Jenkins

象翌微链科技发展有限公司

**六脉30实验室**

**2018年10月**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **文档变更记录** | | | | |
| **版本** | **日期** | **状态** | **修改人** | **修改内容** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

目录

[1 概述 6](#_Toc517790251)

[2 范式 6](#_Toc517790252)

[2.1 物联 6](#_Toc517790253)

[2.2 计算 6](#_Toc517790254)

[2.3 结构 6](#_Toc517790255)

[3 场景 6](#_Toc517790256)

[4 流程 6](#_Toc517790257)

[5 功能 6](#_Toc517790258)

# 概述

Jenkins是一个功能强大的应用程序，允许持续集成和持续交付项目，无论用的是什么平台。这是一个免费的源代码，可以处理任何类型的构建或持续集成。集成Jenkins可以用于一些测试和部署技术。Jenkins是一种软件允许持续集成。

# 目的

## 持续、自动地构建/测试软件项目

## 监控软件开放流程，快速问题定位及处理，提示开放效率

# 特性

开源的java语言开发持续集成工具，支持CI，CD。

易于安装部署配置：可通过yum安装,或下载war包以及通过docker容器等快速实现安装部署，可方便web界面配置管理。

消息通知及测试报告：集成RSS/E-mail通过RSS发布构建结果或当构建完成时通过e-mail通知，生成JUnit/TestNG测试报告。

分布式构建：支持Jenkins能够让多台计算机一起构建/测试。

文件识别:Jenkins能够跟踪哪次构建生成哪些jar，哪次构建使用哪个版本的jar等。

丰富的插件支持:支持扩展插件，你可以开发适合自己团队使用的工具，如git，svn，maven，docker等。

# 流程

产品设计成型 -> 开发人员开发代码 -> 测试人员测试功能 -> 运维人员发布上线   
持续集成 （Continuous integration，简称CI）   
持续交付（Continuous delivery）   
持续部署（continuous deployment）

# 安装部署

## 5.1 yum安装

最低配置： 不少于256M内存，不低于1G磁盘，jdk版本>=8   
安装jdk1.8

yum install -y java-1.8.0-openjdk && \

wget -O /etc/yum.repos.d/jenkins.repo https://pkg.jenkins.io/redhat/jenkins.repo && \

rpm --import https://pkg.jenkins.io/redhat/jenkins.io.key &&\

yum clean all && yum makecache && \

yum install -y jenkins && \

systemctl start jenkins

less /var/log/jenkins/jenkins.log //查询admin密码

访问 http://ip:8080/ 进行安装

jenkins相关目录：

rpm -ql jenkins #查看jenkins安装相关目录

安装目录/var/lib/jenkins

配置文件 /etc/sysconfig/jenkins

日志目录 /var/log/jenkins

2.2 下载war包安装

yum install -y java-1.8.0-openjdk && mkdir -p /usr/local/jenkins/ && \

wget -c -O /usr/local/jenkins/jenkins.war http://mirrors.jenkins.io/war-stable/latest/jenkins.war &&\

nohup java -jar /usr/local/jenkins/jenkins.war &

1

2

3

密码查看：/root/.jenkins/secrets/initialAdminPassword

访问 http://ip:8080/ 进行安装

2.3 docker形式安装

yum install docker -y && systemctl start docker && \

curl -sSL https://get.daocloud.io/daotools/set\_mirror.sh | sh -s && systemctl restart docker

docker search jenkins

docker pull docker.io/jenkins

docker run -d -p 8080:8080 docker.io/jenkins

1

2

3

4

5

docker exec -it 52873fa3dce4 /bin/bash

cat /var/jenkins\_home/secrets #查看密码

访问 http://ip:8080/ 进行安装

2.4 相关工具安装

工具安装可在jenkins的web界面系统管理—管理插件—可选插件勾选需要的，也可以手动安装。

a.git安装

yum install curl-devel expat-devel gettext-devel openssl-devel zlib-devel gcc-c++ perl-ExtUtils-MakeMaker wget autoconf -y

wget https://www.kernel.org/pub/software/scm/git/git-2.7.3.tar.gz

tar xf git-2.7.3.tar.gz

cd git-2.7.3

make configure

./configure --prefix=/usr/local/git

make profix=/usr/local/git

make install

echo "export PATH=$PATH:/usr/local/git/bin" > /etc/profile.d/git.sh

source /etc/profile.d/git.sh

1

2

3

4

5

6

7

8

9

10

b.maven安装

cd /usr/local/ && wget -c wget http://mirror.bit.edu.cn/apache/maven/maven-3/3.5.2/binaries/apache-maven-3.5.2-bin.tar.gz &&\

tar -zxf apache-maven-3.5.2-bin.tar.gz && \

echo "export PATH=$PATH:/usr/local/apache-maven-3.5.2/bin" > /etc/profile.d/maven.sh && \

source /etc/profile.d/maven.sh

1

2

3

4

mvn –version 查看版本

c.svn安装

yum install subversion -y

mkdir /svnrepo

svnadmin create /svnrepo/project1 #创建一个新的Subversion项目

1

2

3

启动项目1：svnserve -d -r /svnrepo/project1/

启动整个项目：svnserve -d -r /svnrepo/

d.jdk安装

某些情况下会遇到openjdk无法编译java程序，或在编译中存在异常，需要我们手动安装jdk，首先去Oracle官网下载对应版本的jdk

tar -zxvf jdk-8u121-linux-x64.gz -C /usr/local && \

ln -sv /usr/local/jdk1.8.0\_121 /usr/local/jdk1.8 && \

echo "export PATH=$PATH:/usr/local/jdk1.8/bin" >/etc/profile.d/java.sh

source /etc/profile.d/java.sh

1

2

3

4

三、WEB界面配置

3.1 密码破解

cd /var/lib/jenkins/users/admin #此为admin用户的目录，也可以对应自己创建的用户

vim config.xml //定位到<passwordHash>那一行

删除改行，改为

<passwordHash>#jbcrypt:$2a$10$pDQks0ytOkCfmpdgpLygrOC3uY7i/XnZHBKRQDhrBPwKoN2f5Kz8C</passwordHash>

重启一下jenkins

新密码为admin

3.2 maven配置

针对之前手动安装的maven需要在web界面配置进行结合。

系统管理—全局工具配置（Maven Configuration）配置：

此处为手动安装的maven，需要取消自动安装maven，name可以自定义，MAVEN\_HOME指定下载安装的maven具体文件目录

3.3 GIT配置

手动安装的git，web界面只需添加具体可执行路径即可

3.4 jdk配置

对应手动安装的jdk，需要web配置java\_home路径

3.5 邮件配置

jenkins有内置的邮件功能但由于其对于成功的发布不会发送邮件可定制化程度低，在此直接配置插件邮件(Editable Email Notification)，2.32以上版本的jenkins，常用的插件都已经在最初部署的时候安装了（如果你当初选择的是suggest的plugins）。

可以使用qq邮件服务器或163等其他邮件服务器，在此配置使用163邮箱，首先需要开启163邮箱的smtp服务。

需要记录此处开启smtp时的密码。

在jenkins 系统管理—系统设置—Extended E-mail Notification

在默认的触发器 Default Triggers 标签下，可以勾选邮件通知触发的情况，如任何情况下均需要选择always，也可以在仅失败情况下发送，也可以对应的邮件接受人不通，进行邮件分类。

以下的内容，如邮件主题，内容均可自定义，可以参考url：

至此全局插件邮件已经配置完毕，需要在具体的项目中使用此邮件进行发送通知。

四、实战php程序自动发布

4.1 插件安装

系统管理—管理插件—已安装

检查是否有“Git plugin”和“Publish Over SSH”两个插件，如果没有，则需点击“可选插件”，找到它并安装

安装好两个插件后，点击“系统管理”– “系统设置”

4.2 SSH配置

系统管理—系统设置—Publish over SSH

在key内填写jenkins服务器的私钥，如果没有需要先在jenkins服务器生成私钥与公钥。ssh-keygen -t rsa回车后会在登录用户的家目录下生成一个.ssh 的目录，此目录下存在id\_rsa私钥与id\_rsa.pub公钥。且讲公钥发布至代码发布的目标服务器上ssh-copy-id -i /root/.ssh/id\_rsa.pub root@IP。

SSH Server配置

name:需要将php程序发布到目标服务器的名称，可自定义

Hostname:填写目录服务器的IP地址

Username:使用那个用户进行发布，此处为进行密钥互信的用户

Remote Directory：此出为发布到目标服务器的相对根路径，建议填写/，防止后续填写路径异常。

注：如果为多台目标服务器，可以继续添加，如果目标服务器存在代理，也可设置proxy

4.3 构建项目

新建Item—填入项目名称—选择构建一个自由风格的软件项目—确定

源码管理选择git

Repository URL 填写具体git上的仓库url，如果为私有，需要继续添加Credentials，如果为公有直接填写url即可，Credentials为none，

构建后够操作

选择（Send files or execute commands over SSH）

SSH Server选择目标服务器如：php-server

Source files：/ #将git拉去下来的原始文件

Remote directory：/var/www/html #发布到目标服务器的制定目录

Exec command：chown apache:apache -R /var/www/html/\* #制定后续的操作

此时可以选择Editable Email Notification来构建邮件通知。

在此处，之前的邮件主题，内容均可以自定义，在高级里面，选择邮件接受人。

点击保存，并立即构建，可以点击console output查看日志

此时打开php程序发现程序文件已经成功发布到目标服务器上

此时可以查看邮件也已经发送成功。