# Jenkins部署文档

1. 预置环境：
2. yum install -y wget vim lsof ntp tree psmisc net-tools gcc gcc-c++ zip unzip pcre pcre-devel zlib zlib-devel openssl openssl-devel lrzsz

2、Tcp系统优化

[root@localhost test]#vi /etc/sysctl.conf

fs.file-max = 65535

net.ipv4.ip\_forward = 1

net.ipv4.tcp\_fin\_timeout = 200

net.ipv4.tcp\_max\_syn\_backlog = 10240

net.ipv4.tcp\_keepalive\_time = 1200

net.ipv4.tcp\_synack\_retries = 3

net.ipv4.tcp\_syn\_retries = 3

net.ipv4.tcp\_max\_orphans = 8192

net.ipv4.tcp\_max\_tw\_buckets = 5000

net.ipv4.tcp\_window\_scaling = 0

net.ipv4.tcp\_sack = 0

net.ipv4.tcp\_timestamps = 0

net.ipv4.tcp\_syncookies = 1

net.ipv4.tcp\_tw\_reuse = 1

net.ipv4.tcp\_tw\_recycle = 1

net.ipv4.ip\_local\_port\_range = 1024 65000

net.ipv4.icmp\_echo\_ignore\_all = 1

net.core.somaxconn= 1024

vm.overcommit\_memory=1

[root@localhost test]#sysctl -p

3、Linux增加文件句柄打开文件数量

[root@localhost test]# vim /etc/security/limits.conf

\* soft nproc 65535

\* hard nproc 65535

\* soft nofile 65535

\* hard nofile 65535

4、Linux检查配置项（查看是否存在）

[root@localhost test]# vi /etc/rc.local

touch /var/lock/subsys/local

[root@localhost test]# vim /etc/sysconfig/network-scripts/ifcfg-em1

ONBOOT=yes

1. SSH连接白名单+禁root登录

[root@localhost test]#vi /etc/hosts.allow

#添加ip注意是整个站的ip或者ip段+内网IP段

sshd:103.232.87.:allow

sshd:113.61.35.:allow

sshd:xxx.xxx.xxx.xxx:allow

[root@localhost test]#vi /etc/hosts.deny

sshd:all:deny

[root@localhost test]#service sshd restart

[root@localhost test]#groupadd will

[root@localhost test]#useradd -g will will

[root@localhost test]#passwd will

Password

Password

[root@localhost test]#vi /etc/ssh/sshd\_config

PermitRootLogin no

[root@localhost test]#service sshd restart

改完之后不要退出 在线测试一下是否禁ping，是否对应主机才可以登陆，是否不可以用root,因为如果你退出了，但是你的配置错误的比如白名单，这时候你自己都不能登陆，麻烦了。

6、全局变量优化

[root@localhost test]#vi /etc/profile ###在末尾加上请注意

########################################################

export HISTTIMEFORMAT="%F %T "

export TMOUT=600

#########################################################

export JAVA\_HOME=/opt/pub/apps/jdk

export JRE\_HOME=$JAVA\_HOME/jre

export CLASSPATH=$JAVA\_HOME/lib:$JAVA\_HOME/lib/tools.jar

export PATH=$PATH:$JAVA\_HOME/bin:$JAVA\_HOME/jre/bin

##########################################################

MAVEN\_HOME=/opt/pub/apps/maven

export MAVEN\_HOME

export PATH=${PATH}:${MAVEN\_HOME}/bin

##########################################################

[root@localhost test]#source /etc/profile

1. 时间的定时同步

timedatectl set-timezone Asia/Shanghai

[root@localhost test]#crontab -e

#ntpdate /h

0 \*/1 \* \* \* /usr/sbin/ntpdate -u time.nist.gov >>/var/log/ntpdate.log 2>&1

1. 目录结构的建立

[root@localhost test]#cd /

[root@localhost test]#mkdir jenkins pub

[root@localhost test]#cd pub && mkdir apps logs src

#刷新权限由低级用户运行程序

[root@localhost test]#chown -R will:will /opt/\*

1. 防火墙selinux

[root@localhost test]#setenforce 0

[root@localhost test]#vim /etc/selinux/config

SELINUX=disabled

#检查防火墙状态

[root@localhost test]#systemctl status firewalld.service

[root@localhost test]#systemctl list-unit-files |grep firewalld

#开机启动

[root@localhost test]#systemctl start firewalld.service

[root@localhost test]#systemctl enable firewalld.service

#转发端口

80>8000端口

[root@localhost test]#

firewall-cmd --add-forward-port=port=80:proto=tcp:toport=8000 --permanent [root@localhost test]#firewall-cmd --reload

1. 安装必要环境：

2.1、安装jdk

从svn拷贝jdk安装包

[root@localhost test]#cd /opt/pub/src

[root@localhost test]#tar -zxvf jdk-8u121.tar.gz -C /opt/pub/apps (注意，这里的路径一定要与之前设置环境变量的路径一致, 安装好后输入下“java -version”测试下是否安装成功)

2.2、安装tomcat安装包并解压

[root@localhost test]#tar -zxvf tomcat.xx.tar.gz -C /opt/Jenkins/apps

删除包

[will@localhost ~]rm -rf tomcat.xx.tar.gz

进入tomcat目录

[will@localhost ~]cd /opt/Jenkins/apps/tomcat\_17001

2.3、tomcat配置

# 修改catalina.sh

[will@localhost apps]$tomcat\_17001/bin

[will@localhost bin]$vim catalina.sh

if [ -z "$CATALINA\_OUT" ] ; then

CATALINA\_OUT= /opt/Jenkins/logs/tomcat\_17001/catalina.out

#保存退出 :wq

# 修改logging.properties、server.xml、context.xml

[will@localhost apps]$ cd tomcat\_17001/conf

[will@localhost conf]$ vim logging.properties

1catalina.org.apache.juli.AsyncFileHandler.level = FINE

1catalina.org.apache.juli.AsyncFileHandler.directory = /opt/Jenkins/logs/tomcat\_17001

1catalina.org.apache.juli.AsyncFileHandler.prefix = catalina.

2localhost.org.apache.juli.AsyncFileHandler.level = FINE

2localhost.org.apache.juli.AsyncFileHandler.directory = /opt/Jenkins/logs/tomcat\_17001

2localhost.org.apache.juli.AsyncFileHandler.prefix = localhost.

3manager.org.apache.juli.AsyncFileHandler.level = FINE

3manager.org.apache.juli.AsyncFileHandler.directory = /opt/Jenkins/logs/tomcat\_17001

3manager.org.apache.juli.AsyncFileHandler.prefix = manager.

4host-manager.org.apache.juli.AsyncFileHandler.level = FINE

4host-manager.org.apache.juli.AsyncFileHandler.directory = /opt/Jenkins/logs/tomcat\_17001

4host-manager.org.apache.juli.AsyncFileHandler.prefix = host-manager.

#保存退出 :wq

[will@localhost ~]vim server.xml

<Server port="17005" shutdown="SHUTDOWN">

<Connector port="17001" protocol="HTTP/1.1"

<!-- Define an AJP 1.3 Connector on port 8009 -->

<Connector port="17009" protocol="AJP/1.3" redirectPort="8443" />

<Valve className="org.apache.catalina.valves.AccessLogValve" directory=" /opt/Jenkins/logs/tomcat\_17001"

prefix="localhost\_access\_log" suffix=".txt"

pattern="%h %l %u %t &quot;%r&quot; %s %b" />

<Context path="/" docBase=" /opt/Jenkins/webapps/jenkins\_17001/jenkins.war " reloadable="true"></Context>

#保存退出 :wq

[will@localhost ~]vim context.xml

<!-- The contents of this file will be loaded for each web application -->

<Context privileged="true">

#保存退出 :wq

2.4、启动tomcat，并把jenkins.war包放入相应的目录下

[will@localhost /]$ /opt/Jenkins/apps/tomcat\_17001/bin/startup.sh

Using CATALINA\_BASE: /opt/Jenkins/apps/tomcat\_17001

Using CATALINA\_HOME: /opt/Jenkins/apps/tomcat\_17001

Using CATALINA\_TMPDIR: /opt/Jenkins/apps/tomcat\_17001/temp

Using JRE\_HOME: /opt/pub/apps/jdk/jre

Using

CLASSPATH: /opt/Jenkins/apps/tomcat\_17001/bin/bootstrap.jar:/opt/Jenkins/apps/tomcat\_17001/bin/tomcat-juli.jar

Tomcat started.

3.1、安装Maven

[will@localhost apps]$tar -zxvf maven.tar.gz -C /opt/pub/apps (注意，这里的路径一定要与之前设置环境变量的路径一致,安装好后输入下“mvn -v”测试下是否安装成功)

4、nginx安装及配置

4.1、nginx安装

[will@localhost src]$$su will (切换的will用户)

[will@localhost src]$cd /opt/pub/src

####下载nginx

[will@localhost src]$wget http://XXXXXX/nginx-1.10.3.tar.gz

下载geoIP需要的库

[will@localhost src]$

wget http://geolite.maxmind.com/download/geoip/database/GeoLiteCity.dat.gz

[will@localhost src]$

wget http://geolite.maxmind.com/download/geoip/database/GeoLiteCountry/GeoIP.dat.gz

[will@localhost src]$

wget http://geolite.maxmind.com/download/geoip/api/c/GeoIP.tar.gz

[will@localhost src]$

tar -zxvf GeoIP.tar.gz -C /opt/pub/apps && cd /opt/pub/apps/GeoIP-1.4.8

[will@localhost GeoIP-1.4.8]$su root （切换root用户安装GeoIP）

[root@localhost GeoIP-1.4.8]#./configure && make && make install

#######返回上级目录安装nginx

[root@localhost GeoIP-1.4.8]#su will

[will@localhost GeoIP-1.4.8]$ cd ../../src

[will@localhost nginx-1.10.3]$tar -zxvf nginx-1.10.3.tar.gz -C /opt/pub/apps/ && cd /opt/pub/apps/nginx-1.10.3

[will@localhost nginx-1.10.3]$

./configure --prefix=/opt/pub/apps/nginx --with-http\_stub\_status\_module --with-http\_ssl\_module --with-http\_geoip\_module --with-http\_gzip\_static\_module --with-ipv6

[will@localhost nginx-1.10.3]$make && make install

[will@localhost nginx-1.10.3]$cd ../ && rm -rf nginx-1.10.3

#####加压两个国家城市的访问二进制库

[will@localhost pub]$ cd src

[will@localhost src]$ gunzip GeoLiteCity.dat.gz && gunzip GeoIP.dat.gz

###移动两个库到nginx下

[will@localhost src]$ mv \*.dat /opt/apps/nginx/conf

4.2、nginx配置文件

#进入nginx配置文件目录

[will@localhost src]$ cd /opt/pub/apps/nginx/conf

#创建目录keys、vhosts

[will@localhost conf]$ mkdir keys vhosts

#从svn里拷贝START-kfcp8888-com-PrivateKEY.key、START-kfcp8888-com-certificate-with-chain.crt到/opt/pub/apps/nginx/conf/keys中

#从svn里拷贝ip.conf到/opt/pub/apps/nginx/conf中

scp ip.conf will@103.244.0.70:/opt/pub/apps/nginx/conf

#删除nginx.conf

[will@localhost conf]$ rm -rf nginx.conf

#从svn里拷贝nginx.conf到本机

scp nginx.conf will@103.244.0.70:/opt/pub/apps/nginx/conf

#修改nginx.conf

[will@localhost conf]$vim nginx.conf （根据实际情况修改红色部分的路径）

error\_log /opt/pub/logs/nginx/error.log error;

pid /opt/pub/logs/nginx/nginx.pid;

#######

geoip\_country /opt/pub/apps/nginx/conf/GeoIP.dat;

geoip\_city /opt/pub/apps/nginx/conf/GeoLiteCity.dat;

#########################

include vhosts/\*.conf;

include vhosts/jenkins/\*.conf;

#保存退出 :wq

#进入vhosts目录

[will@localhost conf]$ cd vhosts

[will@localhost conf]$ mkdir jenkins

#从svn里拷贝配置文件到vhosts目录

scp http\_default\_server.conf https\_default\_server.conf [will@103.244.0.70:/opt/pub/apps/nginx/conf/vhosts](mailto:will@103.244.0.70:/opt/pub/apps/nginx/conf/vhosts)

#根据实际情况修改配置文件内容（以8hcp\_web.conf为例演示,其他可参考）

[will@localhost vhosts]$ cd jenkins

scp http\_default\_server.conf will@103.244.0.70:/opt/download/web/nginx\_conf/vhosts/ Jenkins

[will@localhost jenkins]$ mv web.conf jenkins\_web.conf

[will@localhost ~] vim jenkins\_web.conf

upstream tomcat\_jenkins\_web {

server 127.0.0.1:17001 weight=1;

}

server {

listen 8000;

listen 1443 ssl;

server\_name jenkins.zhushuqt.com;

index index.html index.htm;

root /opt/Jenkins/webapps/jenkins\_17001;

######

location ~ .\*\.(js|css|ico|png|jpg|eot|svg|ttf|woff)$ {

root /opt/Jenkins/webapps/jenkins\_17001;

#expires定义用户浏览器缓存的时间为3天，如果静态页面不常更新，可以设置更长，这样可以节

省带宽和缓解服务器的压力.

# expires 3d;

}

####

location ~ .\*$ {

proxy\_pass [http://tomcat\_jenkins\_web;](http://tomcat_xcp_web;)

access\_log /opt/Jenkins/logs/nginx/web\_access.log main;

error\_log /opt/Jenkins/logs/nginx/web\_error.log error;

#保存退出 :wq

# 创建配置文件中所需目录

[will@localhost ~]cd /opt/Jenkins/logs

[will@localhost ~]mkdir nginx

[will@localhost ~]cd /opt/pub/logs

[will@localhost ~]mkdir nginx

#检查nginx配置文件语法是否正确

[will@localhost ~]/opt/pub/apps/nginx/sbin/nginx -t

nginx: the configuration file /opt/pub/apps/nginx/conf/nginx.conf syntax is ok

nginx: configuration file /opt/pub/apps/nginx/conf/nginx.conf test is successful

#启动nginx

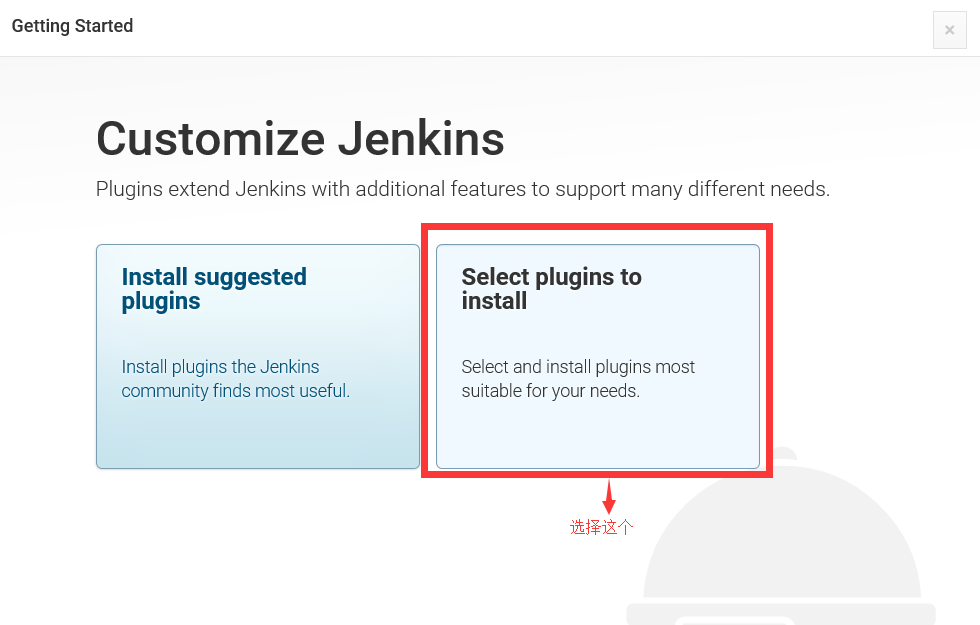
[will@localhost ~]/opt/pub/apps/nginx/sbin/nginx

三、配置Jenkins：

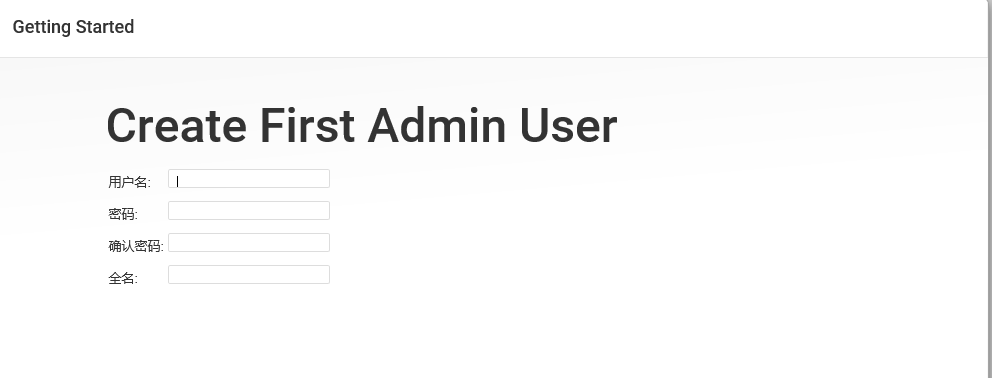
1.1、打开浏览器输入配置的地址如（<http://192.168.189.139:8080/jenkins>）



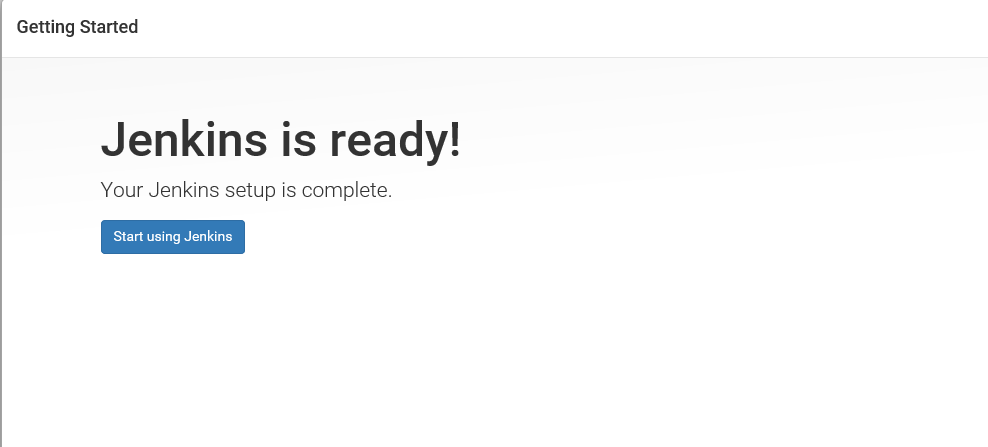
1．2、选择需要安装的插件（Subversion Plug-in、Folders、Publish over SSH），显示如图：插件：

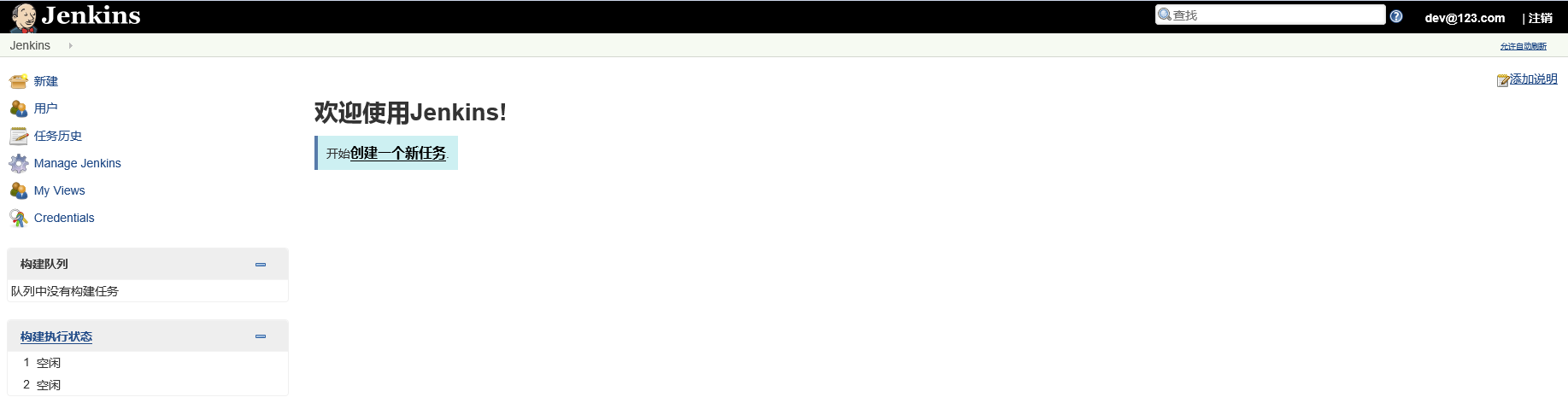


1.3、输入需要配置的帐号和密码并保存，如图：

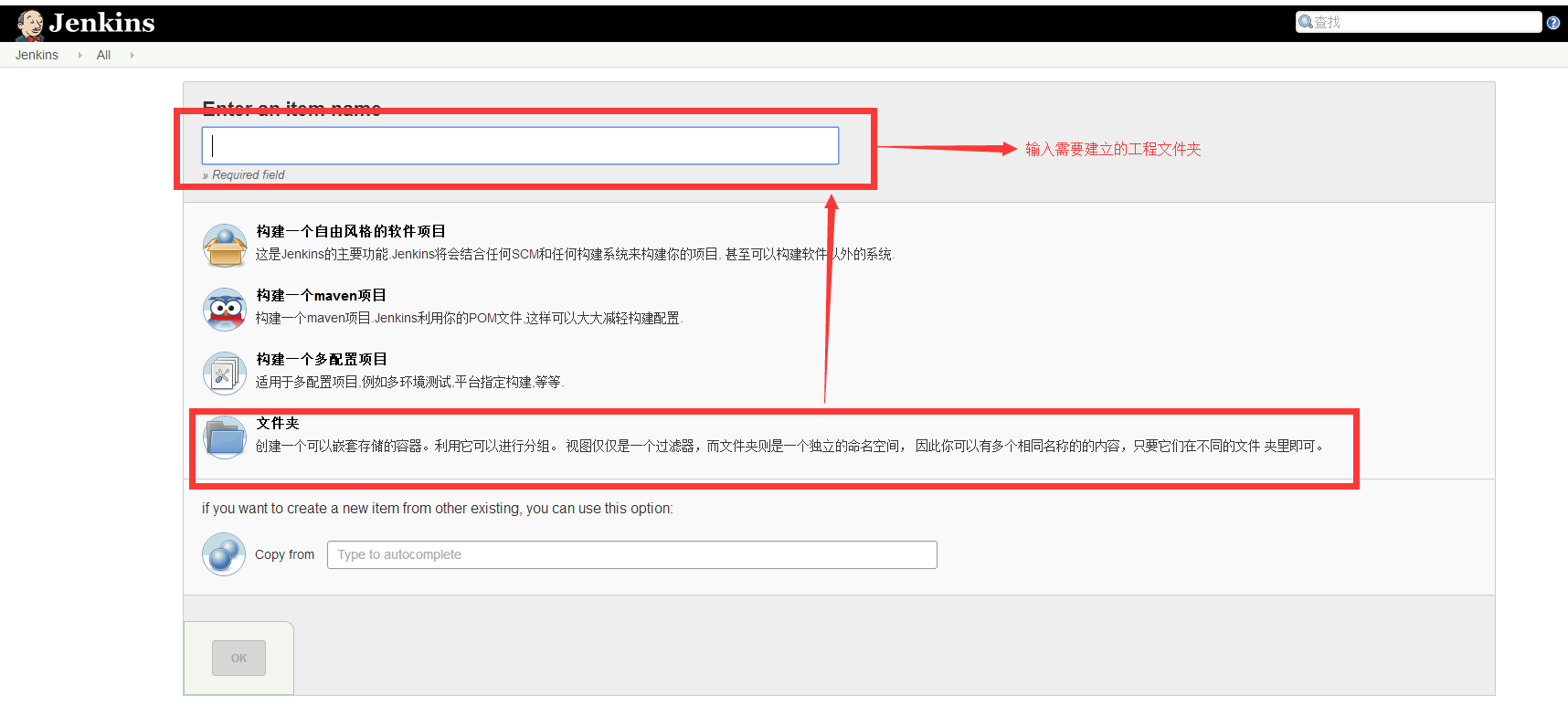


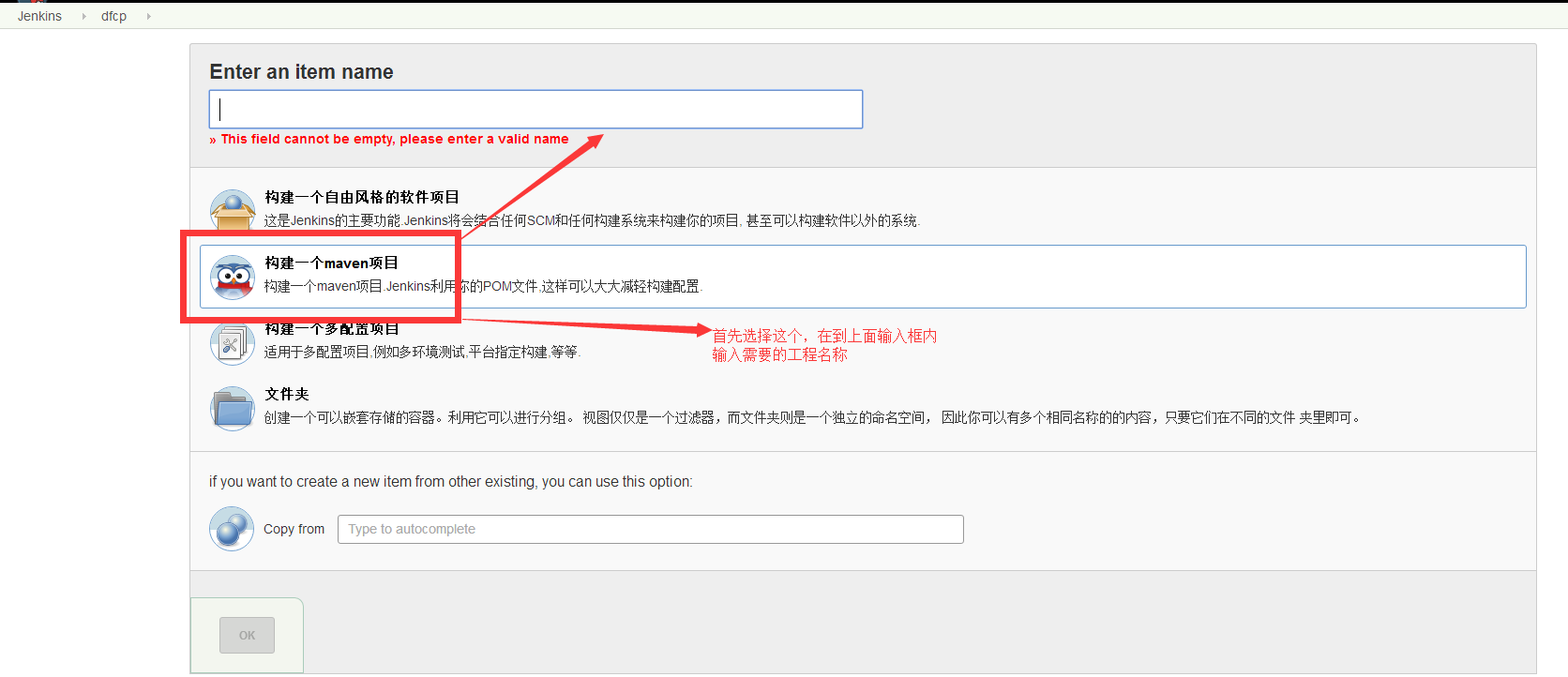
1.4点击“Start using Jenkins”按钮，如图：



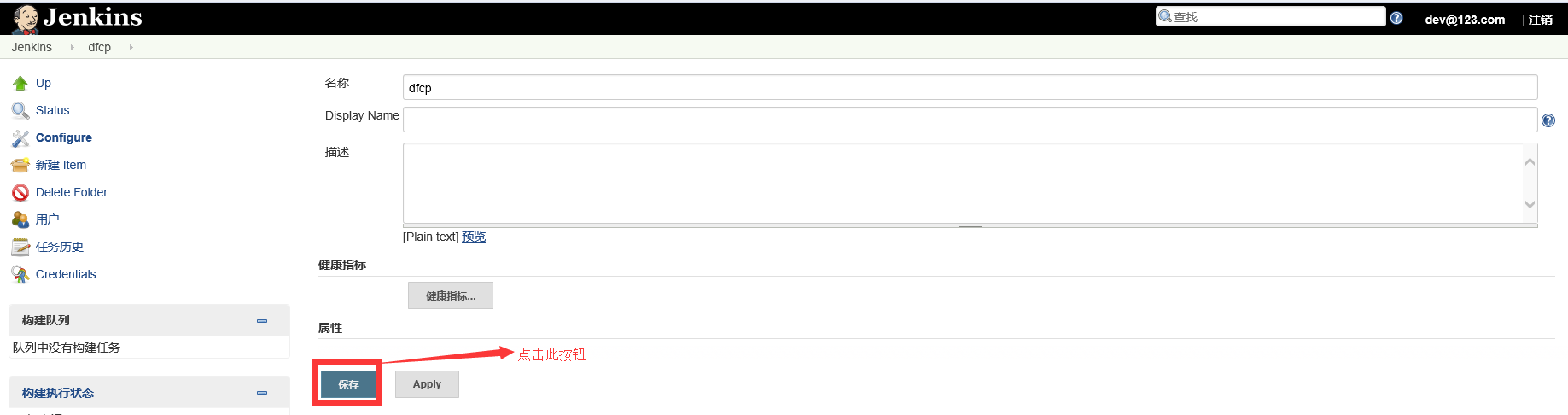


四、新建工程文件夹

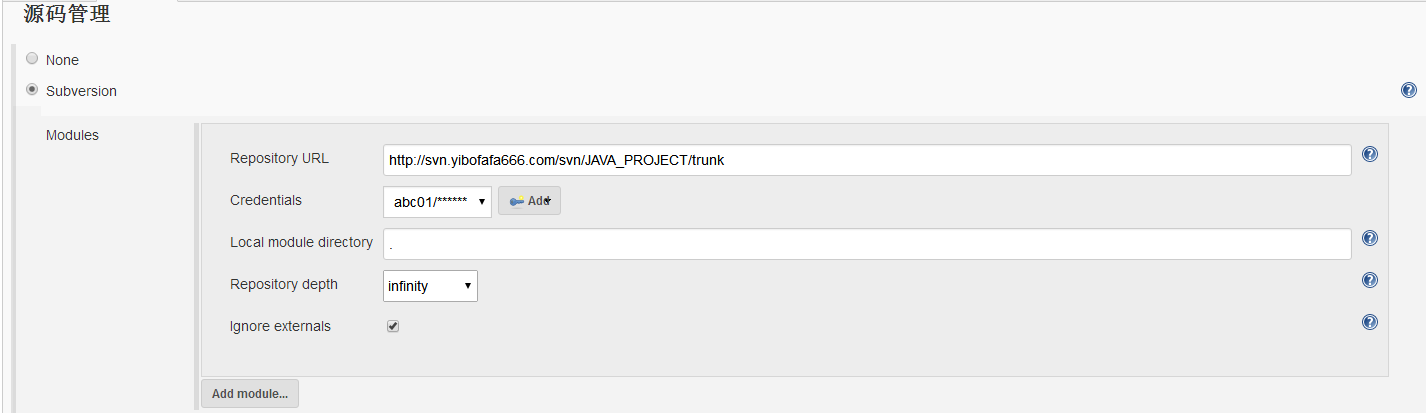


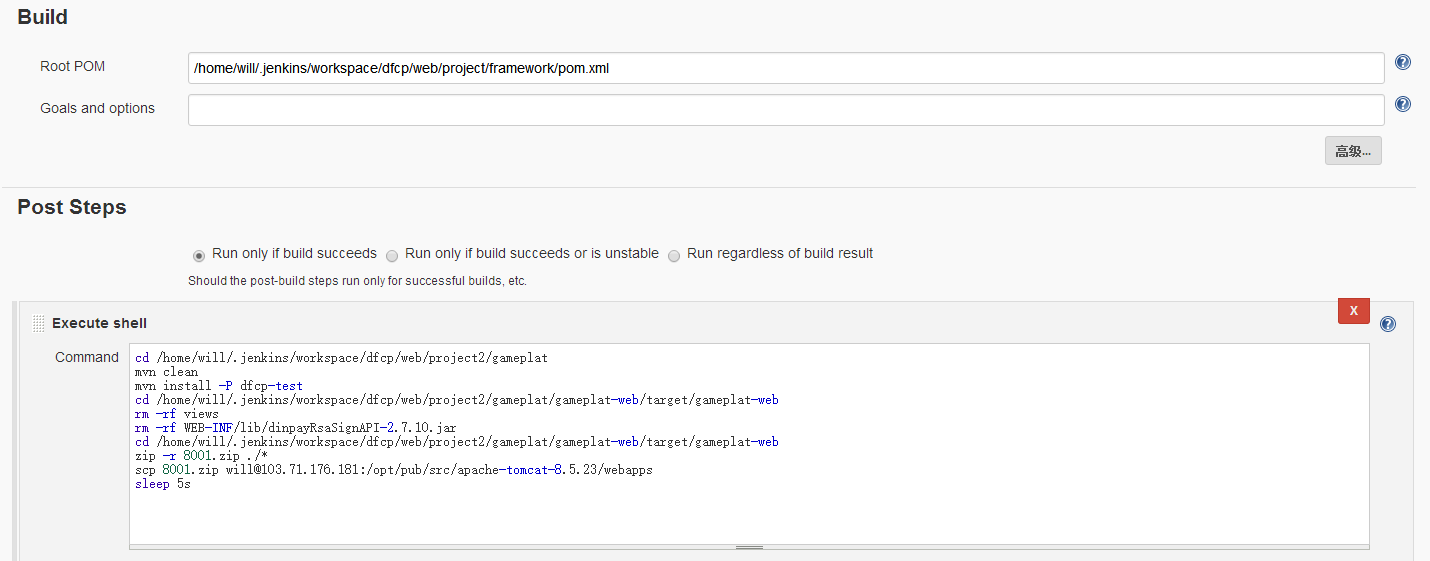


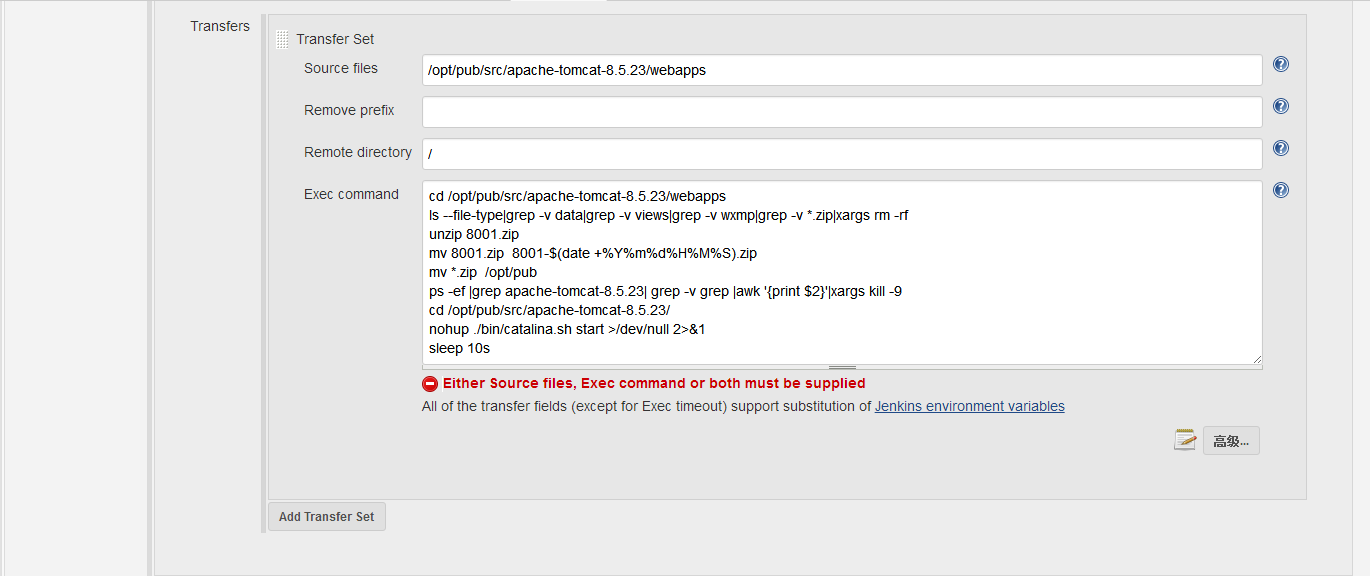




配置SVN地址和密码：







四、发布版本