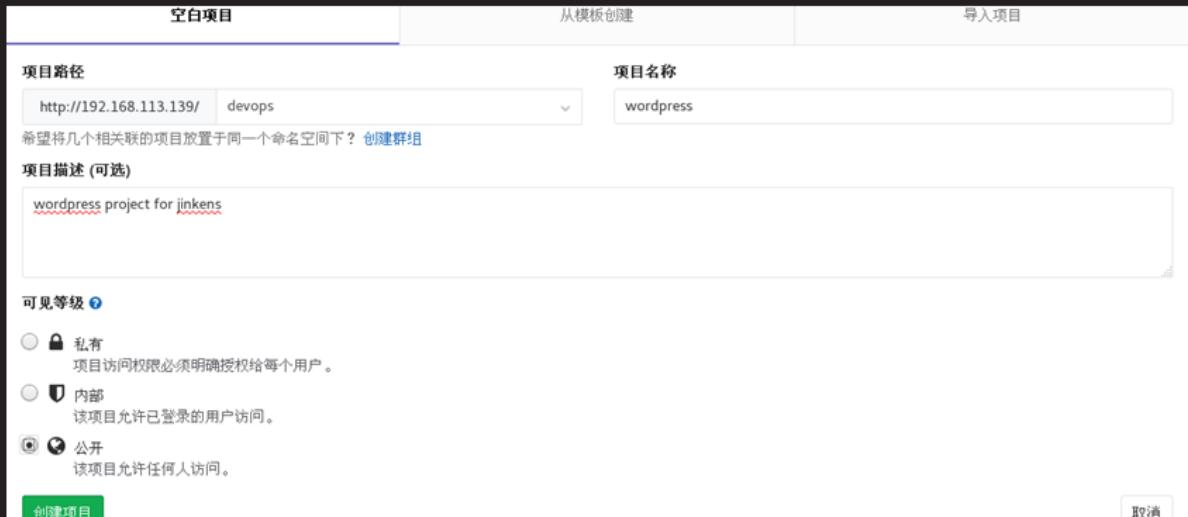
创建wordpress项目

- 创建名为wordpress的项目



创建wordpress项目（续1）

- 创建名为wordpress的项目



空白项目 | 从模板创建 | 导入项目

项目路径: http://192.168.113.139/ devops
希望将几个相关联的项目放置于同一个命名空间下？[创建群组](#)

项目名称: wordpress

项目描述 (可选): wordpress project for jinkens

可见等级:

- 私有 项目访问权限必须明确授权给每个用户。
- 内部 该项目允许已登录的用户访问。
- 公开 该项目允许任何人访问。

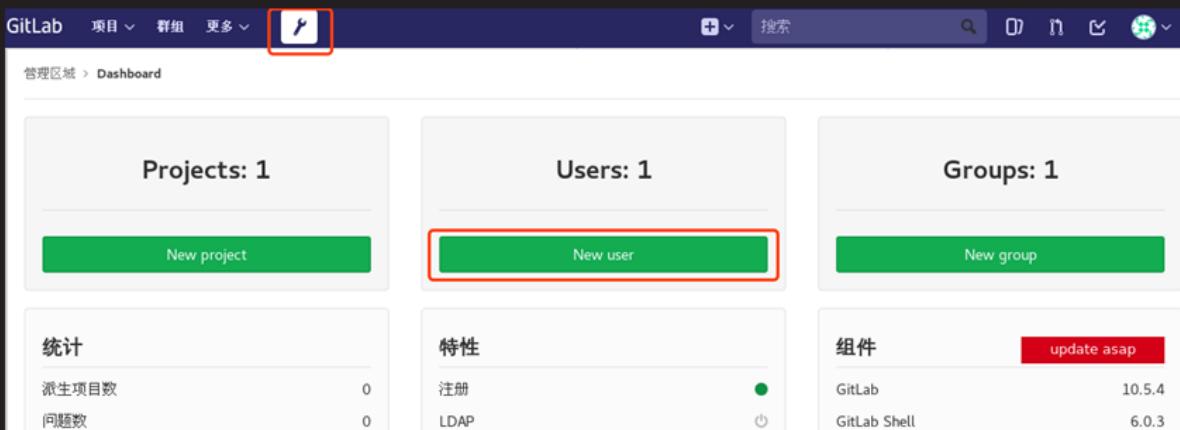
[创建项目](#) [取消](#)

知识讲解



创建用户

- 为wrodrress项目创建主程序员用户



GitLab 项目 群组 更多

管理区域 > Dashboard

Projects: 1 [New project](#)

Users: 1 [New user](#)

Groups: 1 [New group](#)

统计

派生项目数	0
问题数	0

特性

注册	●
LDAP	○

组件

GitLab	10.5.4
GitLab Shell	6.0.3

[update asap](#)

知识讲解



创建用户（续1）

- 为wordpress项目创建主程序员用户

知识讲解

管理区域 > 新用户

新用户

账号

姓名	Zhangzhg * 必须填写
用户名	zhangzg * 必须填写
电子邮箱	zhangzg@tedu.cn * 必须填写



知识讲解

创建用户（续2）

- 为wordpress项目创建主程序员用户

devops 详细信息

成员

devops •
tedu devops
全局

按名称过滤...
名称
新项目
wordpress
wordpress project for jinkens
★ 0 10 minutes ago



创建用户（续3）

- 为wordpress项目创建主程序员用户

The screenshot shows the 'Members' section of a DevOps project named 'devops'. A search bar contains 'Zhangzhg'. A dropdown menu shows 'Programmer' selected. A note says 'Click here to learn more about role permissions'. A green button says 'Add to group'. Below is a search bar for existing members and a sorting dropdown.

The screenshot shows the 'Members' section of a DevOps project named 'devops'. A search bar contains 'Zhangzhg'. A dropdown menu shows 'Programmer' selected. A note says 'Click here to learn more about role permissions'. A green button says 'Add to group'. Below is a search bar for existing members and a sorting dropdown.

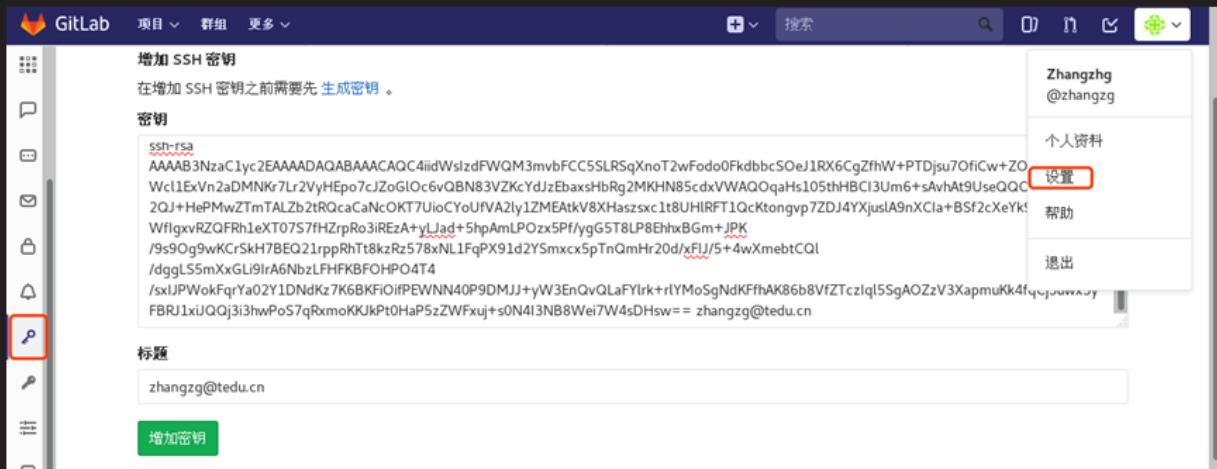
创建用户（续4）

- 用户生成ssh密钥

```
[root@localhost ~]# ssh-keygen -t rsa -C "zhangzg@tedu.cn" -b 4096
[root@localhost ~]# cat ~/.ssh/id_rsa.pub
ssh-rsa
AAAAB3NzaC1yc2EAAAQABAAQACQC4iidWslzdFWQM3mvbFCC5SL
RSqXnoT2wFodoOFkdbbcSOeJ1RX6CgZfhW+PTDjsu7OfiCw+ZOSlaeY0xQ
Wcl1ExVn2aDMNKr7Lr2VyHEpo7cJZoGIOc6vQBN83VZKcYdJzEbaxsHbRg
2MKHN85cdxVWAQOqaHs105thHBCI3Um6+sAvhAt9UseQQOQYBIIHO0
2QJ .....
```

创建用户（续4）

- 新用户登陆重置密码，然后设置ssh密钥



The screenshot shows the GitLab 'Add SSH Key' page. The user 'Zhangzg' is selected from the dropdown menu. The 'SSH Keys' section contains a long string of text representing the SSH public key. The 'Title' field is filled with 'zhangzg@tedu.cn'. A green 'Add Key' button is at the bottom. On the left sidebar, there is a red box around the 'P' icon.

知识讲解



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：

```
[root@localhost ~]# cd wordpress/
[root@localhost wordpress]# git remote rename origin old-origin
error: 不能重命名配置小节 'remote.origin' 到 'remote.old-origin'
上述错误可忽略
[root@localhost wordpress]# git remote add origin
git@192.168.113.139:devops/wordpress.git
[root@localhost wordpress]# git push -u origin --all
[root@localhost wordpress]# git push -u origin --tags
```

知识讲解



在网页中查看项目

- 所有的tag可以通过
<http://192.168.113.139/devops/wordpress/tags>



知识讲解

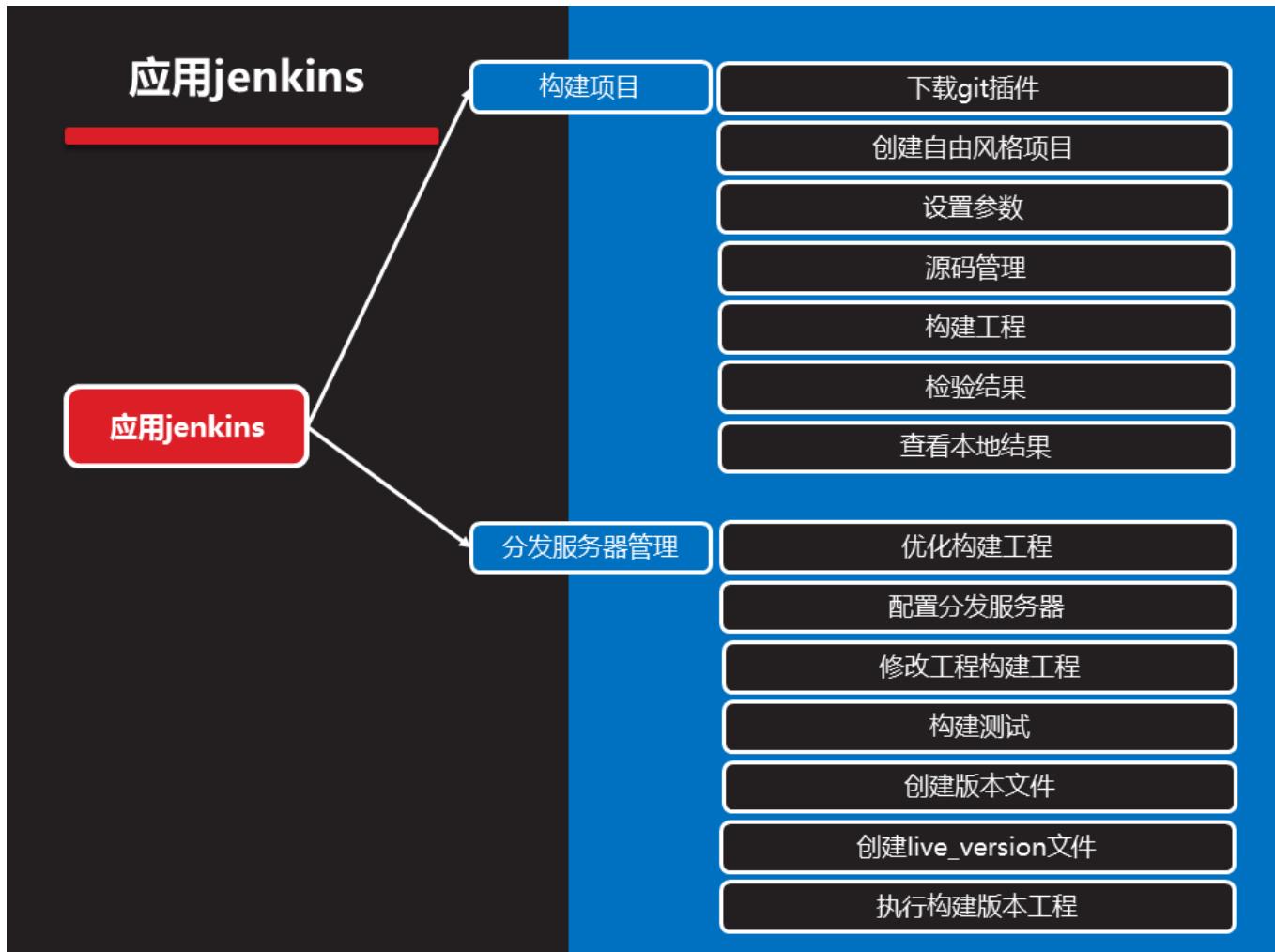


案例3：创建远程仓库

- 在gitlab的devops群组下创建wordpress项目
- 通过devops的主程序员用户将代码上传至wordpress项目
- 通过web查看项目

课堂练习





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

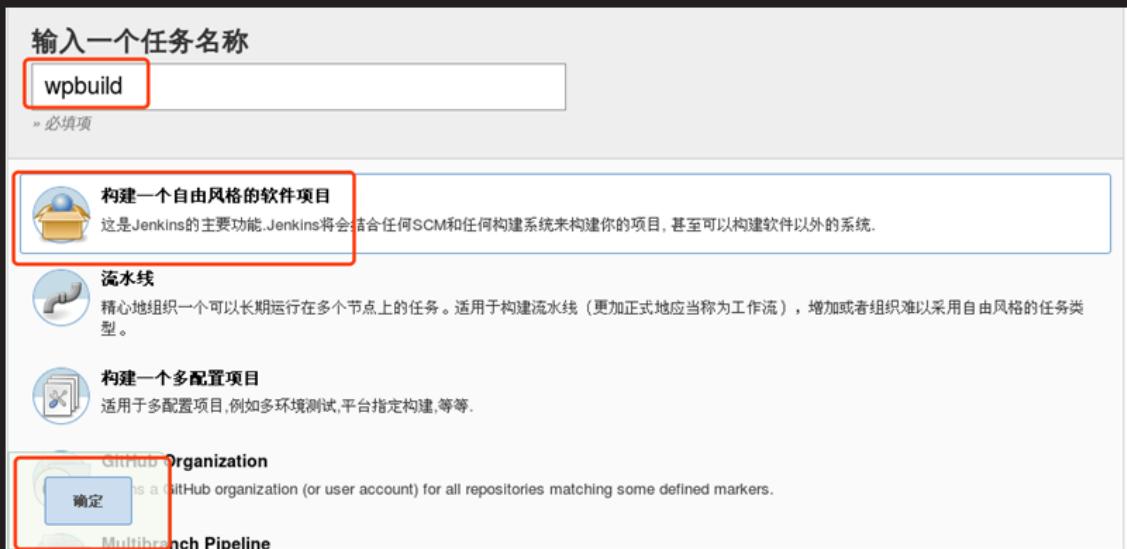
- 新建任务



创建自由风格项目（续1）

知识讲解

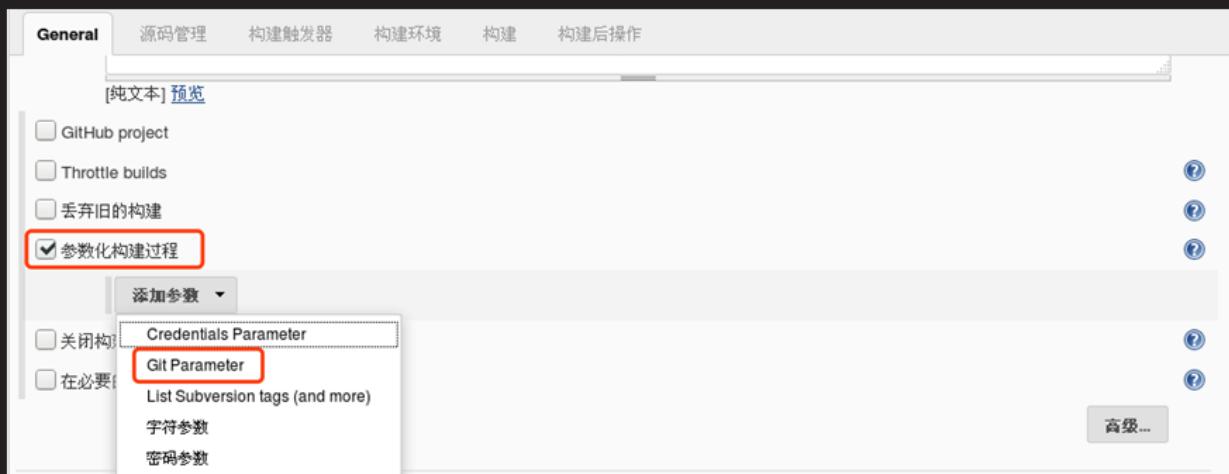
- 选择自由风格



设置参数

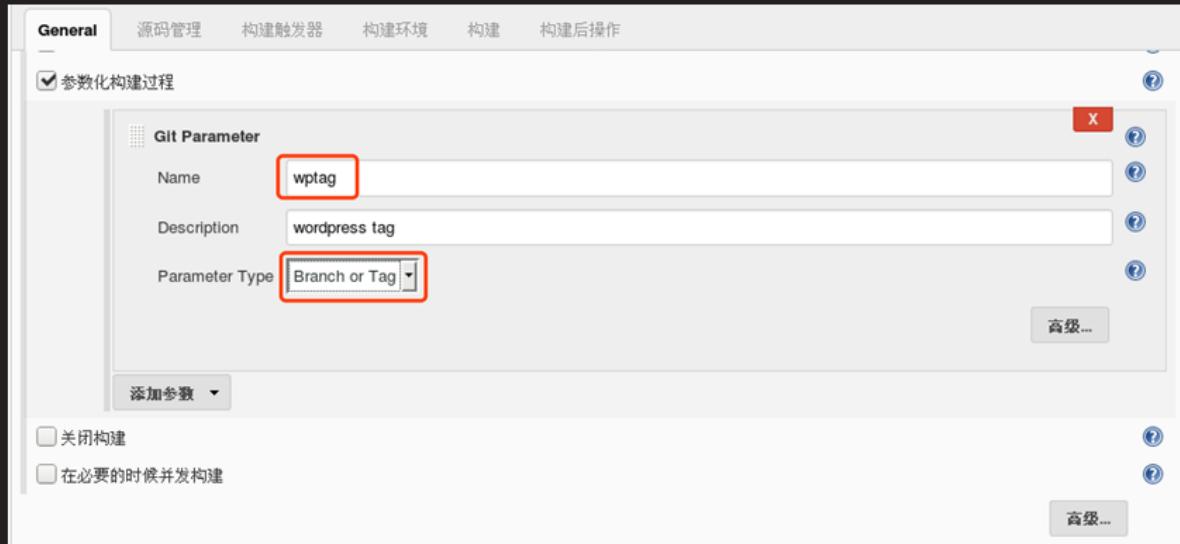
知识讲解

- 添加Git Parameter参数



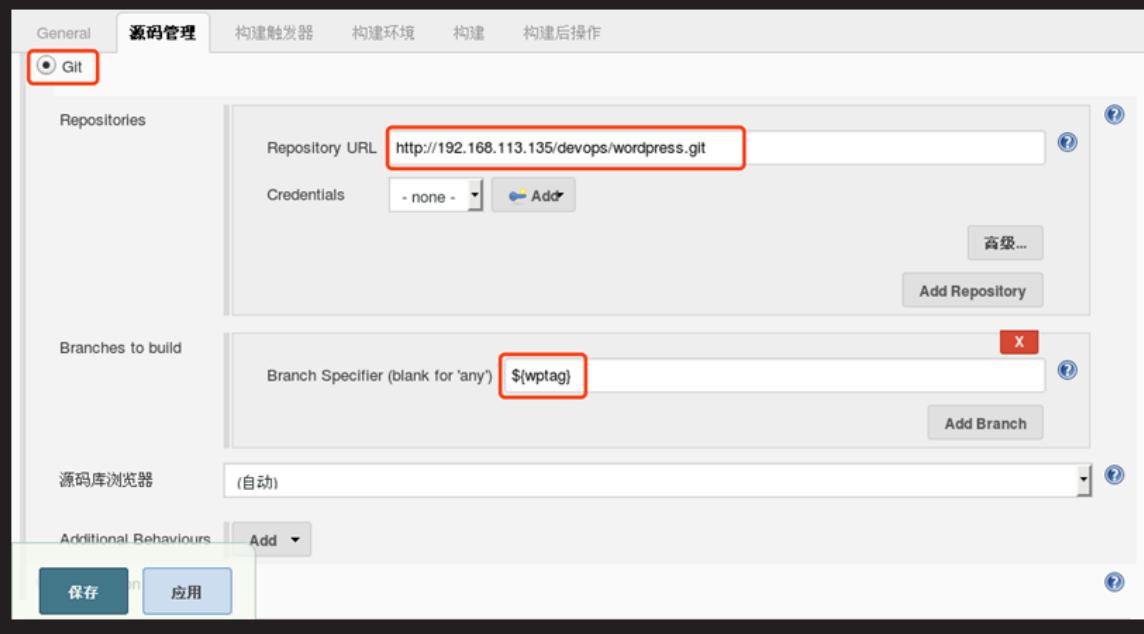
设置参数 (续1)

- 添加Git Parameter参数



源码管理

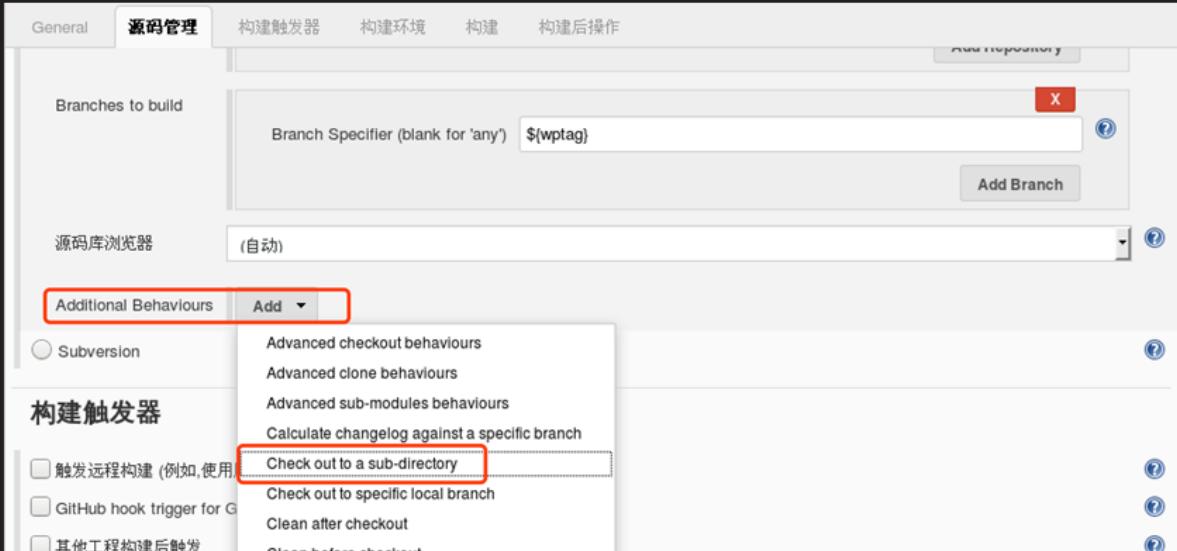
- 源码采用git



源码管理（续1）

- 将源码checkout到子目录

知识讲解



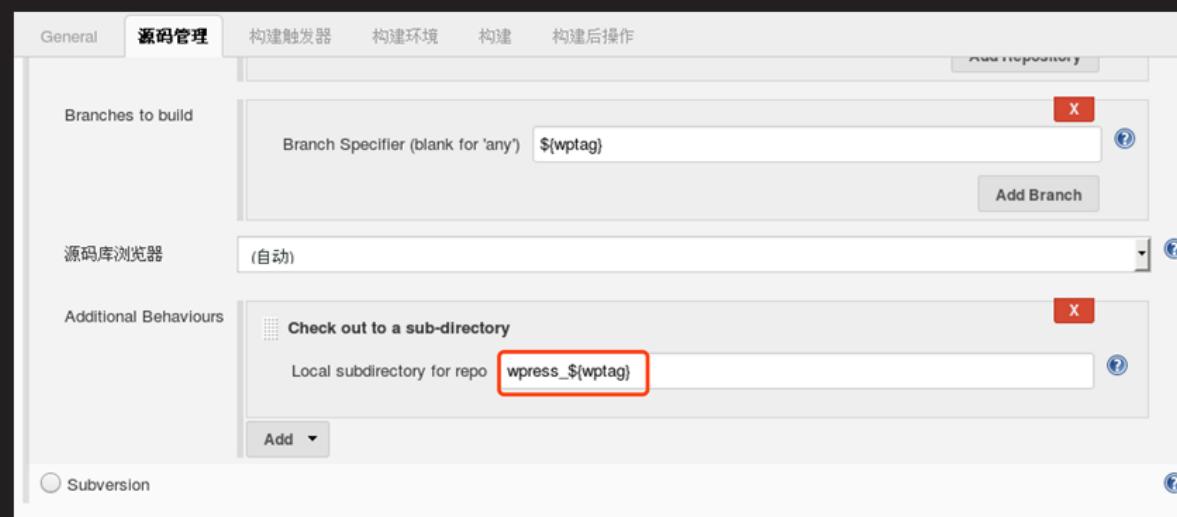
+
+
+

The screenshot shows the Jenkins configuration interface for a job. The 'Source Management' tab is selected. In the 'Additional Behaviours' section, a dropdown menu is open, showing various options. The option 'Check out to a sub-directory' is highlighted with a red box.

源码管理（续2）

- 将源码checkout到子目录

知识讲解



+
+
+

The screenshot shows the Jenkins configuration interface for a job. The 'Source Management' tab is selected. In the 'Additional Behaviours' section, the 'Check out to a sub-directory' option is selected. The 'Local subdirectory for repo' field contains the value 'wpress_\${wptag}', which is highlighted with a red box.

构建工程

- 构建工程

The screenshot shows the Jenkins interface for the 'wpbuild' project. On the left, there is a sidebar with the following options: 返回面板 (Back to Dashboard), 状态 (Status), 修改记录 (Change History), 工作空间 (Workspace), Build with Parameters (highlighted with a red box), 删除工程 (Delete Project), 配置 (Configure), and 重命名 (Rename). On the right, the main area is titled '工程 wpbuild'. It contains two sections: '工作区' (Workspace) and '最新修改记录' (Latest Change History). Below these sections is a '相关链接' (Related Links) section.



构建工程（续1）

- 选择指定的标签

The screenshot shows the Jenkins interface for the 'wpbuild' project. The sidebar is identical to the previous screenshot. In the main area, it says '需要如下参数用于构建项目:' (The following parameters are required for building the project:). A dropdown menu for 'wptag' contains 'v1.0' (which is highlighted with a red box) and 'origin/master'. At the bottom right, there is a large blue button labeled '开始构建' (Start Build), which is also highlighted with a red box.



检验结果

- 选择指定的标签

知识讲解



工程 wpbuild

相关链接

- 返回面板
- 状态
- 修改记录
- 工作空间
- Build with Parameters
- 删除 工程
- 配置
- 重命名

Build History 构建历史

find

#1 2018-6-25 下午7:54 RSS 全部 RSS 失败

检验结果（续1）

- 查看日志输出

知识讲解



构建 #1 (2018-6-25 19:54:33)

控制台输出

No changes.

启动用户 admin

Revision: 50984b10ab0ea24baa947754aa352e8e87fb210b
• v1.0

git

检验结果（续2）

- 查看日志输出



The screenshot shows the Jenkins control console with the title "控制台输出". The log output is as follows:

```

由用户 admin 启动
构建中 在工作空间 /var/lib/jenkins/workspace/wpbuild 中
Cloning the remote Git repository
Cloning repository http://192.168.113.135/devops/wordpress.git
> git init /var/lib/jenkins/workspace/wpbuild/wordpress_v1.0 # timeout=10
Fetching upstream changes from http://192.168.113.135/devops/wordpress.git
> git --version # timeout=10
> git fetch --tags --progress http://192.168.113.135/devops/wordpress.git +refs/heads/*:refs/remotes/origin/*
> git config remote.origin.url http://192.168.113.135/devops/wordpress.git # timeout=10
> git config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* # timeout=10
> git config remote.origin.url http://192.168.113.135/devops/wordpress.git # timeout=10
Fetching upstream changes from http://192.168.113.135/devops/wordpress.git
> git fetch --tags --progress http://192.168.113.135/devops/wordpress.git +refs/heads/*:refs/remotes/origin/*
> git rev-parse origin/v1.0^{commit} # timeout=10
> git rev-parse v1.0^{commit} # timeout=10
Checking out Revision 50984b10ab0ea24baa947754aa352e0e87fb210b (v1.0)
> git config core.sparsecheckout # timeout=10
> git checkout -f 50984b10ab0ea24baa947754aa352e0e87fb210b
Commit message: "wordpress init"
> git rev-list --no-walk 50984b10ab0ea24baa947754aa352e0e87fb210b # timeout=10
Finished: SUCCESS

```

知识讲解



查看本地结果

- 构建好的项目出现在/var/lib/jinkens目录下

```
[root@localhost ~]# cd /var/lib/jenkins/workspace/
[root@localhost workspace]# ls
wpbuild
[root@localhost workspace]# ls wpbuild/
wpress_v1.0
```

知识讲解



案例4：构建工程

课堂练习

- 创建一个自由风格的项目
- 源码管理采用git
- 构建时可以指定tag
- 构建tag为v1.0的源码



分发服务器管理

优化构建工程

知识讲解

- 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成为一个文件
- 为了应用服务器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

知识讲解

- 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

```
[root@localhost ~]# yum install -y httpd
[root@localhost ~]# mkdir -pv /var/www/html/deploy/packages
[root@localhost ~]# chown -R jenkins:jenkins /var/www/html/deploy/
[root@localhost ~]# systemctl start httpd
[root@localhost ~]# systemctl enable httpd
```



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

知识讲解

The screenshot shows the Jenkins interface for the 'wpbuild' project. On the left, there's a sidebar with links: '返回面板', '状态', '修改记录', '工作空间', 'Build with Parameters', '删除工程', '配置' (which is highlighted with a red box), and '重命名'. The main area is titled '工程 wpbuild' and contains sections for '工作区' (with a folder icon) and '最新修改记录' (with a notepad icon). Below these are '相关链接'.

+

修改工程构建工程（续1）

- 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

知识讲解

The screenshot shows the '构建环境' (Build Environment) tab in Jenkins. It includes sections for '构建' (Build) and '构建后操作' (Post-build Actions). In the '构建' section, there's a dropdown menu '增加构建步骤' (Add build step) with options like 'Invoke Ant', 'Invoke Gradle script', 'Run with timeout', 'Set build status to "pending" on GitHub commit', '执行 Windows 批处理命令' (Execute Windows batch command), '执行 shell' (Execute shell) (which is highlighted with a red box), and '调用顶层 Maven 目标' (Invoke top-level Maven goal). Buttons for '保存' (Save) and '应用' (Apply) are at the bottom.

+

修改工程构建工程（续2）

- 为下载的应用打包，以及生成md5可以通过在工程中增加构建步骤完成

知识讲解



+

构建测试

- 执行构建工程

知识讲解



+

构建测试（续1）

- 检查分发服务器的相关目录

```
[root@localhost ~]# ls /var/www/html/deploy/packages/  
wpress_v1.0.tar.gz wpress_v1.0.tar.gz.md5
```

知识
讲解

```
[root@localhost ~]# cat  
/var/www/html/deploy/packages/wpress_v1.0.tar.gz.md5  
e4fbaef54a5f580b4e08d5245cc95268
```



案例5：修改工程

课堂练习

- 修改构建工程，要求如下：
 - 将下载的应用程序打包放在
/var/www/html/deploy/packages目录下
 - 为打包应用程序计算md5值



创建版本文件

知识讲解

- 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

知识讲解

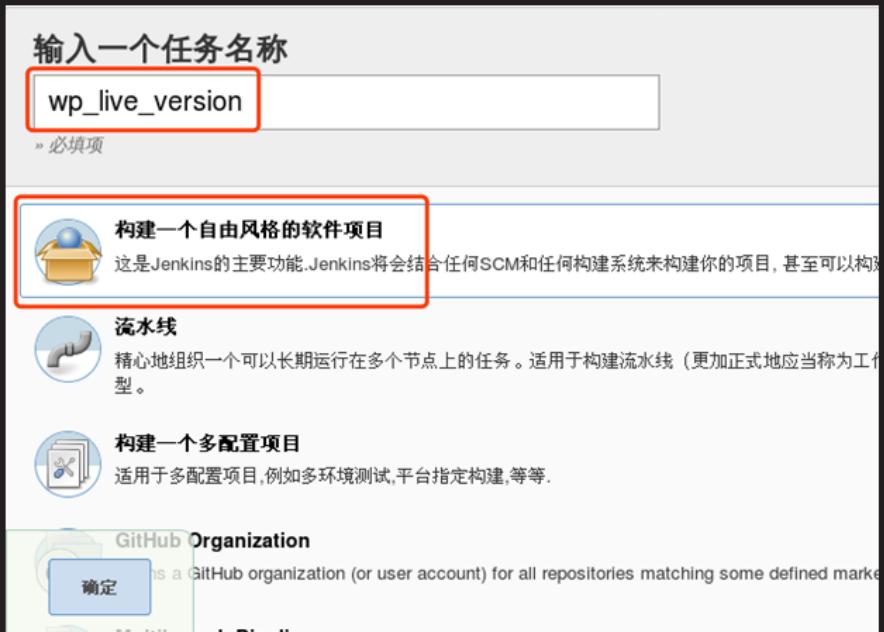
- 为了获所正确的版本，通过jenkins工程实现

The screenshot shows the Jenkins dashboard. On the left, there is a sidebar with icons for '新建任务' (New Job), '用户' (Users), '构建历史' (Build History), '系统管理' (System Management), '我的视图' (My Views), 'Credentials', and '新建视图' (New View). In the center, there is a table listing existing Jenkins jobs. One job, named 'wpbuild', is shown with a status of '11 分 - #6'. A button labeled '创建live_version文件' (Create live_version file) is located at the bottom of the job list. The '新建任务' button is highlighted with a red rectangle.



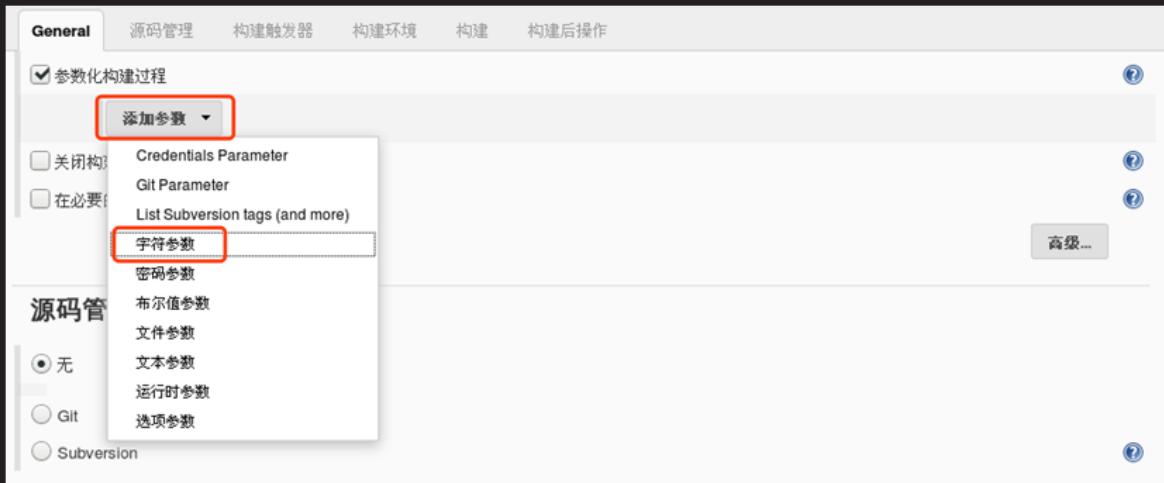
创建live_version文件（续1）

- 为了获所正确的版本，通过jenkins工程实现



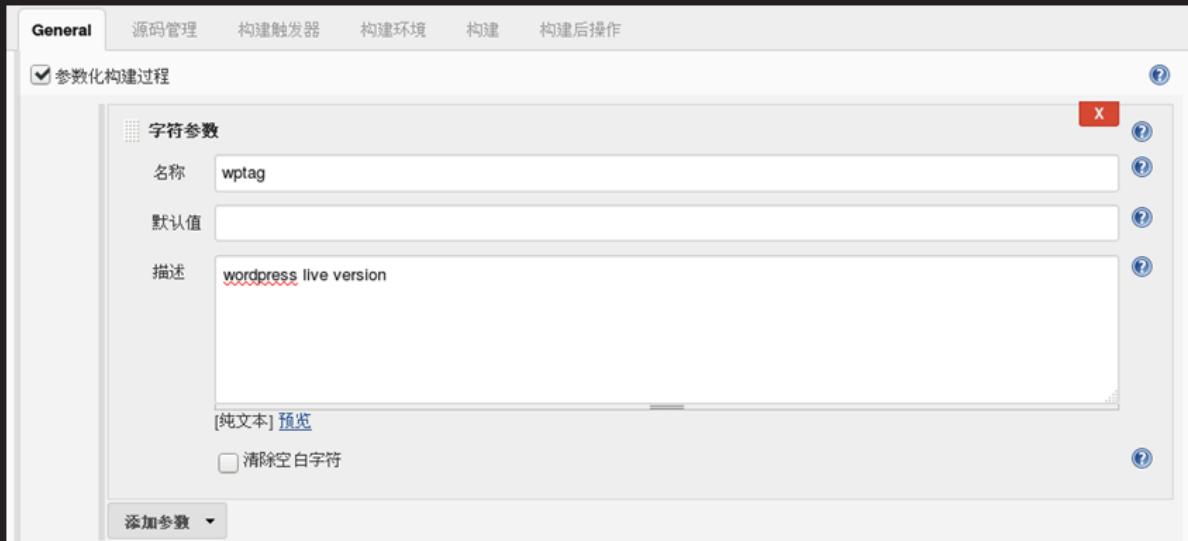
创建live_version文件（续2）

- 为了获所正确的版本，通过jenkins工程实现



创建live_version文件（续3）

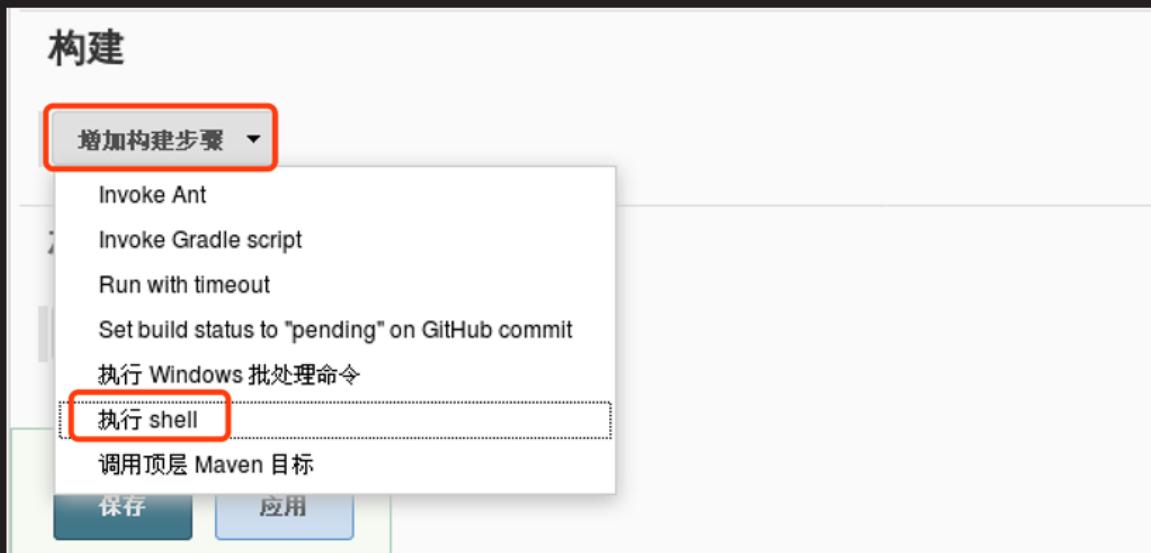
- 为了获所正确的版本，通过jenkins工程实现



知识讲解

创建live_version文件（续4）

- 为了获所正确的版本，通过jenkins工程实现



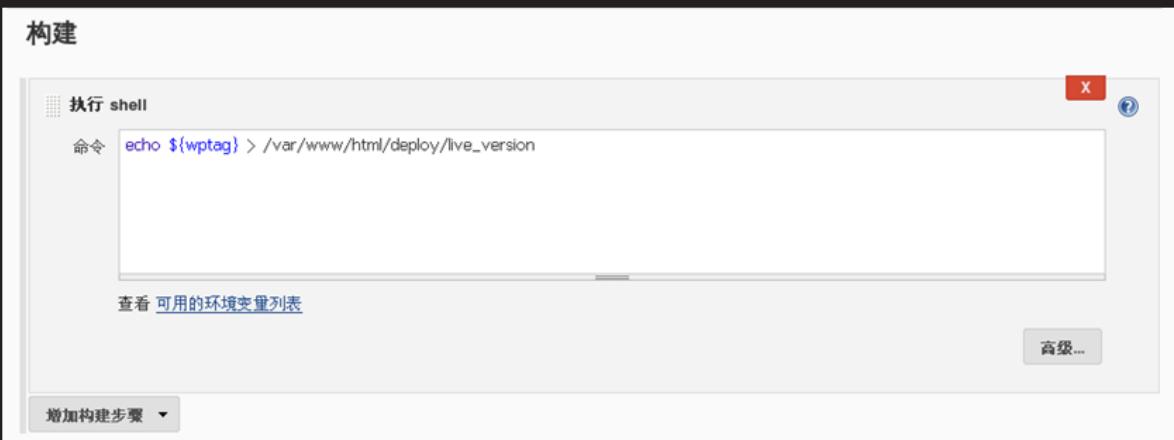
知识讲解



创建live_version文件（续5）

- 为了获所正确的版本，通过jenkins工程实现

知识讲解



知识讲解

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

The screenshot shows the Jenkins project page for 'wp_live_version'. The top navigation bar includes 'Jenkins' and 'wp_live_version'. On the left, there is a sidebar with links: '返回面板' (Return to Panel), '状态' (Status), '修改记录' (Change History), '工作空间' (Workspace), 'Build with Parameters' (Build with Parameters), '删除 工程' (Delete Project), '配置' (Configure), and '重命名' (Rename). The main content area is titled '工程 wp_live_version'. It displays the message '需要如下参数用于构建项目：' (The following parameters are required for building the project:). A parameter 'wptag' is shown with the value 'v1.0' in a text input field. Below the input field is the text 'wordpress live version'. At the bottom is a large blue '开始构建' (Start Build) button.



执行构建版本工程（续1）

- 查看构建结果

```
[root@localhost ~]# ls /var/www/html/deploy/  
live_version packages
```

```
[root@localhost ~]# cat /var/www/html/deploy/live_version  
v1.0
```

知识讲解

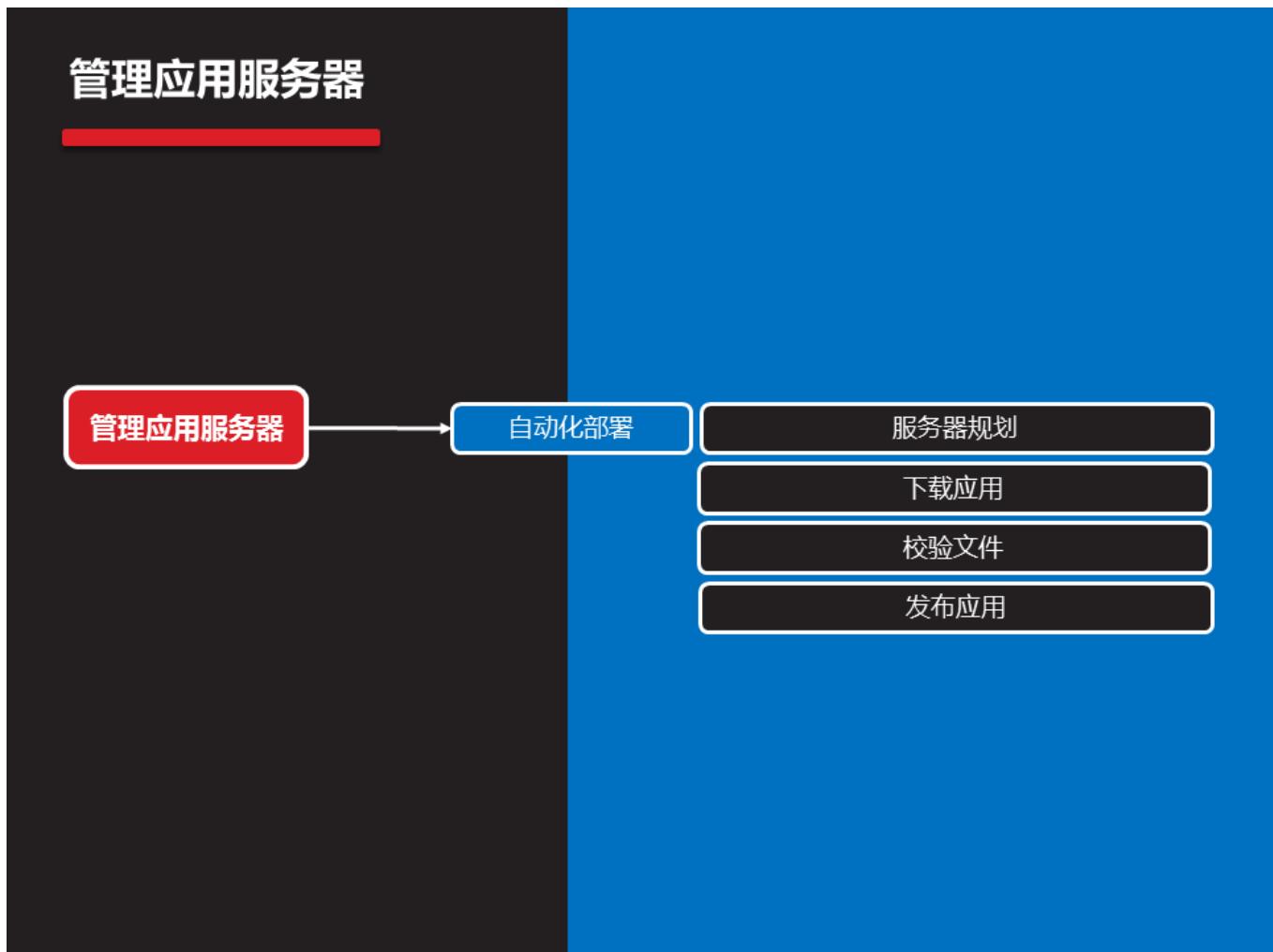


案例6：创建版本文件

- 为了记录应用的当前版本和前一个版本，创建两个工程：
 - 创建live_version，记录应用程序当前版本
 - 创建last_version，记录应用程序前一个版本

课堂练习





Tedu.cn
达内教育

自动化部署

服务器规划

知识讲解

- 为了方便版本的切换，可以规划如下目录
 - /var/www/download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软链接，指向需要部署的应用版本



下载应用

知识讲解

- 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



校验文件

知识讲解

- 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

知识讲解

- 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



案例7：发布应用

课堂练习

- 编写应用发布程序
 - 根据分发服务器的版本，下载对应的应用
 - 校验应用程序
 - 发布指定程序



总结和答疑

