

```
link: null
title: 珠峰架构师成长计划
description: 启动网站
keywords: null
author: null
date: null
publisher: 珠峰架构师成长计划
stats: paragraph=46 sentences=74, words=235
```

1. Jenkins

- 通用的开源平台
- 常用于自动化测试和持续集成

2. window安装 <#>

2.1 下载Jenkins

- [jenkins \(https://jenkins.io/\)](https://jenkins.io/)
- [jenkins.war \(http://ftp-chi.osuosl.org/pub/jenkins/war/2.173/jenkins.war\)](http://ftp-chi.osuosl.org/pub/jenkins/war/2.173/jenkins.war)

启动网站

```
java -jar jenkins.war
```

War文件（扩展名为.War,Web Application Archive）包含全部Web应用程序。在这种情形下，一个Web应用程序被定义为一组文件、类和资源，用户可以对jar文件进行封装，并把它作为小型服务器（servlet）访问。war是一个可以直接访问的web模块，通常用于测试，打成部署到容器中。以Tomcat来说，将war包放置至webapps目录下，然后启动Tomcat，这个包就会自动解压，就相当于发布了。war包是生成的一种Web应用程序格式，与jar类似，是很多文件的压缩包。war包中的文件按照一定目录结构来组织。根据其根目录下包含有html和jsp文件，或者包含这两种文件的目录，另外还有WEB-INF目录。WEB-INF目录下含有一个web.xml文件和一classes目录，web.xml是这个应用的配置文件，而classes目录下则包含编译好的servlet类sp，或者servlet所依赖的其他类（JAVABean）。通常这些所依赖的类也可以打包成web.xml下的lib目录下

2.2 安装Jenkins

- Rebuilder 重新构建
- Safe Restart 安全重启

2.3 权限配置

- 配置全局安全属性
- 添加用户
- C:\Users....jenkins\secrets\initialAdminPassword

```
/configureSecurity/
```

[illegible]

3. linux安装

- [jdk \(https://www.oracle.com/technetwork/java/javase/downloads/java-archive-javase10-4425482.html\)](https://www.oracle.com/technetwork/java/javase/downloads/java-archive-javase10-4425482.html)

3.1 安装JDK

3.1.1 下载JDK

```
cd /usr/local/src
wget http:
tar -xzf jdk1.8.0_211.tar.gz
cp -r /usr/local/src/jdk1.8.0_211 /usr/java
ln -s /usr/java/jdk1.8.0_211/bin/java /usr/bin/java
```

3.1.2 修改环境变量

修改配置文件 /etc/profile

```
JAVA_HOME=/usr/java/jdk1.8.0_211
export CLASSPATH=.:${JAVA_HOME}/jre/lib/rt.jar:${JAVA_HOME}/lib/dt.jar:${JAVA_HOME}/lib/tools.jar
export PATH=$PATH:${JAVA_HOME}/bin
```

```
source /etc/profile
java --version
```

3.2 安装git

- 服务器安装git
- 生成公钥和私钥
- 将公钥上传到github服务器，实现免密码联通

```
yum install -y git
git config --global user.name 'zhufengnodejs'
git config --global user.email 'zhufengnodejs@126.com'
ssh-keygen -t rsa -C 'zhufengnodejs@126.com'

ssh git@github.com
You've successfully authenticated
```

3.3 安装maven

使用 命令 创建Maven的普通Java项目 mvn archetype:create -DgroupId=packageName -DartifactId=projectName 创建Maven的Web项目 mvn archetype:create -DgroupId=packageName -DartifactId=webappName -DarchetypeArtifactId=maven-archetype-webapp 编译源代码 mvn compile 运行测试 mvn test 打war包 mvn package 清除产生的项目 mvn clean 上传到私服 mvn deploy

3.3.1 下载并解压

- 安装maven (<http://maven.apache.org/>)

```
cd /usr/local/src
wget apache-maven-3.6.1-bin.zip
unzip apache-maven-3.6.1-bin.zip
cd apache-maven-3.6.1
/usr/local/src/apache-maven-3.6.1
```

3.3.2 环境变量

vi /etc/profile

```
export MAVEN_HOME=/usr/local/src/apache-maven-3.6.1
export PATH=$PATH:${MAVEN_HOME}/bin
```

```
source /etc/profile
mvn -version
```

3.4 安装tomcat

- 下载tomcat
- 解压tomcat
- 配置执行权限和端口并关闭防火墙
- 验证安装是否正确

3.4.1 下载并tomcat

- [tomcat \(http://tomcat.apache.org/\)](http://tomcat.apache.org/)

```
wget http://tomcat.apache.org/tomcat-9.0.19/apache-tomcat-9.0.19.zip
unzip apache-tomcat-9.0.19.zip
```

3.4.2 配置tomcat

```
# 给当前路径和所有子路径的所有的文件增加可执行权限
chmod a+x -R *
```

/usr/local/src/apache-tomcat-9.0.19/conf/server.xml 启动tomcat

```
/usr/local/src/apache-tomcat-9.0.19/bin/startup.sh
```

3.4.3 关闭防火墙

功能 命令 停止防火墙 systemctl stop firewalld.service 永久关闭防火墙 systemctl disable firewalld.service