

1 下载

<https://jenkins.io/download/> 去找war包下载。（会有很多安装方式，这里以war包形式）

<http://mirrors.jenkins.io/war-stable/latest/jenkins.war>

2 部署

放到tomcat webapp下即可运行，注意要把war包重命名为ROOT.war

3 启动

启动tomcat即可

4 配置

第一次启动会出现下面界面，找到文件配置放入密码即可

入门

解锁jenkins

为了确保管理员安全地安装jenkins，密码已写入到日志中（[不知道在哪里？](#)）该文件在服务器上：

```
C:\Users\Administrator\.jenkins\secrets\initialAdminPassword
```

请从本地复制密码并粘贴到下面。

管理员密码

5 插件安装（这里没有选择，后面也可以通过配置在来添加插件）

第一次启动会出现进行选择

自定义jenkins

插件通过附加特性来扩展jenkins以满足不同的需求。

安装推荐的插件

安装jenkins社区推荐的插件。

选择插件来安装

选择并安装最适合的插件。

这里选择推荐的插件进行安装。

✓ Folders	✓ OWASP Markup Formatter	✓ Build Timeout	✓ Credentials Binding
✓ 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

6 添加管理员账号

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:

这里全部写的root

7 进行配置 -- 全局工具配置



全局工具配置

工具配置，包括它们的位置和自动安装器

1) jdk配置

把自动安装复选框去掉，配置机器上面的jdk

JDK

别名

JAVA_HOME

☐ 自动安装

如果这里的 JAVA_HOME不正确，当 JAVA_HOME后面的文本框失去焦点就会报错。

JDK

别名

JAVA_HOME

⚠ D:\jdk1.8.0_151a is not a directory on the Jenkins master (but perhaps it exists on some agents)

2) Git配置

Git

Name

git

Path to Git executable

D:\Git\bin\git.exe

☐ 自动安装

3) Maven配置

Maven

Name

maven-3.5.2

MAVEN_HOME

D:\apache-maven-3.5.2

☐ 自动安装

配置完成之后 保存。

8 插件追加

Maven Integration(Maven集成，否则新建的时候没有Maven Project)

[Publish Over SSH](#) (ssh 文件传输，将打包的文件传输到项目服务器)

Subversion Plug-in插件(svn插件)

Deploy to container Plugin插件

配置 publish Over ssh：系统管理--系统配置中

SSH Server

Name

aliyun

Hostname

120.79.183.246

Username

root

Remote Directory

/

☒ Use password authentication, or use a different key

Passphrase / Password

.....

依次配置

远程名称 - 随意些

host地址

登录用户名

远程目录 - 需要该用户具有该目录权限，root用户直接输入根目录 / 就好

登录密码 - 需要勾选复选框 (use password authentication, or use a different key)

9 新建一个任务 -- 构建一个Maven项目

输入一个任务名称

first

» 必填项

 **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

 **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

 **流水线**
精心地组织一个可以长期运行在多个节点上的任务。适

 **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.

 **GitHub Organization**
Scans a GitHub organization (or user account) for all re

 **Multibranch Pipeline**
Creates a set of Pipeline projects according to detected

 **文件夹**
创建一个可以嵌套存储的容器。利用它可以进行分组。并

确定

配置任务：
配置git

Git

Repositories

Repository URL

配置maven 打包命令

Build

Root POM

Goals and options

配置打包完成之后，传输到服务器tomcat 下，并且在传输完成之后执行命令

如下图 构建后操作（**Post-build Actions**） ——》 Send build artifacts over SSH



1 图中红色内容 是警告 不是错误 。具体可用看帮助

依次填写

服务器名称 - 下拉选择，在 publish Over ssh插件中配置

source files —— **/ROOT.war （需要传输的文件筛选）

remove prefix —— webadmin/target/（war的目录前缀）

remote dictionary —— /usr/local/apache-tomcat-8.5.30/webapps （要保存在远程机器的目录地址）

exec command —— /usr/local/apache-tomcat-8.5.30/bin/restart.sh（在远程机器上要执行的脚本）

2 restart.sh 内容如下（删除旧的项目文件，重启tomcat）：

```
1 ## java env
2 export JAVA_HOME=/usr/local/jdk1.8.0_161
3 export JRE_HOME=$JAVA_HOME/jre
4
5 ## restart tomcat
6 /usr/local/apache-tomcat-8.5.30/bin/shutdown.sh
7 sleep 3
8 ## 删除root目录，留下的是新发布的root包
9 rm -rf /usr/local/apache-tomcat-8.5.30/bin/webapps/ROOT
10 /usr/local/apache-tomcat-8.5.30/bin/startup.sh
```

3 Jenkins 打包之后的文件存储路径 Jenkins 工作空间/任务名称 开始

项目的目录结果大致如下

```
1 parent
2 | pom.xml
3 |
4 |---common
5 | | pom.xml
6 | |
7 | |---src
8 | | |---main
9 | | | |---java
10 | | | |
11 | | | |---resources
12 | | |
13 | | |---test
14 | |
15 | |---target
16 | | | common-0.0.1.jar
17 |
18 |---repo
```

```

19 | | pom.xml
20 | |
21 | |─src
22 | | |─main
23 | | | |
24 | | | |─resources
25 | | | |
26 | | |─test
27 | |
28 | |─target
29 | | | repo-0.0.1.jar
30 |
31 |─service
32 | | pom.xml
33 | |
34 | |─src
35 | | |─main
36 | | | |─java
37 | | | | |
38 | | | | |─resources
39 | | | | | | servicebase.conf
40 | | | | | |
41 | | | | | |─dev
42 | | | | | | | service.conf
43 | | | | | |
44 | | | | | |─test
45 | | | | | | | service.conf
46 | | | | |
47 | | | |─test
48 | | |
49 | | |─target
50 | | | | service-0.0.1.jar
51 |
52 |─webadmin
53 | | pom.xml
54 | |
55 | |─src
56 | | |─main
57 | | | |─java
58 | | | | |
59 | | | |─resources
60 | | | | | application.properties
61 | | | | |
62 | | | | |─dev
63 | | | | | | webconfig.properties
64 | | | | | |
65 | | | | | |─test
66 | | | | | | | webconfig.properties
67 | | | |─webapp
68 | | |─test
69 | | | |─java
70 | |
71 | |─target
72 | | | ROOT.war

```

```
73 | webadmin-0.0.1.war
74
```

到这里就完成了重头开始安装Jenkins， 和新建一个maven项目任务， 从 git上面下载源码进行打包并且自动发布的流程。