

Jenkins+Maven+SVN+Tomcat 持续集成环境快捷部署

搭建持续集成环境 Jenkins+Maven+SVN+Tomcat 实现自动编译打包部署。

1.环境准备

- (1) JDK1.8.0_131 #不低于 1.7 版本，这里用最新版本
- (2) Apache Maven 3.3.9 #可以选择 3.2.5 或者 3.3.9；不要使用 3.5.0 版本！！
- (3) SVN 客户端 (Subversion 1.6.11) #程序版本控制 SVN1.6.11
- (4) Tomcat1.7(apache-tomcat-7.0.77) #Tomcat7.0.77 或者 7.0.75
- (5) Centos6.7

2.安装配置

JAVA 环境安装配置

下载最新版本 1.8.0_131

```
cd /tmp/
```

```
wget http://download.oracle.com/otn-pub/java/jdk/8u131-b11/d54c1d3a095b4ff2b6607d096fa80163/jdk-8u131-linux-x64.tar.gz
```

```
tar xzf jdk-8u131-linux-x64.tar.gz -C /usr/local/
```

```
vim /etc/profile
```

添加如下内容

```
export JAVA_HOME=/usr/local/jdk1.8.0_131
```

```
export PATH=$JAVA_HOME/bin:$PATH
```

```
export CLASSPATH=. : $JAVA_HOME/jre/lib/rt.jar:$JAVA_HOME/lib/dt.jar:$JAVA_HOME/lib/tools.jar
```

然后执行

```
source /etc/profile
```

Maven 安装

到 Apache 官网下载 maven3.3.9 版本

```
wget
```

<http://mirrors.tuna.tsinghua.edu.cn/apache/maven/maven-3/3.3.9/binaries/apache-maven-3.3.9-bin.tar.gz>

```
tar xzf apache-maven-3.3.9-bin.tar.gz -C /usr/local/
```

```
vim /etc/profile
```

添加下面内容

```
export MAVEN_HOME=/usr/local/apache-maven-3.3.9
```

```
export PATH=$PATH:$MAVEN_HOME/bin
```

```
source /etc/profile
```

查看版本

```
[root@localhost tmp]# mvn -v
```

```
Apache Maven 3.3.9 (bb52d8502b132ec0a5a3f4c09453c07478323dc5; 2015-11-11T00:41:47+08:00)
```

```
Maven home: /usr/local/apache-maven-3.3.9
```

```
Java version: 1.8.0_131, vendor: Oracle Corporation
```

```
Java home: /usr/local/jdk1.8.0_131/jre
```

```
Default locale: zh_CN, platform encoding: UTF-8
```

```
OS name: "linux", version: "2.6.32-573.el6.x86_64", arch: "amd64", family: "unix"
```

Tomcat 安装

下载 Tomcat7.0_77

```
wget http://apache.fayea.com/tomcat/tomcat-7/v7.0.77/bin/apache-tomcat-7.0.77.zip
```

```
unzip apache-tomcat-7.0.77.zip
```

```
mv apache-tomcat-7.0.77 /usr/local/
```

```
cd /usr/local/apache-tomcat-7.0.77
```

安装 Jenkins

```
wget http://mirrors.jenkins.io/war-stable/1.609.3/jenkins.war
```

```
mv jenkins.war /usr/local/apache-tomcat-7.0.77/webapps/
```

```
cd /usr/local/apache-tomcat-7.0.77/webapps/
```

在 webapps 目录下启动 jenkins

执行命令 `java -jar jenkins.war`

(关闭 `selinux` 和 `iptables`)

浏览器访问 jenkins `http://192.168.1.7:8080/`

-----分割线-----

局域网内利用 GitLab+Jenkins 自动生成 GitBook 并发布(Nginx) <http://www.linuxidc.com/Linux/2016-05/131136.htm>

Linux+Git+Maven+Jenkins+Neuxs 自动化编译环境搭建 <http://www.linuxidc.com/Linux/2016-02/128652.htm>

CentOS6 安装 Jenkins <http://www.linuxidc.com/Linux/2016-05/131365.htm>

使用 Jenkins 配置 Git+Maven 的自动化构建 <http://www.linuxidc.com/Linux/2016-02/128641.htm>

Jenkins+Maven+Git 搭建持续集成和自动化部署的配置手记 <http://www.linuxidc.com/Linux/2015-06/118606.htm>

Jenkins 的分布式构建及部署——节点 <http://www.linuxidc.com/Linux/2015-05/116903.htm>

Jenkins 的详细介绍: [请点这里](#)

Jenkins 的下载地址: [请点这里](#)

-----分割线-----

安装配置 `svn`

可以参考 <http://superleedo.blog.51cto.com/12164670/1888804>

`yum -y install subversion`

`mkdir /var/svn`

`svnadmin create /var/svn/project`

`svn import /etc/java file:///var/svn/project -m "java"`

启动 `svn`

`svnserve -d -r /var/svn/project`

用 windows 客户端检出 `svn://192.168.1.7/project` 测试是否成功

需要设置密码的请查看上面链接文档自行设置

使用 **Maven** 创建项目

通过 `maven` 创建 `helloworld` 项目

`cd /tmp/`

使用 `mvn` 命令创建

`mvn archetype:generate -DgroupId=helloworld -DartifactId=helloworld`

[INFO] Scanning for projects...

[INFO]

[INFO] -----

[INFO] Building Maven Stub Project (No POM) 1

[INFO] -----

.....信息多，略去.....

[INFO] Generating project in Interactive mode ##此处可能会卡住，可以选择一直等待 1 个小时左右就正常进行了，只要这次链接下载成功了，以后就不会出现这种情况了；也可以用重新创建时加入忽略跳过参数-DarchetypeCatalog=internal ##

.....

.....

Define value for property 'version': 1.0-SNAPSHOT: : ##执行到此处停止了，回车即可

[INFO] Using property: package = helloworld

Confirm properties configuration:

groupId: helloworld

artifactId: helloworld

version: 1.0-SNAPSHOT

package: helloworld

Y: : ##输入 Y，回车

。。。。下面等待直到成功

tree helloworld/

helloworld/

├── pom.xml

└── src

 ├── main

 └── java

 └── helloworld

 └── App.java

 └── test

 └── java

 └── helloworld

 └── AppTest.java

下面将创建的项目导入到 svn 版本库中/var/svn/project 中

svn import /tmp/helloworld file:///var/svn/project/ -m "helloworld"

3 配置 Jenkins

浏览器访问 jenkins <http://192.168.1.7:8080/> 进入 Jenkins 管理界面



Jenkins Web 的全局环境配置:

系统管理=>系统设置配置 jdk 和 mvn 的家目录, 顺便查看自己的 Jenkins 家目录是否改变为自己定义的



jenkins 的家目录如下图

Jenkins

配置

新建

用户

任务历史

项目关系

检查文件指纹

系统管理

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

主目录

/root/.jenkins

系统消息

执行者数量

2

标记

用法

尽可能的使用这个节点

生成前等待时间

5

SCM签出重试次数

0

☐ Restrict project naming

全局属性

☐ Environment variables

☐ Tool Locations

Maven Configuration

Default settings provider

Use default maven settings

jdk/mvn 的配置如下图：

配置

JDK

JDK 安装

JDK

别名

jdk

JAVA_HOME

/usr/local/jdk1.8.0_131

☐ 自动安装

新增 JDK

系统下JDK 安装列表

Ant

Ant 安装

新增 Ant

系统下Ant 安装列表

Maven

Maven 安装

Maven

Name

maven3.3.9

MAVEN_HOME

/usr/local/apache-maven-3.3.9

☐ 自动安装

新增 Maven

系统下Maven 安装列表

mvn 的仓库为了方便使用默认的：

Jenkins 填写 URL，SVN 选择自己安装的版本

Jenkins > 配置

Maven项目配置

全局MAVEN_OPTS

Local Maven Repository 默认即可

☒ Help make Jenkins better by sending anonymous usage statistics and crash reports to the Jenkins project.

Jenkins Location

Jenkins URL Jenkins访问路径

系统管理员邮件地址

SSH Server

SSHD Port ☐ 指定端口: ☒ 随机选取 ☐ 禁用

CVS

Default Compression Level

Private Key Location

Private Key Password

Known Hosts Location

Authentication

Subversion

Subversion Workspace Version svn版本

Exclusion revprop name

接下来创建 helloworld 项目任务

新建任务

Jenkins

Jenkins > All >

新建

用户

任务历史

项目关系

检查文件指纹

系统管理

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

Item名称

☐ 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东西。

☒ 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。

☐ External Job
这个类型的任务允许你记录执行在外部Jenkins的任务，任务甚至运行在远程机器上。这可以让Jenkins作为你所有参阅 [这个文档查看详细内容](#)。

☐ 构建一个多配置项目
适用于多配置项目，例如多环境测试，平台指定构建，等等。

☐ 复制已有的Item
要复制的任务名称

Jenkins

Jenkins > helloworld1 > 配置

返回面板
状态
修改记录
工作空间
立即构建
删除 Maven project
配置
模块

Build History 构建历史 =
RSS 全部 RSS 失败

项目名称 helloworld

描述 test测试

策略 ☒ 丢弃旧的构建

Log Rotation

保持构建的天数 10
如果非空，构建记录将保存此天数

保持构建的最大个数 10
如果非空，最多此数目的构建记录将被保存

☐ 参数化构建过程
☐ 关闭构建 (重新开启构建前不允许进行新的构建)
☐ 在必要的时候开发构建

高级项目选项

Jenkins > helloworld > 配置

高级...

源码管理

☐ None
☐ CVS
☐ CVS Projectset
☒ Subversion Modules

Repository URL svn://192.168.1.7

Local module directory (optional) .

Repository depth infinity

Ignore externals ☐

Add more locations...

Check-out Strategy Always check out a fresh copy
Delete everything first, then perform "svn checkout". While this takes time to execute, it ensures that the workspace is in the pristine state.

源码库浏览器 (自动)

pom.xml 文件，选择跳过测试(不跳过测试，耗时还可能报错):

添加内容如下 (clean install -Dmaven.test.skip=true)

Jenkins > helloworld > 配置

构建触发器

☒ Build whenever a SNAPSHOT dependency is built ?
☐ 触发远程构建 (例如,使用脚本) ?
☐ Build after other projects are built ?
☐ Build periodically ?
☐ Poll SCM ?

Pre Steps

Add pre-build step ▼

Build

Root POM ?
Goals and options ?
高级...

Post Steps

☐ Run only if build succeeds
☒ Run only if build succeeds or is unstable
☐ Run regardless of build result
Should the post-build steps run only for successful builds, etc.
Add post-build step ▼

先点击应用再点击保存

设置好任务后，新建工作空间，也就是执行编译打包，此过程时间会比较长，需要耐心等待



Jenkins

Jenkins > helloworld >

返回面板
状态
修改记录
工作空间
立即构建
删除 Maven project
配置
模块

Maven project helloworld

hellotest

 工作区
 最新修改

Build History

构建历史

#8	2017-4-30 下午10:47
----	-------------------

 [RSS 全部](#)  [RSS 失败](#)

相关连接

- [Last build\(#8\).2 小时 44 分之前](#)
- [Last stable build\(#8\).2 小时 44 分之前](#)
- [Last successful build\(#8\).2 小时 44 分之前](#)

构建完成后出现 jar 包，在后台工作空间中会多一个 target 目录

The screenshot shows the Jenkins web interface. At the top, the Jenkins logo and name are visible. Below the navigation bar, there's a sidebar on the left with links like '返回到工程', '状态集', '变更记录', 'Console Output', '编辑编译信息', '删除本次生成', 'Executed Mojos', 'See Fingerprints', and 'Redeploy Artifacts'. The main area displays the build title '构建 helloworld (2017-4-30 22:47:31)'. Below the title, there's a red box highlighting the '构建产生文件' (Build Artifacts) section. It lists two files: 'helloworld-1.0-SNAPSHOT.jar' (2.00 KB) and 'helloworld-1.0-SNAPSHOT.pom' (752 B), each with a 'view' link. Below the artifacts, there's a note '没有变更。' (No changes).

4 构建过程中常见错误及解决方法

svn 的 URL 配置问题，路径出现不正确或者提示要输入账号密码？？

SVN 的路径是项目的上一级目录，如果配置的 svn 有账号密码，就根据提示填入账号密码即可

找不到 pom.xml??

pom.xml 文件是必须的，maven 创建的项目都会有这个文件，其他项目支持 maven 且有这个文件才行

所有测试的时候我选择用 maven 命令创建一个测试项目，保证不出现 pom.xml 找不到的问题

maven 创建项目时在 generating project in interactive mode 卡住了？？

解决方法是可以选择一直等待就行了大概一个小时左右，只是首次会出现这个情况（推荐等待完成）

也可以在命令后加参数 -DarchetypeCatalog=internal

```
mvn archetype:generate -DgroupId=helloworld -DartifactId=helloworld -DarchetypeCatalog=internal
```

在执行创建工作空间时候，创建不成功，出现错误??

.....

```
java.lang.NoSuchFieldError: DEFAULT_USER_SETTINGS_FILE
```

.....

这是 jar 冲突，版本太高引起的，maven3.5.0 版本有这个问题，使用 3.3.9 低版本就可以了

总结：以上这些完成后，算是基础环境搭建已经成功完成，剩下的就是根据自己的项目实际情况深入学习配置使用 Jenkins 持续集成了，谷歌还有官网是解决各种问题的最好地方！

点击这里的链接进入精彩的[Linux公社](http://www.Linuxidc.com)网站

Linux公社（www.Linuxidc.com）于2006年9月25日注册并开通网站，Linux现在已经成为一种广受关注和支持的一种操作系统，IDC是互联网数据中心，LinuxIDC就是关于Linux的数据中心。

[Linux公社](http://www.Linuxidc.com)是专业的Linux系统门户网站，实时发布最新Linux资讯，包括Linux、Ubuntu、Fedora、RedHat、红旗Linux、Linux教程、Linux认证、SUSE Linux、Android、Oracle、Hadoop、CentOS、MySQL、Apache、Nginx、Tomcat、Python、Java、C语言、OpenStack、集群等技术。

Linux公社（LinuxIDC.com）设置了有一定影响力的Linux专题栏目。

Linux公社 主站网址：www.linuxidc.com 旗下网站：www.linuxidc.net

包括：[Ubuntu 专题](#) [Fedora 专题](#) [Android 专题](#) [Oracle 专题](#) [Hadoop 专题](#)
[RedHat 专题](#) [SUSE 专题](#) [红旗 Linux 专题](#) [CentOS 专题](#)



Linux 公社微信公众号：[linuxidc_com](#)



微信扫一扫

Linuxidc.com

订阅专业的最新Linux资讯及开源技术教程。

搜索微信公众号：[linuxidc_com](#)