## 1.Linux安装jdk1.8

1. 查看是否安装了jdk

rpm -qa | grep java

2. 下载并上传jdk压缩包到Linux上

通过xshell连接Linux，xftp上传jdk压缩包

3. 将jdk安装到 /usr/java 目录下

cd /usr

mkdir java

//将压缩包复制到java目录下

cp /home/jdk-8u151-linux-x64.tar.gz /usr/java/

//解压jdk到当前目录下

tar -zxvf jdk-8u151-linux-x64.tar.gz

4. 修改环境变量

cd /etc/

vi profile

//添加一下内容

export JAVA\_HOME=/usr/java/jdk1.8.0\_151

export JRE\_HOME=${JAVA\_HOME}/jre

export CLASSPATH=.:${JAVA\_HOME}/lib:${JRE\_HOME}/lib:$CLASSPATH

export JAVA\_PATH=${JAVA\_HOME}/bin:${JRE\_HOME}/bin

export PATH=$PATH:${JAVA\_PATH}

5. 重启生效

source /etc/profile

6. 验证安装是否成功

java -version

## 2. 安装tomcat

1. 下载并上传tomcat安装包到 /home目录下

2. 将tomcat安装到/usr/tomcat目录下

cd /usr

mkdir tomcat

cp /home/apache-tomcat-8.5.24.tar.gz /usr/tomcat/

tar -zxvf apache-tomcat-8.5.24.tar.gz

//设置软连接

ln -s /usr/tomcat/apache-tomcat-8.5.24 /usr/tomcat8

3. 启动tomcat

cd /usr/tomcat/apache-tomcat-8.5.24/bin/

./start.sh

//关闭tomcat

./shutdown.sh

4. 访问tomcat

<http://65.49.215.42:8080/>

5. 防火墙设置

//增加8080端口到防火墙配置中

/sbin/iptables -I INPUT -p tcp --dport 8080 -j ACCEPT

//保存配置

/etc/rc.d/init.d/iptables save

//重启防火墙

/etc/rc.d/init.d/iptables restart

6. 修改tomcat的监听端口为8081

cd /usr/tomcat/apache-tomcat-8.5.24/conf/

vi setting.xml

<Connector port="8081" protocol="HTTP/1.1"

connectionTimeout="20000"

redirectPort="8443" />

## 3. 安装MySQL

1. 查看是否已经安装了MySQL

rpm -qa | grep mysql

2. 卸载MySQL

rpm -e --nodeps mysql-libs-5.1.73-7.el6.x86\_64

3. 查看可安装的版本

yum list | grep mysql

4. 安装

yum install -y mysql-server mysql mysql-devel

5. 查看MySQL版本信息

rpm -qi mysql-server

6. MySQL初始化及配置

service mysqld start

7. 检查并设置MySQL开机启动

chkconfig –list | grep mysqld

chkconfig mysqld on

8. 为MySQL的root账号设置密码root

/usr/bin/mysqladmin -u root password ‘root’

//修改root账号密码

mysqladmin -u root -p password 'new-password'

9. 防火墙设置

//增加3306端口到防火墙配置中

/sbin/iptables -I INPUT -p tcp --dport 3306 -j ACCEPT

//保存配置

/etc/rc.d/init.d/iptables save

//重启防火墙

/etc/rc.d/init.d/iptables restart

10. MySQL远程登陆权限设置

登陆MySQL后

grant all privileges on \*.\* to 'root' @'%' identified by 'root';

flush privileges

## 4. 安装redis

1. 下载并上传redis安装包到Linux上

2. 安装解压redis安装包到/usr/redis目录下

cp /home/redis-3.2.11.tar.gz /usr/redis/

tar -zxvf redis-3.2.11.tar.gz

3. 编译redis

cd redis-3.2.11

make

4. 编译成功后，进入src文件夹，执行make install进行Redis安装

make install

5. redis 部署

//为了方便管理，将redis文件中的conf配置文件和常用命令移到统一文件中

//在redis目录下新建两个目录

mkdir bin

mkdir etc

//移动文件

mv redis.conf etc/

mv mkreleasehdr.sh redis-benchmark redis-check-aof redis-check-dump redis-cli redis-server /usr/redis/redis-3.2.11/bin

6. 后台启动redis服务

//首先编辑conf文件，将daemonize属性改为yes（表明需要在后台运行）

cd etc

vi redis.conf

7. 再次启动redis

redis-server /usr/redis/redis-3.2.11/etc/redis.conf

./redis-cli

8. 查看redis端口是否被占用

netstat -tunpl | grep 6379

9. 关闭redis

redis-cli shutdown

10. 设置按照service的形式来启动redis

执行命令：vi /etc/init.d/redis，创建脚本文件，脚本代码如下：

# Date 2017-12-15

# chkconfig: 2345 10 90

# description: Start and Stop redis

REDISPORT=6379

EXEC=/usr/redis/redis-3.2.11/bin/redis-server

REDIS\_CLI=/usr/redis/redis-3.2.11/bin/redis-cli

PIDFILE=/var/run/redis.pid

CONF="/usr/redis/redis-3.2.11/etc/redis.conf"

case "$1" in

start)

if [ -f $PIDFILE ]

then

echo "$PIDFILE exists, process is already running or crashed."

else

echo "Starting Redis server..."

$EXEC $CONF

fi

if [ "$?"="0" ]

then

echo "Redis is running..."

fi

;;

stop)

if [-f $PIDFILE ]

then

echo "$PIDFILE exists, process is not running."

else

PID=$(cat $PIDFILE)

echo "Stopping..."

$REDIS\_CLI -p $REDISPORT SHUTDOWN

sleep 2

while [ -x $PIDFILE ]

do

echo "Waiting for Redis to shutdown..."

sleep 1

done

echo "Redis stopped"

fi

;;

restart|force-reload)

${0} stop

${0} start

;;

\*)

echo "Usage: /etc/init.d/redis {start|stop|restart|force-reload}" >&2

exit 1

esac

//给文件添加权限

chmod 755 /etc/init.d/redis

11. 设置开机自启

chkconfig --add redis

chkconfig redis on

12. 将Redis的命令所在目录添加到系统参数PATH中

vi /etc/profile

添加： export PATH="$PATH:/usr/redis/redis-3.2.11/bin"

重启生效 source /etc/profile

13. 安装成功，可以通过直接使用 redis-cli 调用redis了

## 5. 安装nginx

安装nginx之前需要先安装openssl、zlib、pcre

SSL功能需要openssl库，下载地址：http://www.openssl.org/

　　　　gzip模块需要zlib库，下载地址：http://www.zlib.net/

　　　　rewrite模块需要pcre库，下载地址：http://www.pcre.org/

1. 安装openssl

cp /home/openssl-1.0.2n.tar.gz /usr/software/

tar -zxvf openssl-1.0.2n.tar.gz

cd openssl-1.0.2n

./config

make

make install

1. 安装zlib

cp /home/zlib-1.2.11.tar.gz /usr/software/

tar -zxvf zlib-1.2.11

cd zlib-1.2.11

./ configure

make

make install

1. 安装pcre

cp /home/pcre-8.41.tar.gz /usr/software/

tar -zxvf pcre-8.41.tar.gz

cd pcre-8.41

./ configure

make

make install

若出现 error: You need a C++ compiler for C++ support，问题 需要安装c++信息

yum install -y gcc gcc-c++

1. 安装nginx

cp /home/nginx-1.12.2.tar.gz /usr/nginx/

tar -zxvf nginx-1.12.2.tar.gz

cd nginx-1.12.2

./configure --with-pcre=../../software/pcre-8.41/ --with-zlib=../../software/zlib-1.2.11 --with-openssl=../../software/openssl-1.0.2n/

make

make install

In file included from src/core/ngx\_core.h:71,

from src/core/nginx.c:9:

src/core/ngx\_regex.h:15:18: error: pcre.h: No such file or directory

In file included from src/core/ngx\_core.h:71,

from src/core/nginx.c:9:

src/core/ngx\_regex.h:24: error: expected specifier-qualifier-list before ‘pcre’

make[1]: \*\*\* [objs/src/core/nginx.o] Error 1

make[1]: Leaving directory `/usr/nginx/nginx-1.12.2'

make: \*\*\* [build] Error 2

以上问题是因为pcre需要依赖pcre-devel，需要先安装这个库，安装如下：

yum install pcre-devel

1. 查看nginx安装路径

wheris nginx

1. 启动nginx

cd /usr/local/nginx/sbin/

./nginx -t

//启动 nginx 在/usr/local/nginx/sbin/ 路径下

./nginx

//查看nginx端口

netstat -ntlp

1. 修改nginx端口

cd /usr/local/nginx/conf/

vi nginx.conf

server {

listen 8089;

server\_name localhost;

#charset koi8-r;

#access\_log logs/host.access.log main;

location / {

root html;

index index.html index.htm;

}

1. 使用server命令启动nginx服务

cd /etc/init.d/

vi nginx 添加以下脚本内容

#!/bin/sh

#

# nginx - this script starts and stops the nginx daemon

#

# chkconfig: - 85 15

# description: NGINX is an HTTP(S) server, HTTP(S) reverse \

# proxy and IMAP/POP3 proxy server

# processname: nginx

# config: /etc/nginx/nginx.conf

# config: /etc/sysconfig/nginx

# pidfile: /var/run/nginx.pid

# Source function library.

. /etc/rc.d/init.d/functions

# Source networking configuration.

. /etc/sysconfig/network

# Check that networking is up.

[ "$NETWORKING" = "no" ] && exit 0

nginx="/usr/local/nginx/sbin/nginx"

prog=$(basename $nginx)

NGINX\_CONF\_FILE="/usr/local/nginx/conf/nginx.conf"

[ -f /etc/sysconfig/nginx ] && . /etc/sysconfig/nginx

lockfile=/var/lock/subsys/nginx

make\_dirs() {

# make required directories

user=`$nginx -V 2>&1 | grep "configure arguments:" | sed 's/[^\*]\*--user=\([^ ]\*\).\*/\1/g' -`

if [ -z "`grep $user /etc/passwd`" ]; then

useradd -M -s /bin/nologin $user

fi

options=`$nginx -V 2>&1 | grep 'configure arguments:'`

for opt in $options; do

if [ `echo $opt | grep '.\*-temp-path'` ]; then

value=`echo $opt | cut -d "=" -f 2`

if [ ! -d "$value" ]; then

# echo "creating" $value

mkdir -p $value && chown -R $user $value

fi

fi

done

}

start() {

[ -x $nginx ] || exit 5

[ -f $NGINX\_CONF\_FILE ] || exit 6

make\_dirs

echo -n $"Starting $prog: "

daemon $nginx -c $NGINX\_CONF\_FILE

retval=$?

echo

[ $retval -eq 0 ] && touch $lockfile

return $retval

}

stop() {

echo -n $"Stopping $prog: "

killproc $prog -QUIT

retval=$?

echo

[ $retval -eq 0 ] && rm -f $lockfile

return $retval

}

restart() {

configtest || return $?

stop

sleep 1

start

}

reload() {

configtest || return $?

echo -n $"Reloading $prog: "

killproc $nginx -HUP

RETVAL=$?

echo

}

force\_reload() {

restart

}

configtest() {

$nginx -t -c $NGINX\_CONF\_FILE

}

rh\_status() {

status $prog

}

rh\_status\_q() {

rh\_status >/dev/null 2>&1

}

case "$1" in

start)

rh\_status\_q && exit 0

$1

;;

stop)

rh\_status\_q || exit 0

$1

;;

restart|configtest)

$1

;;

reload)

rh\_status\_q || exit 7

$1

;;

force-reload)

force\_reload

;;

status)

rh\_status

;;

condrestart|try-restart)

rh\_status\_q || exit 0

;;

\*)

echo $"Usage: $0 {start|stop|status|restart|condrestart|try-restart|reload|force-reload|configtest}"

exit 2

esac

添加脚本权限

chmod 755 nginx

将脚本添加进chkconfig

chkconfig –add nginx

1. 将nginx设置为开机启动

chkconfig nginx on

1. 至此，nginx就可以使用以下命令进行启动和关闭，并且实现了开机自启

service nginx start

service nginx stop

## 6. 安装SVN

1. 通过yum安装svnserve

yum -y install subversion

1. 创建svn版本库

mkdir /var/svn/svnrepos/trunk

svnadmin create /var/svn/svnrepos/trunk

1. 配置修改

cd /var/svn/svnrepos/trunk/conf

目录下有3个文件：

authz：负责账号权限的管理，控制账号是否读写权限

passwd：负责账号和密码的用户名单管理

svnserve.conf：svn服务器配置文件

1. 修改authz文件信息

vi authz

在文件末尾添加：

[/]

admin = rw

root = rw

1. 修改passwd文件信息

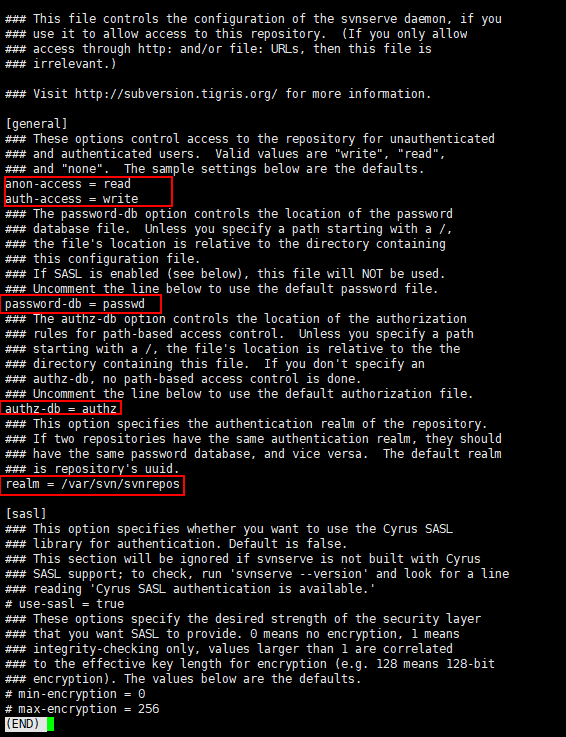
vi passwd

在文件末尾添加

admin = admin123

root = root123

1. 修改svnserve.conf



1. 启动svn

svnserve -d -r /vsr/svn/svnrepos

1. 关闭svn

killall svnserve

1. 查看svn是否启动

netstat -ln | grep 3690

1. 客户端连接svn

svn客户端输入 svn://ip 即可连接上

1. svn设置开机自启

在 /var/svn/svnrepos 目录下新建脚本 startup.sh，脚本内容如下：

svnserve -d -r /var/svn/svnrepos

为脚本添加权限 chmod 755 startup.sh

然后编辑以下文件

cd /etc/rc.d

vi rc.local

在文件末尾添加以下内容：

/var/svn/svnrepos/startup.sh

即可

相关软件的开机自启都可以通过这个方法实现

安装maven

1. 下载并删除maven到Linux上
2. 解压maven

mkdir /usr/maven

cd /usr/maven

cp /home/apache-maven-3.5.2-bin.tar.gz /usr/maven/

tar -zxvf apache-maven-3.5.2-bin.tar.gz

1. 修改环境变量

在/etc/profile文件中添加以下几行

export MAVEN\_HOME=/usr/maven/apache-maven-3.5.2

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

执行 source /etc/profile 生效

1. 验证是否生效

mvn -v

## 7. 安装Jenkins

1. 安装前需要先安装tomcat、jdk、maven
2. 下载jenkins.war包到tomcat的webapp目录下
3. 修改tomcat中server.xml中的端口号，并将其设置为开机自启

server.xml文件中有三个端口设置

　　<Server port="8005" shutdown="SHUTDOWN"> ：关闭时使用

　　<Connector port="8080" protocol="HTTP/1.1" connectionTimeout="20000" redirectPort="8443" /> ： 一般应用使用

<Connector port="8009" protocol="AJP/1.3" redirectPort="8443" />：为AJP端口，即容器使用，如 APACHE能通过AJP协议访问Tomcat的8009端口

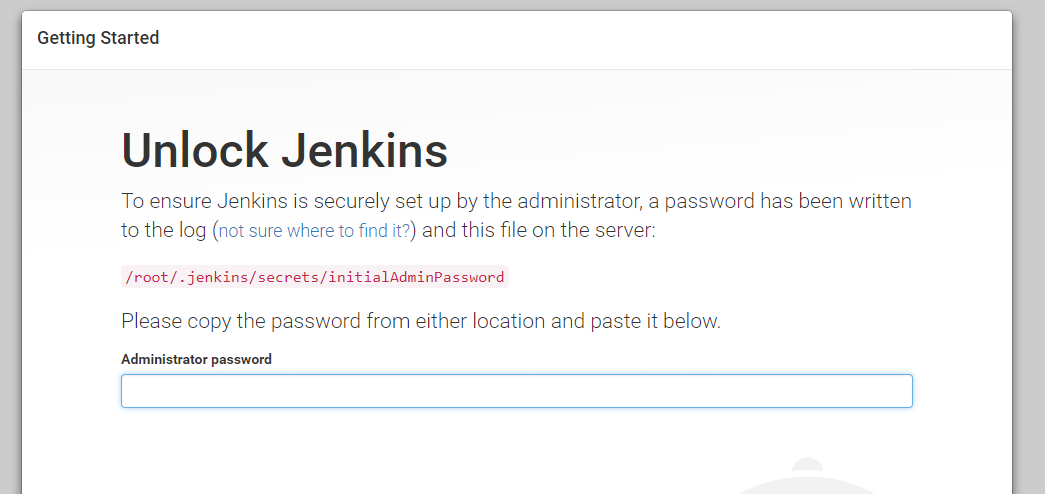
修改三个端口号为 8006/8081/8010

设置开机自启，修改/etc/rc.local文件

1. 启动Jenkins

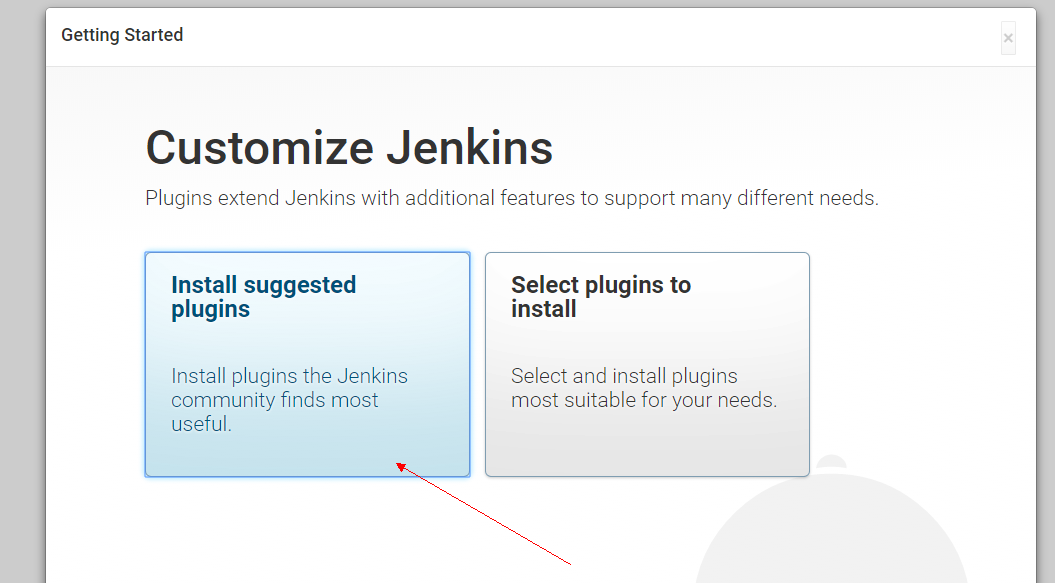
启动tomcat，在浏览器中输入<http://ip:8081/jenkins>

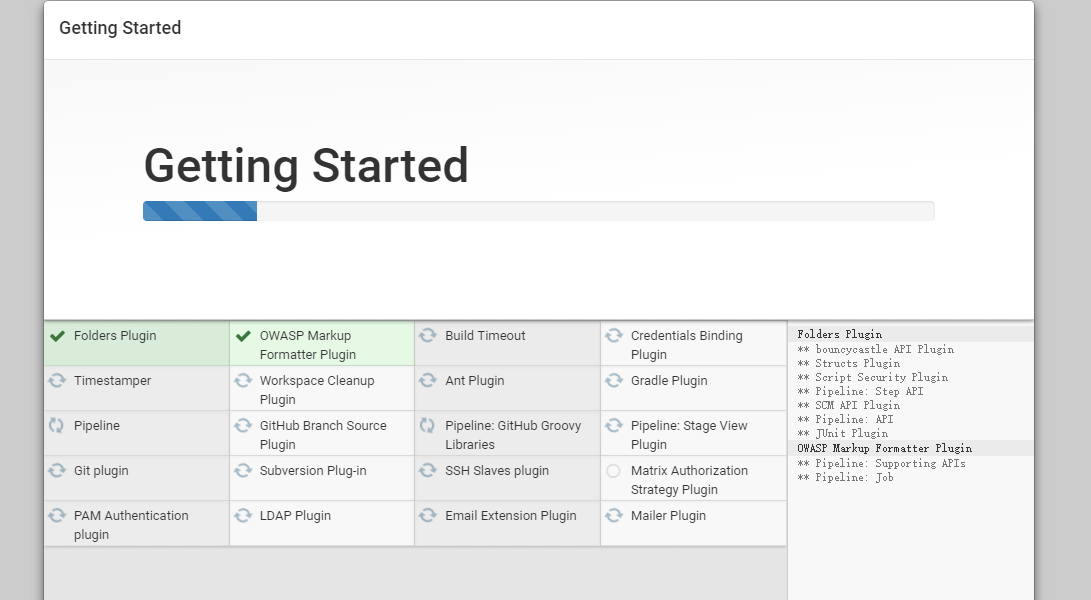
1. 设置Jenkins

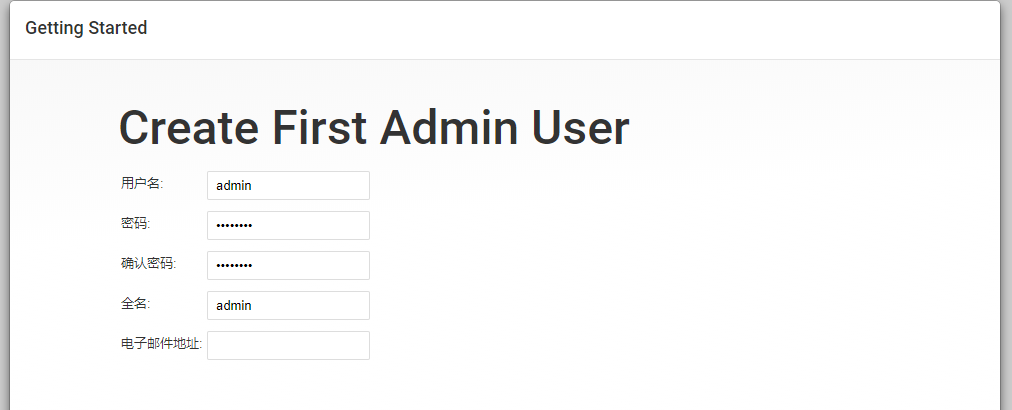


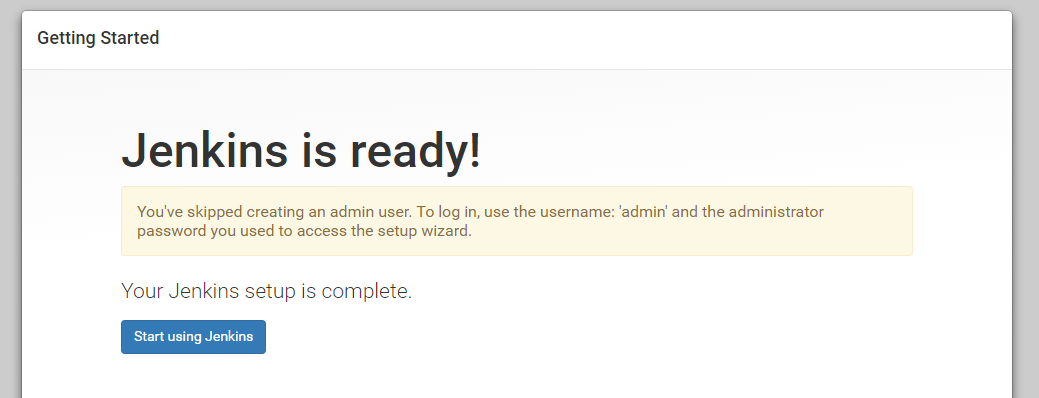
vi /root/.jenkins/secrets/initialAdminPassword

获取密码









Jenkins 插件下载地址

<http://updates.jenkins-ci.org/download/plugins/>

1. 重启jenkin后登陆不进去：

进入Jenkins的主目录 cd /root/.jenkins

vi config.xml

修改 <useSecurity>false</useSecurity>

删除<authorizationStrategy>\*\*\*\*\*</authorizationStrategy>和<securityRealm>\*\*\*\*</securityRealm>两项内容

重启Jenkins