

jenkins

```
version: '3'
services:
  docker_jenkins:
    restart: always
    image: jenkins/jenkins:lts
    container_name: docker_jenkins
    ports:
      - '8080:8080'
      - '50000:50000'
    volumes:
      - ./data:/var/jenkins_home
      - /var/run/docker.sock:/var/run/docker.sock
      - /usr/bin/docker:/usr/bin/docker
```

创建 data 目录并设置权限

mkdir ./data sudo chown -R 1000 ./data //把当前目录的拥有者赋值给uid 1000 若已启动则需要重建下服务

构建Jenkins服务

docker-compose up -d

...访问 <http://ip:8080> 进行初始化...

jenkins在docker中安装后查看登录令牌

使用命令 docker logs 容器名称/容器ID 查看访问日志，即可查看到登录令牌

若未设置账户，如何查看admin的密码

一不小心，未创建新的账户，可通过查看挂载目录下的 /secrets/initialAdminPassword 中的 Key 作为密码登录 账户：admin 密码：cat ./data/secrets/initialAdminPassword

在页面输入密码，点击Continue，进入插件安装页面。我们点击左边的Install suggested plugins，安装推荐插件就好。



Customize Jenkins

Plugins extend Jenkins with additional features to support many different needs.

Install suggested plugins

Install plugins the Jenkins community finds most useful.

Select plugins to install

Select and install plugins most suitable for your needs.

Jenkins 2.60.3

Getting Started



✓ Folders	OWASP Markup Formatter	Build Timeout	Credentials Binding	Folders ** bouncycastle API ** Struts
🔄 Timestampers	Workspace Cleanup	Ant	Gradle	
🔄 Pipeline	GitHub Branch Source	Pipeline: GitHub Groovy Libraries	Pipeline: Stage View	
🔄 Git	Subversion	SSH Slaves	Matrix Authorization Strategy	
🔄 PAM Authentication	LDAP	Email Extension	Mailer	
				** - required dependency

Jenkins 2.60.3

安装好插件后，系统会提示建立管理员账户。

Getting Started

Create First Admin User

用户名:

密码:

确认密码:

全名:

电子邮件地址:

Jenkins 2.60.3

Continue as admin

Save and Finish

创建管理员账户后进入系统，我们先来配置常用工具。

Jenkins

新建

用户

任务历史

系统管理

My Views

Credentials

构建队列

队列中没有构建任务

构建执行状态

1 空闲

2 空闲

管理Jenkins

Jenkins新版本 (2.107.3)可点击 download (变更说明)下载。

Warnings have been published for the following currently installed components:

Jenkins 2.60.3 core and libraries:

Multiple security vulnerabilities in Jenkins 2.120 and earlier, and LTS 2.107.2 and earlier

Multiple security vulnerabilities in Jenkins 2.88 and earlier, and LTS 2.73.2 and earlier

Multiple security vulnerabilities in Jenkins 2.106 and earlier, and LTS 2.89.3 and earlier

Multiple security vulnerabilities in Jenkins 2.83 and earlier, and LTS 2.73.1 and earlier

Multiple security vulnerabilities in Jenkins 2.94 and earlier, and LTS 2.89.1 and earlier

Multiple security vulnerabilities in Jenkins 2.115 and earlier, and LTS 2.107.1 and earlier

Configure which of these warnings are shown

系统设置

全局设置&路径

Configure Global Security

Secure Jenkins, define who is allowed to access/use the system.

Configure Credentials

Configure the credential providers and types

Global Tool Configuration

Configure tools, their locations and automatic installers.

读取设置

放弃当前内存中所有的设置信息并从配置文件中重新读取 仅用于当您手动修改配置文件时重新读取设置。

管理插件

添加、删除、禁用或启用Jenkins功能扩展插件。

系统信息

进入工具配置页，首先看到的就是jdk配置，这里我们先使用自动安装的java8。

JDK

JDK 安装

JDK

别名

java8

自动安装

从 java.sun.com 安装

版本: Java SE Development Kit 8u172

我同意 Java SE Development Kit 的许可协议

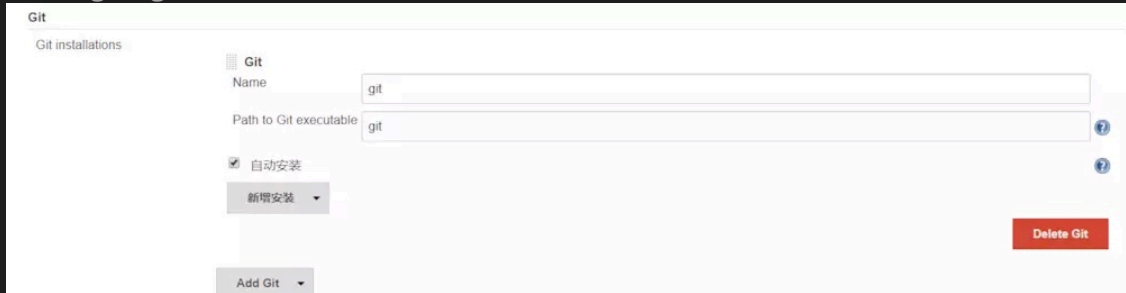
Installing JDK requires Oracle account. Please enter your username/password

新增安装

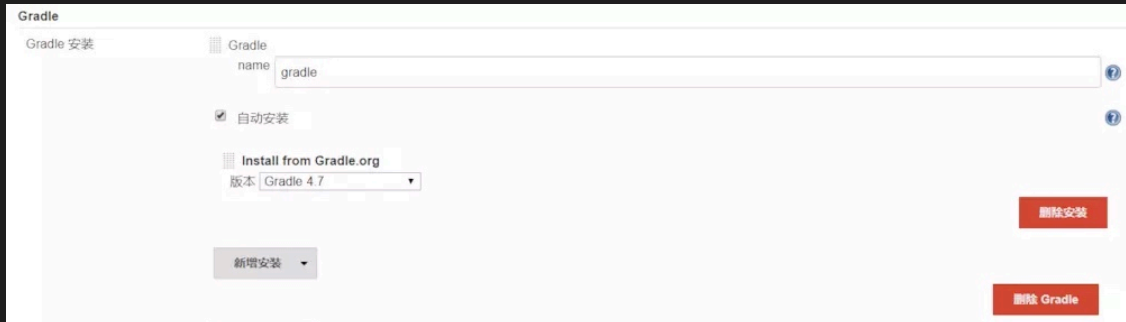
删除安装

删除 JDK

注意，自动安装jdk需要你填入oracle的账号密码，这样Jenkins才能自动帮你到oracle安装。如果没有oracle账号，建议先去Oracle官网注册一下。配置好jdk自动安装后，我们继续配置git。git的字段安装更简单，起个别名，勾上自动安装。



配置好git我们继续配置Gradle，一样简单，起个别名，勾上自动安装，选择Gradle的版本。



生成SSHKey

容器从github上面下载代码的时候需要用到。因为咱们这个Jenkins是基于docker的，不是直接安装在服务器上，这里有很大的不同。

要生成容器的SSHKey，首先要先进入容器

```
docker exec -it jenkins /bin/bash
# 进入容器后生成sshkey
ssh-keygen -t rsa -C "123@qq.com"
```

```
[root@ip-10-0-0-154 ~]# docker exec -it jenkins /bin/bash
jenkins@873812bdfc5b:/$ ssh-keygen -t rsa -C "jenkins@873812bdfc5b@qq.com"
Generating public/private rsa key pair.
Enter file in which to save the key (/var/jenkins_home/.ssh/id_rsa):
Created directory '/var/jenkins_home/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /var/jenkins_home/.ssh/id_rsa.
Your public key has been saved in /var/jenkins_home/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:pV+rQxBziecLN0v4nbZ10S3twxJvw5/E8z+hN9zg4KA jenkins@873812bdfc5b@qq.com
The key's randomart image is:
+---[RSA 2048]-----+
|
|  .  .
| * +
|
| . 0 .
|
| +o . o .
| oSo ...o +
| . .+. .B=
| . .+.o++%+
| E =o.o==0
| .oo ..B
+-----[SHA256]-----+
jenkins@873812bdfc5b:/$
```

在容器内输入exit离开容器，使用如下命令获取公钥

```
tail /var/lib/docker/volumes/jenkins/_data/.ssh/id_rsa.pub
```

```
[root@ip-10-0-0-154 ~]# tail /var/lib/docker/volumes/jenkins/_data/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQBAQ...
jenkins@873812bdfc5b:/$
```

jenkins构建时支持git选择分支

安装Git Parameter插件

The screenshot shows the Jenkins web interface. On the left sidebar, the 'System Management' tab is selected. In the main content area, the 'Manage Jenkins' page is displayed. Under the 'Manage Jenkins' section, the 'Manage Plugins' option is highlighted with a red box. The 'Manage Plugins' section shows a list of plugins, including 'Git Parameter', which is marked as 'Available for update'.

然后点击选插件，在右上角输入Git Parameter，找到Git Parameter后点击左下方的的直接安装

The screenshot shows the Jenkins 'Available Plugins' page. The search bar at the top right contains 'Git Parameter'. The table below lists several plugins, including 'CCM Plug-in', 'ExCop Runner plugin', and 'MSBuild Plugin'. The 'Git Parameter' plugin is not visible in the current view.

安装	名称	版本
<input type="checkbox"/>	CCM Plug-in	3.1
<input type="checkbox"/>	ExCop Runner plugin	1.1
<input type="checkbox"/>	MSBuild Plugin	1.26

插件安装成功后，在项目配置中后看到多了一个“参数化构建过程” 勾选“参数化构建过程”，然后添加Git Parameter,配置如下

The screenshot shows the 'Git Parameter' configuration page in Jenkins. At the top, there are three checkboxes: '启用项目安全' (disabled), '丢弃旧的构建' (disabled), and '参数化构建过程' (checked). Below this is the 'Git Parameter' section with the following fields: 'Name' (BRANCH), 'Description' (empty), 'Parameter Type' (Branch), 'Branch' (http://blog.csdn.net/), 'Branch filter' (*), 'Tag filter' (*), 'Sort mode' (ASCENDING), 'Default Value' (origin/dev), and 'Quick filter' (checkbox). A red '删除' (Delete) button is at the bottom right.

在源码管理中添加如下：

The screenshot shows the 'Source Management' configuration page in Jenkins. Under the 'Git' section, there are 'Repositories' and 'Branches to build' sections. The 'Repositories' section has a 'Repository URL' field (redacted) and a 'Credentials' dropdown (redacted). The 'Branches to build' section has a 'Branch Specifier (blank for 'any')' field with the value '\$BRANCH'. There are buttons for 'Add Repository', 'Delete Repository', 'Add Branch', and 'Delete Branch'. At the bottom, there is an 'Additional Behaviours' section with an 'Add' button.

大功告成，原本的“立刻构建”变成了“Build with Parameters”

The screenshot shows the Jenkins Project configuration page. On the left sidebar, there is a list of actions: '返回面板', '状态', '修改记录', '工作空间', 'Build with Parameters' (highlighted), '删除 Project', and '配置'. The main area shows the 'Project' name (redacted) and a list of parameters for the build: 'BRANCH' with values 'origin/approve_2', 'origin/dev', 'origin/feature-0.0.2', and 'origin/master'. Below the list, there is a '开始构建' (Build) button. At the bottom, there is a 'Build History' section with a search bar and a list of builds.

General

源码管理

构建触发器

构建环境

构建

构建后操作

描述

支付系统

[纯文本] 预览

☐ GitHub 项目

☐ This build requires lockable resources

☐ Throttle builds

☐ 丢弃旧的构建

☒ 参数化构建过程

布尔值参数

名称

确定

默认值

☒

描述

[纯文本] 预览

保存

应用

Git Parameter

Git Parameter

Name

branch

Description

分支选择

[纯文本] 预览

Parameter Type

Branch

Default Value

origin/master

高级...

添加参数

☐ 关闭构建

☐ 在必要的时候并发构建

高级...

源码管理

源码管理

☐ 无

☒ Git

Repositories

Repository URL

http://192.168.80.131:8080/tuheishan/clab.git

Credentials

tuheishan/*****

添加

高级...

Add Repository

Branches to build

Branch Specifier (blank for 'any')

\$branch

Add Branch

源码库浏览器

(自动)

Additional Behaviours

新增

☐ Subversion

构建触发器

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
- ☐ GitHub hook trigger for GITScm polling
- ☐ 轮询 SCM



构建环境

- ☐ Delete workspace before build starts
- ☐ Use secret text(s) or file(s)
- ☐ Send files or execute commands over SSH before the build starts
- ☒ Send files or execute commands over SSH after the build runs



SSH Publishers

SSH Server

Name 192.168.80.136



高级...

Transfers

Transfer Set



Source files clab-spring-cloud-pay/target/*.jar



Remove prefix clab-spring-cloud-pay/target/



Remote directory /



Exec command pwd



保存

应用

Transfer Set



Source files clab-spring-cloud-pay/Dockerfile



Remove prefix clab-spring-cloud-pay/



Remote directory /

构建触发器



Exec command pwd



All of the transfer fields (except for Exec timeout) support substitution of [Jenkins environment variables](#)

高级...

Transfer Set



Source files docker-compose.yml



Remove prefix /



Remote directory /



Exec command pwd



All of the transfer fields (except for Exec timeout) support substitution of [Jenkins environment variables](#)

高级...

Transfer Set

Source files

run.sh

Remove prefix

/

Remote directory

/

Exec command

cd /opt
pwd
chmod 777 run.sh
./run.sh

All of the transfer fields (except for Exec timeout) support substitution of [Jenkins environment variables](#)

高级...

Add Transfer Set

Add Server

高级...

General 源码管理 构建触发器 构建环境 构建 构建后操作

☒ Add timestamps to the Console Output
☐ Inspect build log for published Gradle build scans
☐ With Ant

构建

调用顶层 Maven 目标

Maven 版本

maven

目标

clean install -Dmaven.test.skip=true -Ptest

高级...

增加构建步骤

构建后操作

增加构建后操作步骤

```
#!/bin/bash
echo '==开始docker'
pwd
echo '==删除旧镜像'
docker rmi -f 192.168.80.134:5000/pay:1.0.0
echo '==打包镜像'
sudo docker build --rm -t pay:1.0.0 .
echo '==tag===== '
docker tag pay:1.0.0 192.168.80.134:5000/pay:1.0.0
echo '==推送到镜像仓库===== '
docker push 192.168.80.134:5000/pay:1.0.0
echo '==启动镜像===== '
docker-compose up -d
```

```
version: '2'
services:
  hive-auth2-server:
    image: pay:1.0.0
    restart: always
    ports:
      - 9090:9090
    volumes:
      - /data/logs/pay/:/logs/pay/
```

```
FROM java:8-jre

EXPOSE 9090

VOLUME /tmp
ADD target/clab-spring-cloud-pay-1.0.0.RELEASE.jar /app.jar
RUN bash -c 'touch /app.jar'
ENV JAVA_OPTS="-Xmx512m -Xms512m"
ENV CONFIG_PROFILE=dev

ENTRYPOINT ["sh", "-c", "java $JAVA_OPTS \
-Dfile.encoding=UTF8 -Duser.timezone=GMT+08 \
-Djava.security.egd=file:/dev/./urandom \
-jar /app.jar \
--spring.profiles.active=$CONFIG_PROFILE"]
```

#####思路

jenkins是docker安装的，所以内部命令较少，可以手动进入docker安装所需命令，这样，就可以在jenkins中执行大部分shell脚本，从而打包后，可以在jenkins配置执行脚本，将制作镜像和上传镜像仓库步骤放到jenkins中，ssh远程服务器只需要拉取镜像，运行就行。