Docker部署Jenkins

参考链接：https://github.com/jenkinsci/docker/blob/master/README.md

1 准备目录

在home目录下面创建`jenkins`目录

```
mkdir /home/jenkins
cd /home/jenkins
mkdir jenkins_home
# 把当前目录的拥有者赋值给uid 1000
chown -R 1000 jenkins_home
# 创建maven仓库文件目录
mkdir -p maven/m2
```

将资源目录下面的setting.xml上传到`maven/m2`目录下面

2 开放端口

注意：如果是云服务器，不要使用下列方式开放端口，请在云控制里面配置入方向规则即可

```
# 开放端口
firewall-cmd --add-port 8380/tcp --permanent
# 重新加载防火墙
firewall-cmd --reload
```

3 服务编排

执行命令行`vi docker-compose.yml`创建服务编排文件，然后在文件中写入下列内容后保存。

当然你也可以使用资源目录中提供`yml`文件直接上传到`jenkins`目录即可，然后跳过本步骤。

```
version: '3'
services:
  jenkins:
    image: jenkins/jenkins:lts
    container_name: jenkins
    privileged: true
    user: root
    environment:
      TZ: Asia/Shanghai
    ports:
      - "8380:8080"
      - "50000:50000"
    volumes:
      - /home/jenkins/jenkins_home:/var/jenkins_home
      - /home/jenkins/maven/m2:/root/.m2
      - /var/run/docker.sock:/var/run/docker.sock
```

4 启动服务

执行命令`docker-compose up`。

首次启动会拉取镜像，效果如下图所示。

```
[root@localhost jenkins]# docker-compose up
[+] Running 0/13
 ⠏ jenkins Pulling                                                    69.5s
   ⠹ e9aef93137af Pulling fs layer                                     3.0s
   ⠹ 941a647ff357 Pulling fs layer                                     3.0s
   ⠹ f0f251032519 Pulling fs layer                                     3.0s
   ⠹ d4c632816288 Waiting                                              3.0s
   ⠹ 07b0d8ca8ab7 Waiting                                              3.0s
   ⠹ 995009f196ec Waiting                                              3.0s
   ⠹ 25f6587a42fb Waiting                                              3.0s
   ⠹ a559a5c3d549 Waiting                                              3.0s
   ⠹ aa636159e94b Waiting                                              3.0s
   ⠹ 862717e1a01c Waiting                                              3.0s
   ⠹ 6778d605d2d1 Waiting                                              3.0s
```

如果看到类似下面的结果表示服务器正常启动。

```
jenkins  |
jenkins  | *************************************************************
jenkins  | *************************************************************
jenkins  | *************************************************************
jenkins  |
jenkins  | Jenkins initial setup is required. An admin user has been created and a password generated.
jenkins  | Please use the following password to proceed to installation:
jenkins  |
jenkins  | bb75ca090d1f4694991fd2a9e4cb9e7b
jenkins  |
jenkins  | This may also be found at: /var/jenkins_home/secrets/initialAdminPassword
jenkins  |
jenkins  | *************************************************************
jenkins  | *************************************************************
jenkins  | *************************************************************
```

1 这是管理员初始密码

然后`Ctrl + C`结束前台启动，成功停止效果如下图所示

```
jenkins  | Jenkins initial setup is required. An admin user has been created and a password generated.
jenkins  | Please use the following password to proceed to installation:
jenkins  |
jenkins  | bb75ca090d1f4694991fd2a9e4cb9e7b
jenkins  |
jenkins  | This may also be found at: /var/jenkins_home/secrets/initialAdminPassword
jenkins  |
jenkins  | *************************************************************
jenkins  | *************************************************************
jenkins  | *************************************************************
jenkins  |
jenkins  | 2024-07-11 12:52:35.515+0000 [id=34] INFO    jenkins.InitReactorRunner$1#onAttained: Completed initialization
jenkins  | 2024-07-11 12:52:35.545+0000 [id=24] INFO    hudson.lifecycle.Lifecycle#onReady: Jenkins is fully up and running
jenkins  | 2024-07-11 12:52:36.916+0000 [id=47] INFO    h.m.DownloadService$Downloadable#load: Obtained the updated data fil
jenkins  | 2024-07-11 12:52:36.916+0000 [id=47] INFO    hudson.util.Retrier#start: Performed the action check updates server
^CGracefully stopping... (press Ctrl+C again to force)
[+] Running 1/1
 ⠿ Container jenkins  Stopped
canceled
[root@localhost jenkins]#
```

切换到后台模式启动服务，使用命令`docker-compose up -d`，启动成功如下图所示

```
[root@localhost jenkins]# docker-compose up -d
[+] Running 1/1
 ⠿ Container jenkins  Started
[root@localhost jenkins]#
```

使用`docker ps`指令查看进程

```
[root@localhost jenkins]# docker-compose up -d
[+] Running 1/1
 # Container jenkins  Started
[root@localhost jenkins]# docker ps | grep jenkins
f6ee73f25a2e   jenkins/jenkins:lts        "/usr/bin/tini -- /u…"   About a minute ago   Up About a minute   0.0.0.0:50000->50000/tcp,
080/tcp   jenkins
[root@localhost jenkins]#
```

5 初始化服务

5.1 初始化配置

浏览器访问 http://ip:8380

输入管理员密码，我此时的管理员密码是 bb75ca090d1f4694991fd2a9e4cb9e7b，通过启动日志查看到，使用下面的命令。

```
1  docker logs jenkins
```

```
*************************************************************
*************************************************************
*************************************************************

Jenkins initial setup is required. An admin user has been created and a password generated.
Please use the following password to proceed to installation:

bb75ca090d1f4694991fd2a9e4cb9e7b

This may also be found at: /var/jenkins_home/secrets/initialAdminPassword

*************************************************************
*************************************************************
*************************************************************
```

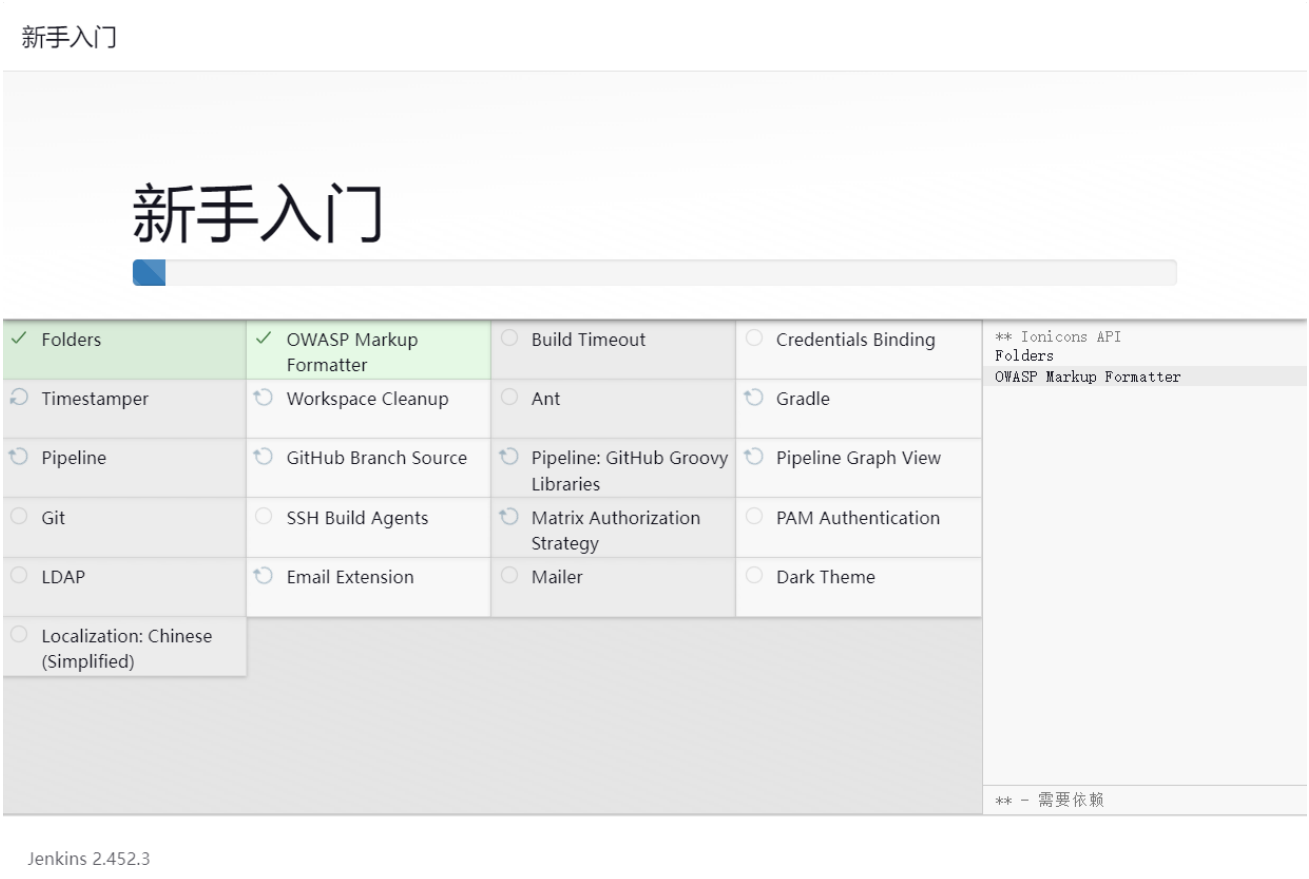
输入密码后，进入插件安装选择页面

新手入门

选择安装推荐插件，然后静静的等待安装即可。

如果在安装过程中如果出现插件安装失败，多半是网络问题，上个代理试试吧。

新手入门



TIPS：安装过程可能存在安装失败，可能是网络问题，也有可能是版本兼容问题，多重试几次，如果还是失败，可以跳过不管。

插件安装完后，会进入创建管理员账号页面

在表单中输入你的信息，然后保存进入配置实例url页面，本地尝试的话默认即可。

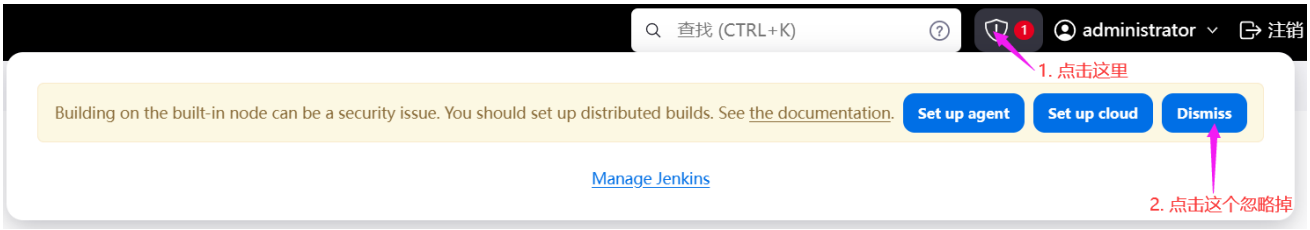
点击完成后，进入就绪页面。

新手入门

点击开始使用，进入系统。

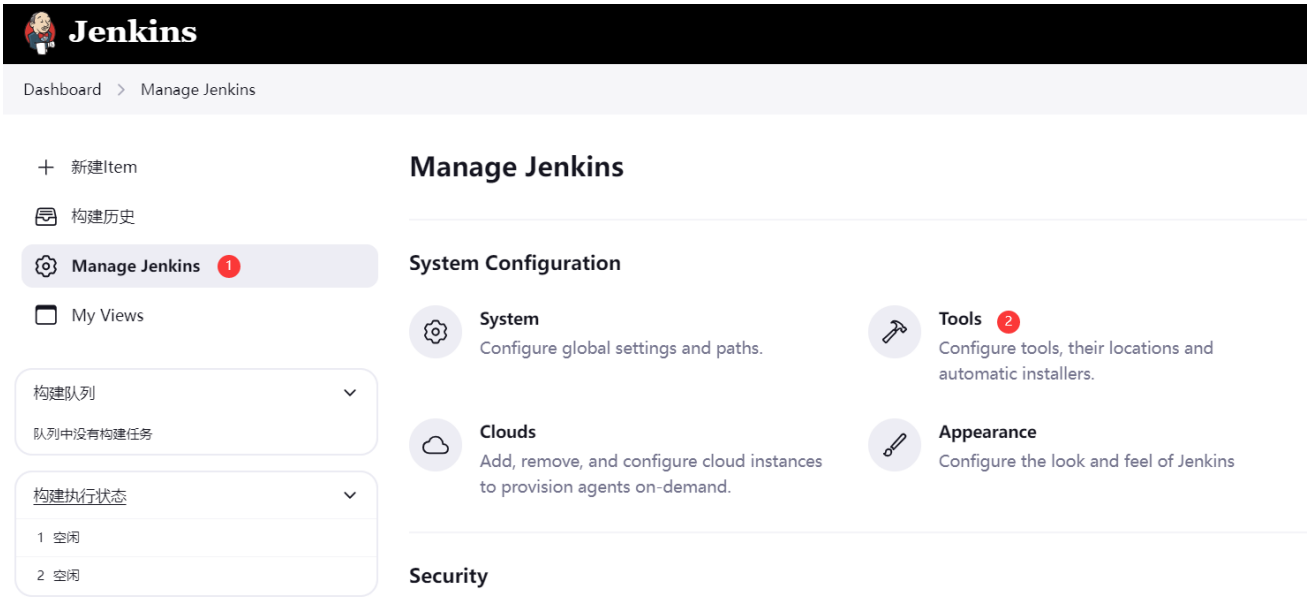
到此服务器初始完成，可以使用了。

进入系统可能会看到警告信息或安全提醒，下面来修正一下。



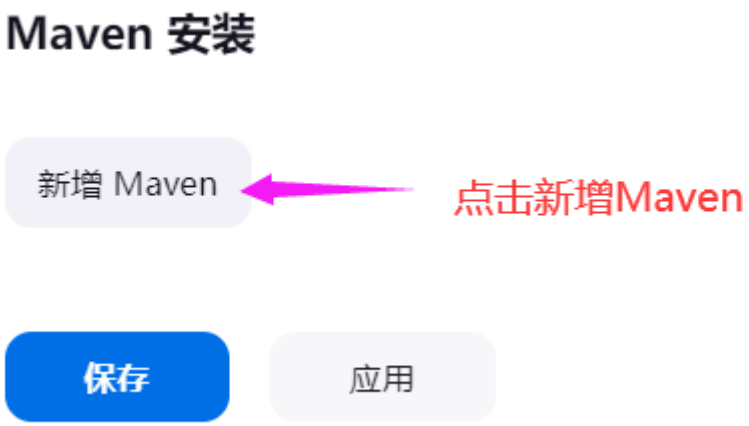
5.2 全局工具

通过下图所示的路径可以找到全局工具配置



5.2.1 添加Maven（可选）

这个配置Java方向需要配置的，打开配置页面后，拉到最下面，然后新增一个maven



输入Maven版本信息

5.2.2 设置JDK版本（可选）

这个也是Java方向可能需要配置的。

首先需要创建一个目录来放jdk

```
cd /home/jenkins/jenkins_home
mkdir jdks
cd jdks
```

然后到Oracle官网或镜像网站去下载一个你要的JDK版本的安装包，下载x64架构的Linux tar.gz包，不要下载成arm架构的包

- 官方地址：https://www.oracle.com/java/technologies/downloads/
- 镜像地址：https://d10.injdk.cn/openjdk/oraclejdk/

下面以Java8举例，我的安装包的名称是 jdk-8u421-linux-x64.tar.gz

包和目录准备好后，服务器对应目录结构如下图所示

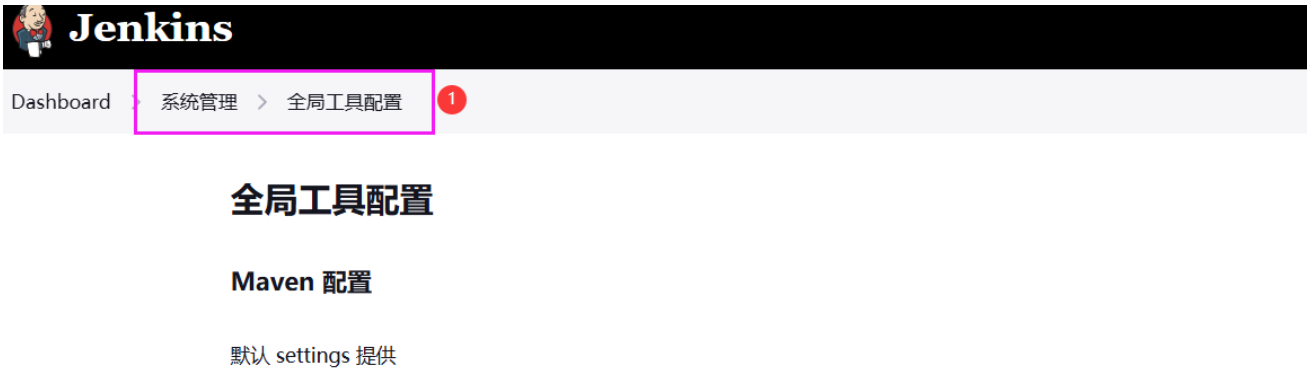
```
[root@localhost ~]# cd /home/jenkins/jenkins_home/
[root@localhost jenkins_home]# mkdir jdks
[root@localhost jenkins_home]# cd jdks/
[root@localhost jdks]# wget https://d10.injdk.cn/openjdk/oraclejdk/8/jdk-8u421-linux-x64.tar.gz
--2025-06-19 17:44:35--  https://d10.injdk.cn/openjdk/oraclejdk/8/jdk-8u421-linux-x64.tar.gz
正在解析主机 d10.injdk.cn (d10.injdk.cn)... 110.42.44.84
正在连接 d10.injdk.cn (d10.injdk.cn)|110.42.44.84|:443... 已连接。
已发出 HTTP 请求，正在等待回应... 200 OK
长度：146729827 (140M) [application/gzip]
正在保存至："jdk-8u421-linux-x64.tar.gz"

jdk-8u421-linux-x64.tar.gz                    100%[=============================================

2025-06-19 17:44:48 (11.1 MB/s) - 已保存 "jdk-8u421-linux-x64.tar.gz" [146729827/146729827])

[root@localhost jdks]# ls
jdk-8u421-linux-x64.tar.gz
[root@localhost jdks]# pwd
/home/jenkins/jenkins_home/jdks
[root@localhost jdks]#
```

使用tar命令解压安装包，命令行如：`tar -zxvf jdk-8u421-linux-x64.tar.gz`，解压完成后如下图所示

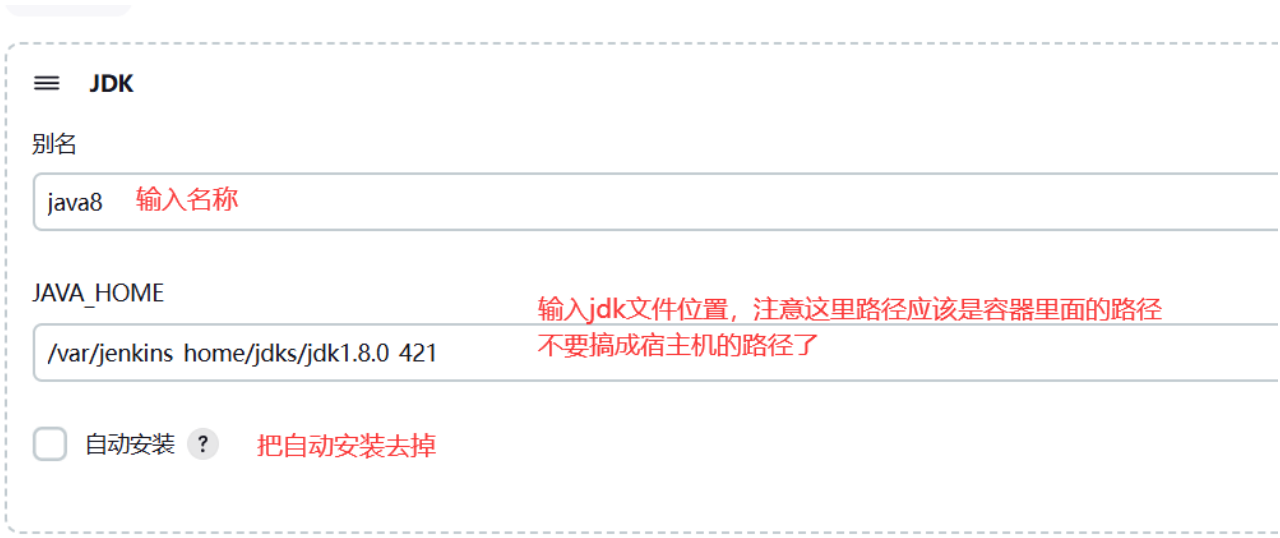
```
jdk1.8.0_421/man/ja_JP.UTF-8/man1/javapackager.1
jdk1.8.0_421/man/ja
jdk1.8.0_421/release
jdk1.8.0_421/src.zip
jdk1.8.0_421/jvisualvm.txt
jdk1.8.0_421/jmc.txt
jdk1.8.0_421/jre/lib/applet/
[root@localhost jdks]# ls
jdk1.8.0_421  jdk-8u421-linux-x64.tar.gz
[root@localhost jdks]#
```

提示：此时dks对应容器里面的目录是 /var/jenkins_home/jdks/

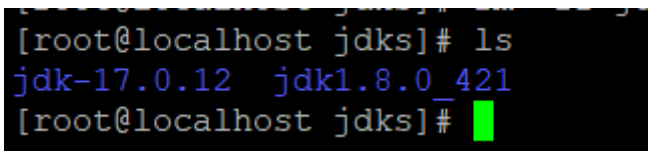
然后在全局工具配置中，新增JDK配置



然后按照下图方式输入数据



未来支持在任务中可以选择jdk，我再配置一个jdk17，配置方式和前面的步骤一样



填写完成后点击，保存按钮。

然后你再后续使用任务的时候可以选择你要使用的JDK版本了，示例效果如下图所示

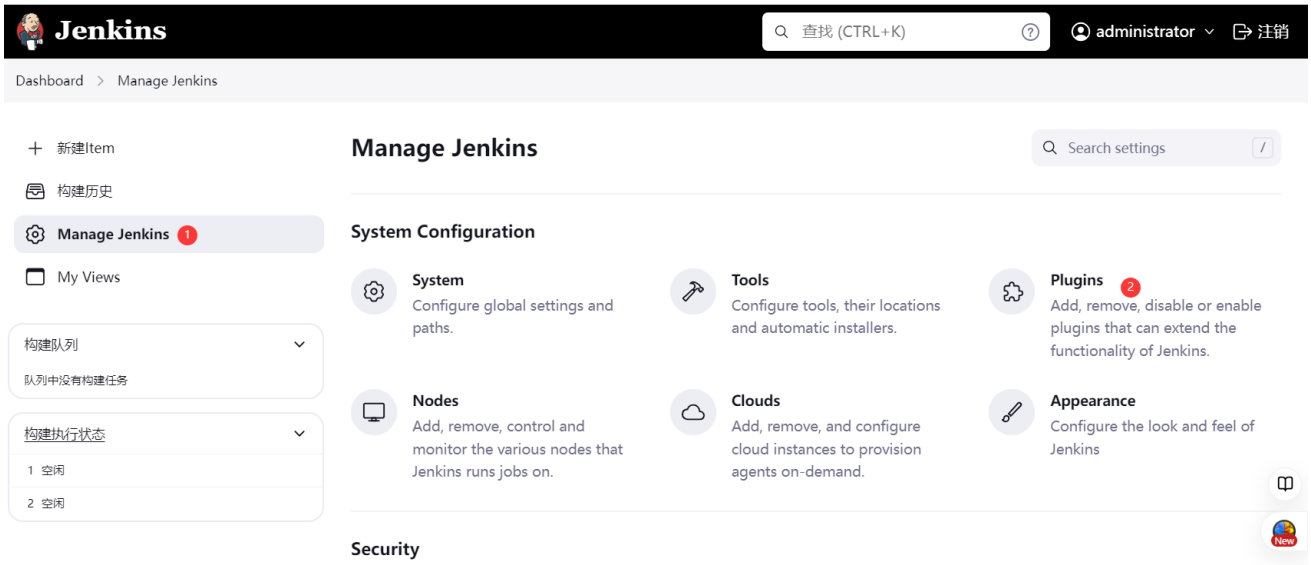
当然如果你只配置了一个jdk，在任务这里就没有选择操作，直接使用你指定的哪个jdk

5.3 安装SSH插件

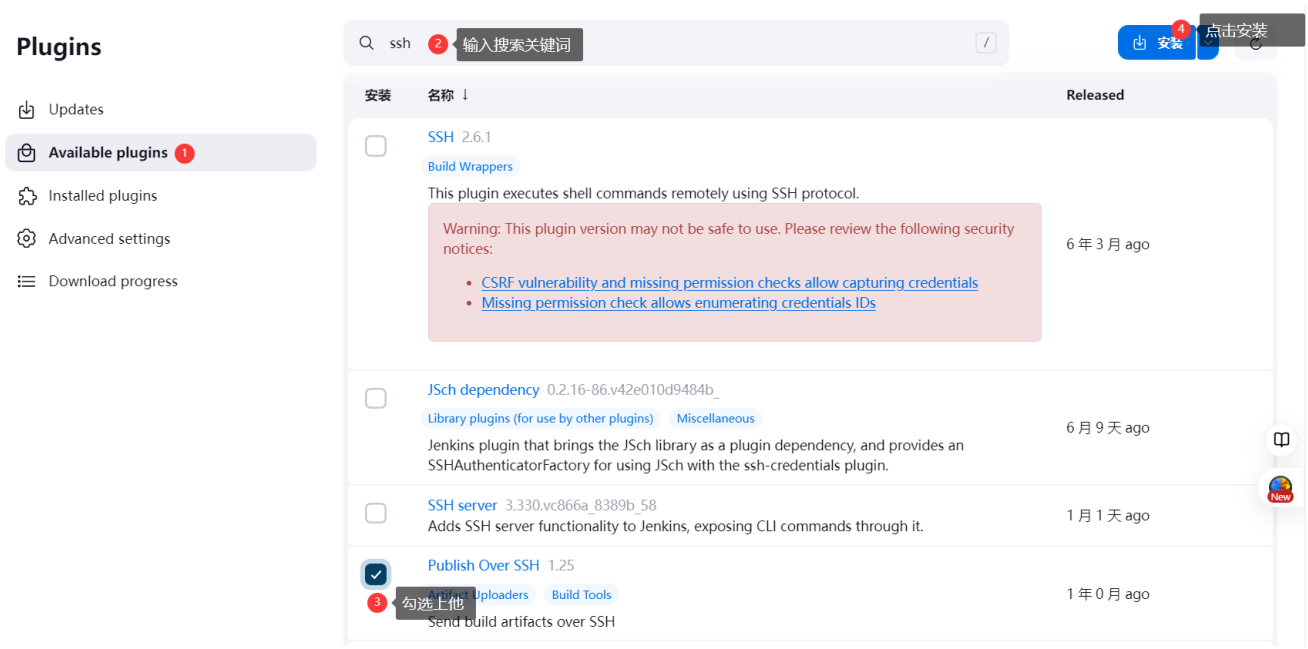
如果想要在Jenkins中通过SSH远程登录服务器执行脚本，需要安装SSH插件，可以通过以下方式安装。

5.3.1 安装插件

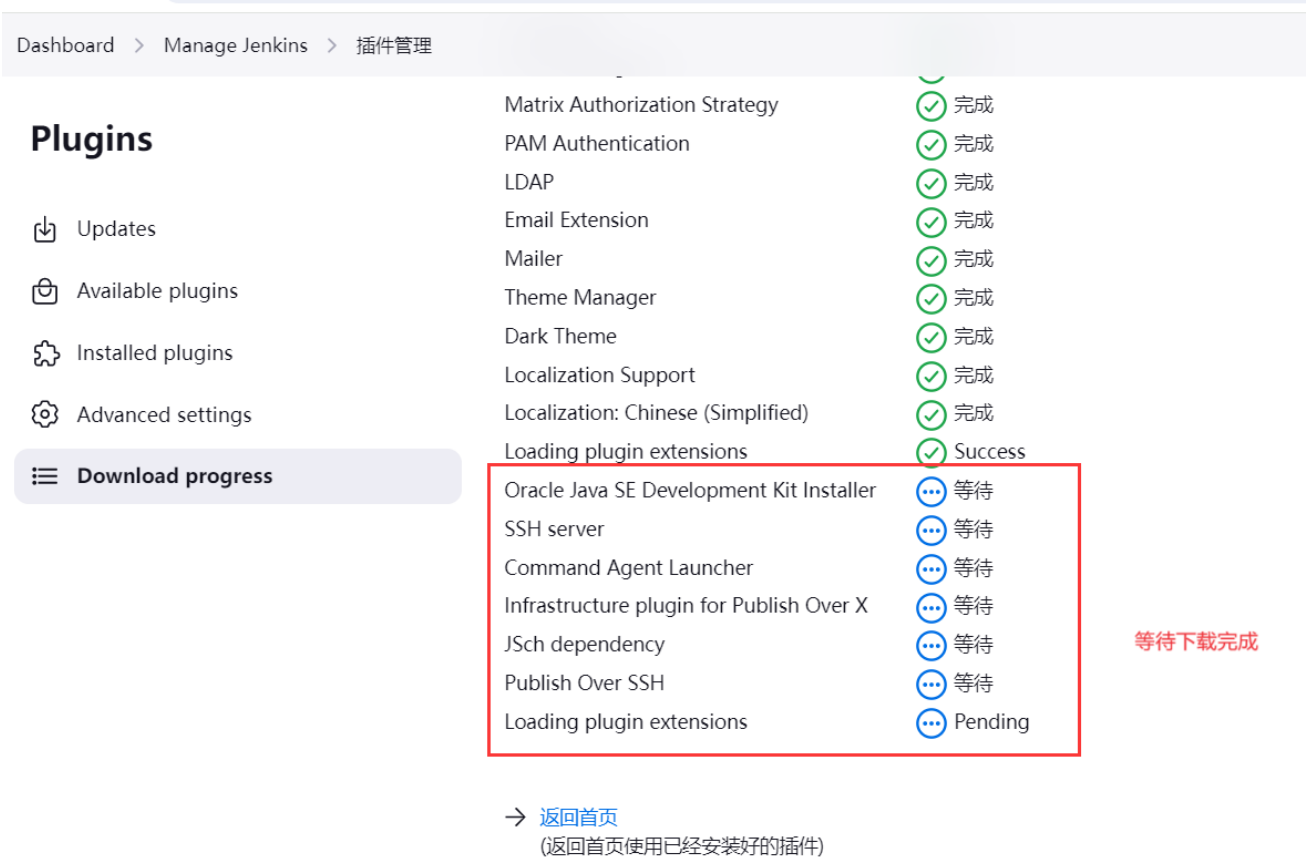
进入插件管理页面



选择可用插件选项，然后输入搜索关键词



下面是安装过程页面



下载完成后，点击勾选重启Jenkins

勾选上之后，进入关闭提示界面

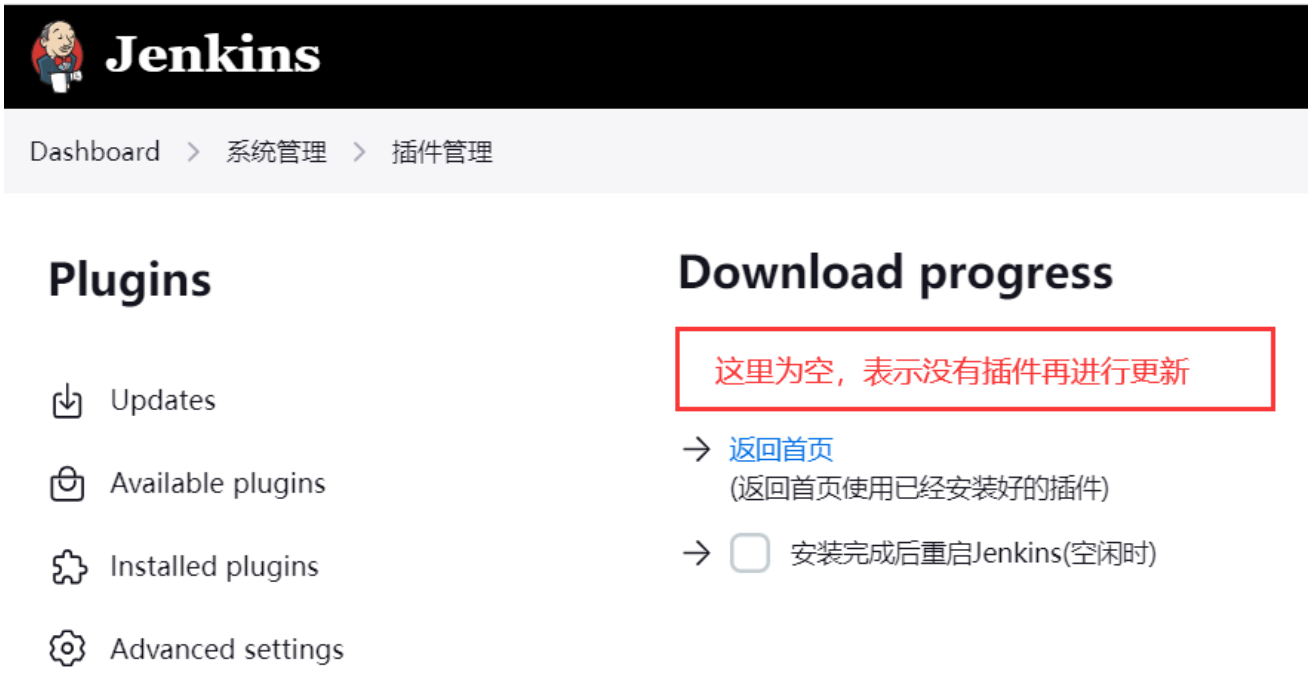
关闭后，服务器如果没有自动重启（docker安装都需要手动重启一下），执行通过命令行手动启动一下服务。

```
docker-compose start
```

```
[root@localhost jenkins]# docker-compose start
[+] Running 1/1
 # Container jenkins  Started
[root@localhost jenkins]#
```

重启成功后可能会自动进入登录界面

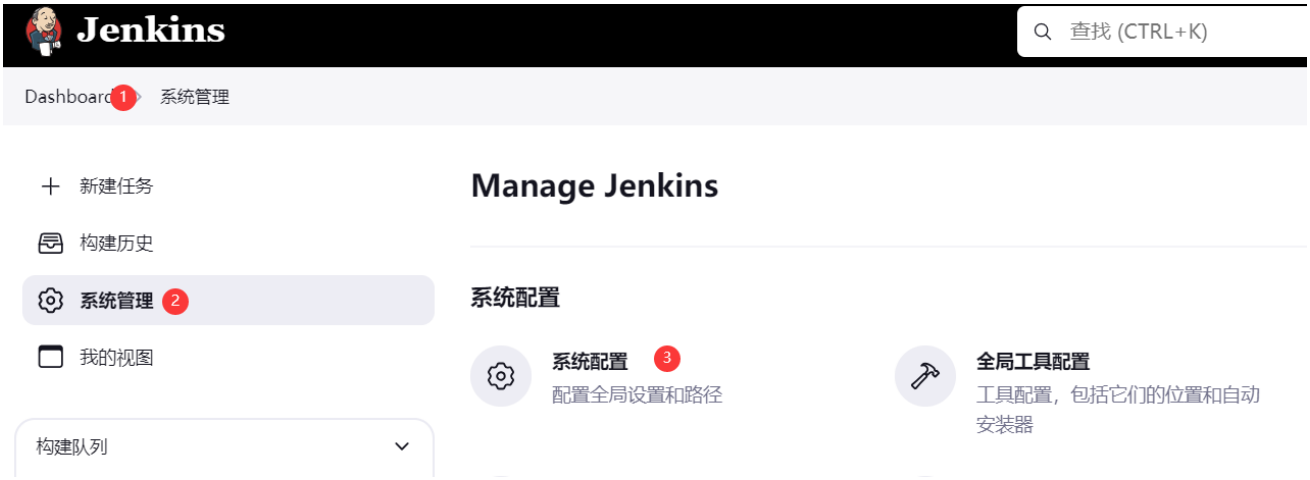
输入登录信息点击登录后，会进入到刚才的下载界面



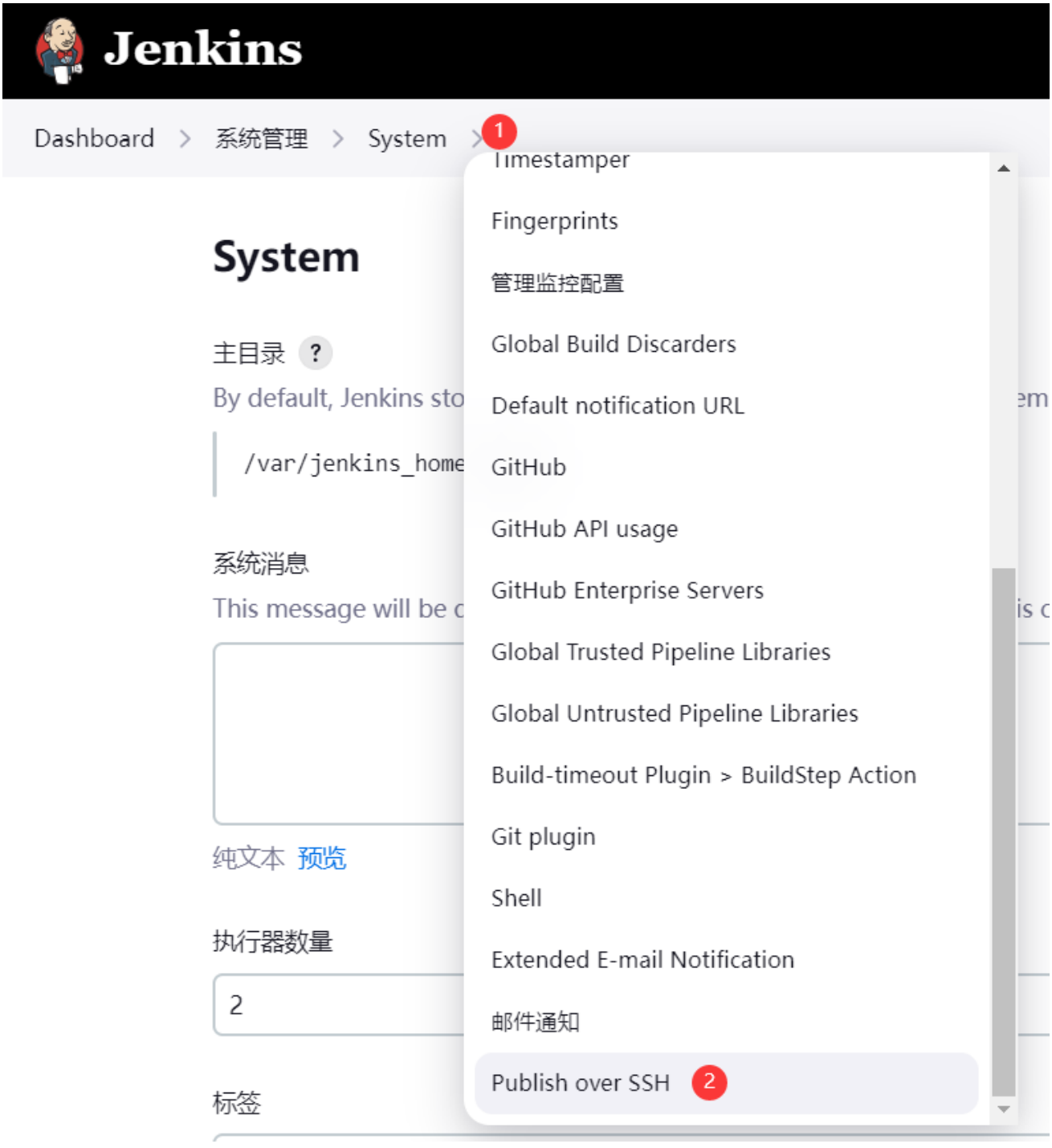
5.3.2 配置插件

插件安装好后，可以在系统配置中添加要远程连接的服务端，如果后续需要添加更多的Linux服务器，也是通过下面的流程添加。

首先找到系统配置。

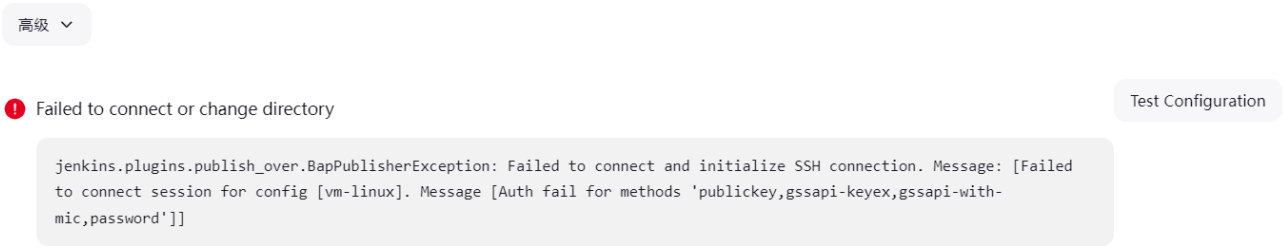


找到SSH Server



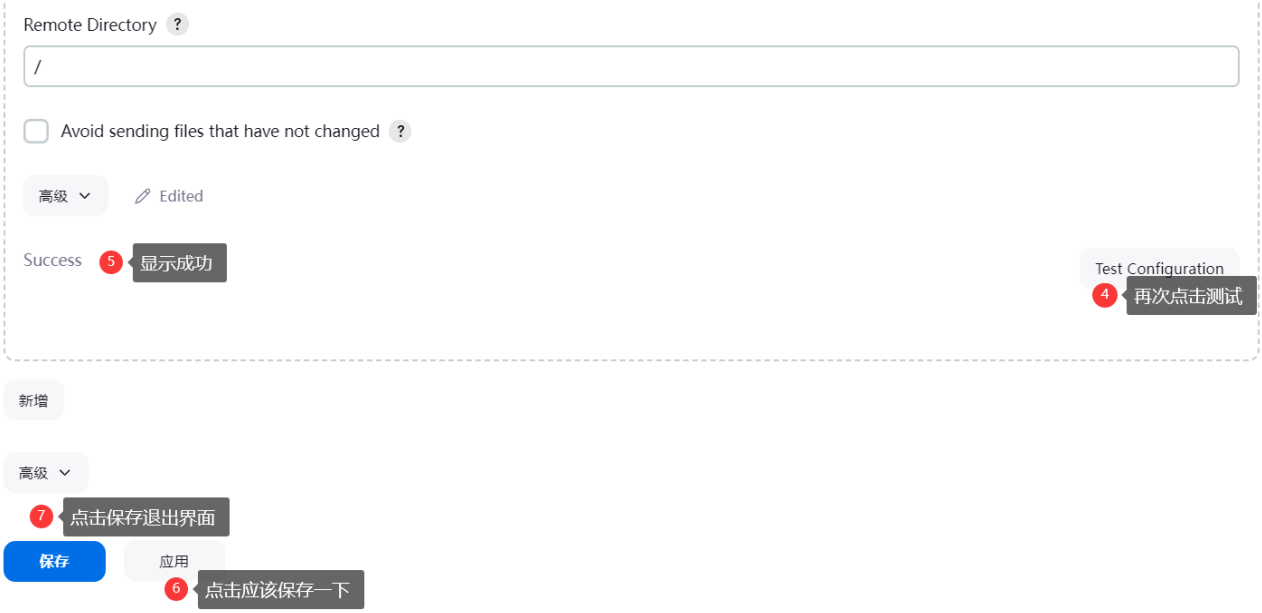
输入信息

此时应该是连接失败

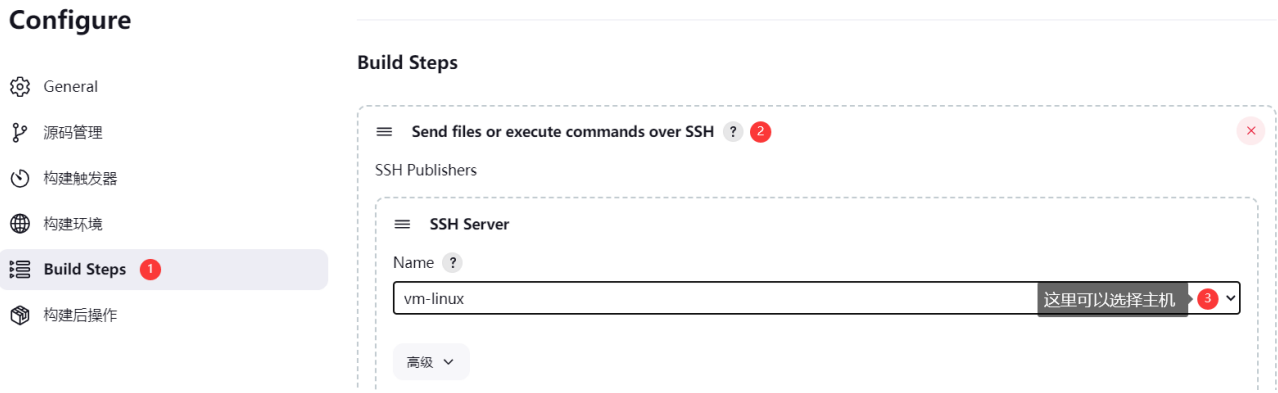


点击高级输入密码

测试与保存



这样你就可以在构建过程中使用了，示例如下图，***这个你可以在后面的构建任务中测试。***



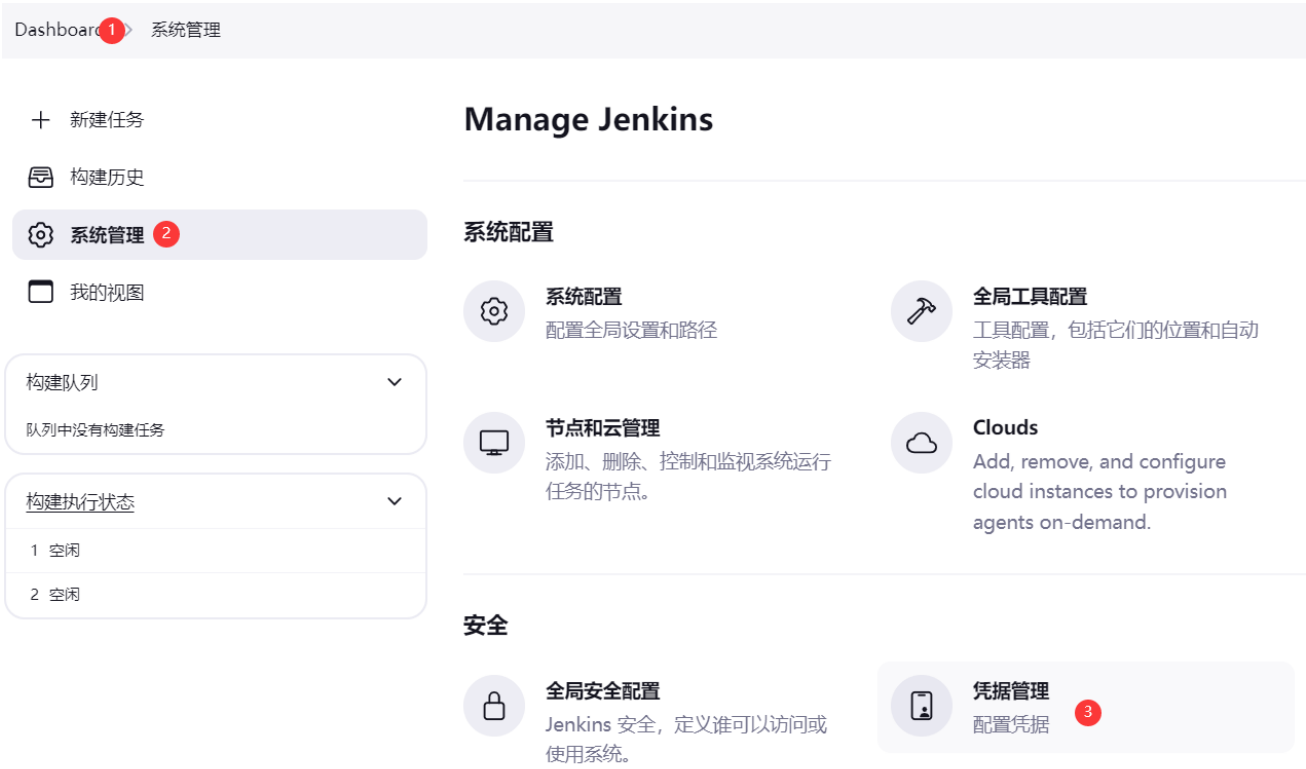
6 其他配置

6.1 添加凭据

项目一般会托管到gitee、github、gitlab等git托管平台上面，你可以提前添加好账号凭证，方便你后续使用。

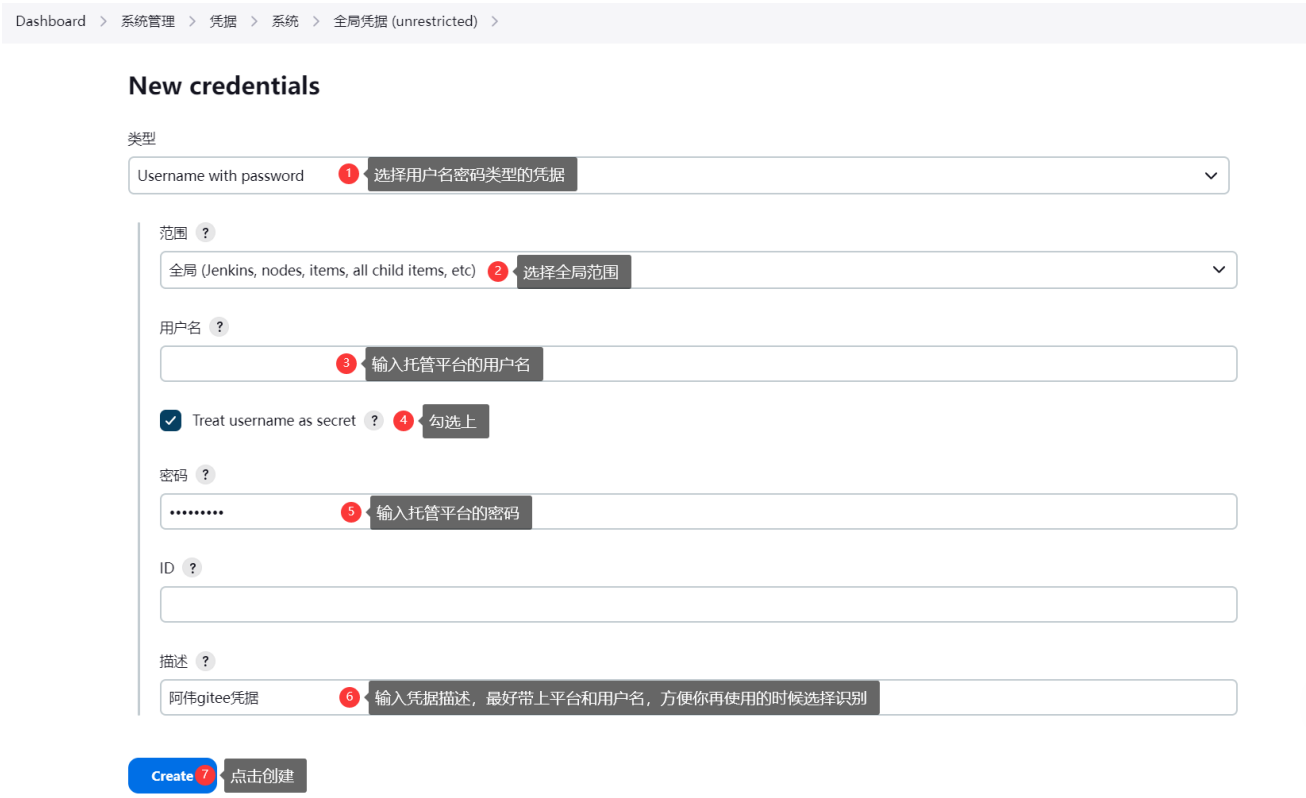
6.1.1 找到凭证管理

在Jenkins中找到凭证管理



添加一个全局凭证

6.1.2 填写凭据内容



创建成功后会返回凭据列表，在这里能够看到你的凭据

后面你再使用的时候选择上对应凭据就行了

6.2 配置GitHub SSH（可选）

为了解决GitHub源码管理问题，有时候无法clone源码，所以需要配置SSH。

6.2.1 生成ssh秘钥

首先进入Jenkins容器

```
docker exec -it jenkins /bin/bash
```

使用ssh-keygen命令生成密钥

```
ssh-keygen -t rsa -b 4096 -C "GitHub邮箱"
```

遇到输入内容，一路回车即可，如下图所示：

```
[root@localhost ~]# docker exec -it jenkins /bin/bash
root@65b146725084:/# ssh-keygen -t rsa -b 4096 -C "2[redacted]@qq.com"
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Created directory '/root/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa
Your public key has been saved in /root/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:Bw7IuxdSALfkWCL29Ve8eR0dNvvJt14QFzg1kt+8AFY 2[redacted]@qq.com
The key's randomart image is:
+---[RSA 4096]----+
|..o.=.    ...E+Bo|
|...O.+.   .+ +o.*|
|  ..= o.... + =o+|
|     o o.. o o.==|
|    o . S . . oo+|
|     o . .     oo|
|    . .        ..|
|     .        . .|
|              . .|
+----[SHA256]-----+
```

文件保存的目录

此时会默认把秘钥对文件生成到`/root/.ssh/`目录中。

6.2.2 配置GitHub

首先获取公钥，执行下面命令行

```
cat /root/.ssh/id_rsa.pub
```

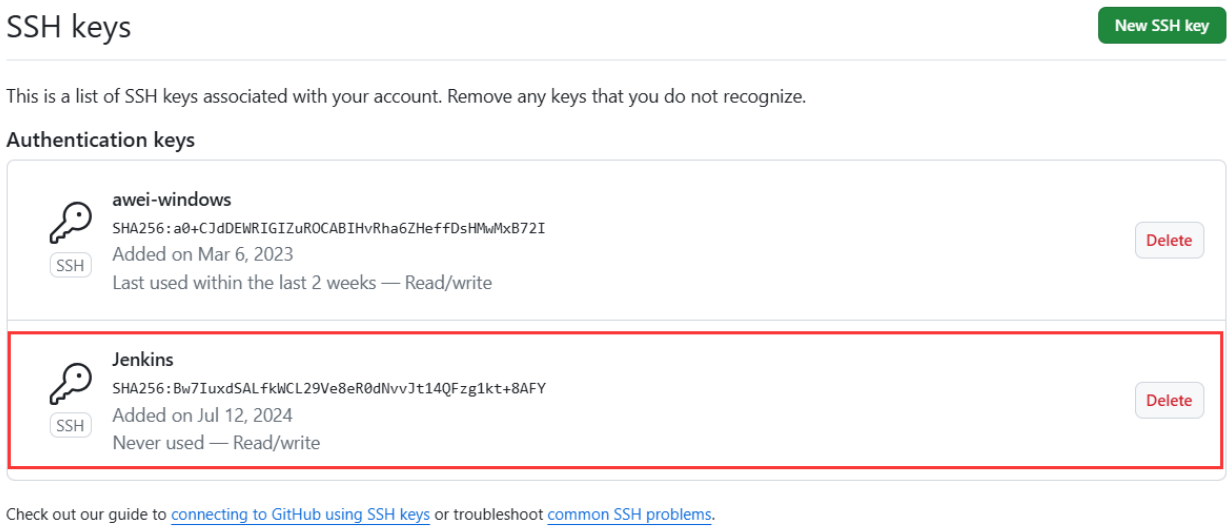
```
root@65b146725084:/# cat /root/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAACAQDLMA6+muGZt5O/0dJb/laJIPT2jfCd7ZAhCzR4wCZFIdSr3DD4YwR9Kd3pzEFGAgCz3d
nzyiijv2onfLbPuSh36XW3RQb5nSQcfV5AN8xYoz04DninR/4vEY+gavjyryvnKmbEeTuLaNmolOrJn1Mmq8jz0qxP3tY55FXO5k5ULXla
/iMTNJ2MAkJnkaqJitceptXjN8mOLZ2fs+Huiwk9oZN5U07/QG0X4I/YkX+LitK7j9ejTiX5Z3jF0ToctUWkaY5ATxJvIUAfku1YrJUCqS
H2jow8SQmRAfxZpk9Y85zCP9DTz4hOGqcN/AK2Hv1HTWbMbTL7MoWRB3cAqQsLSjLAR+oz6wxgPA0Ob3+SeDDgOxi3ieQG4ddusH1EhCy+
```

复制公钥内容备用。

打开GitHub SSH Key新建链接：https://github.com/settings/ssh/new

然后在执行下图所示的操作

添加成功后

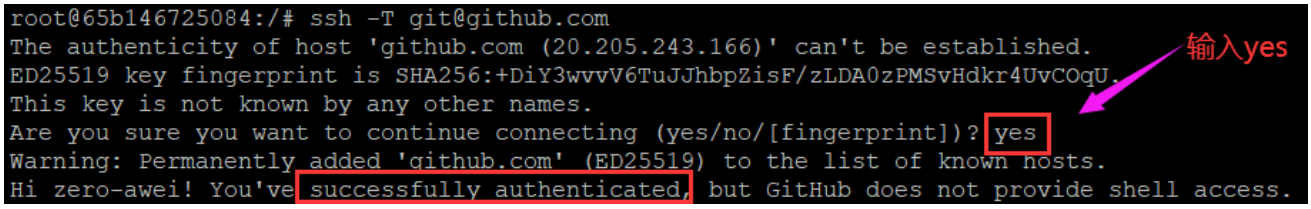


连接测试

在Jenkins容器里面执行命令

```
ssh -T git@github.com
```

执行结果如如下图所示



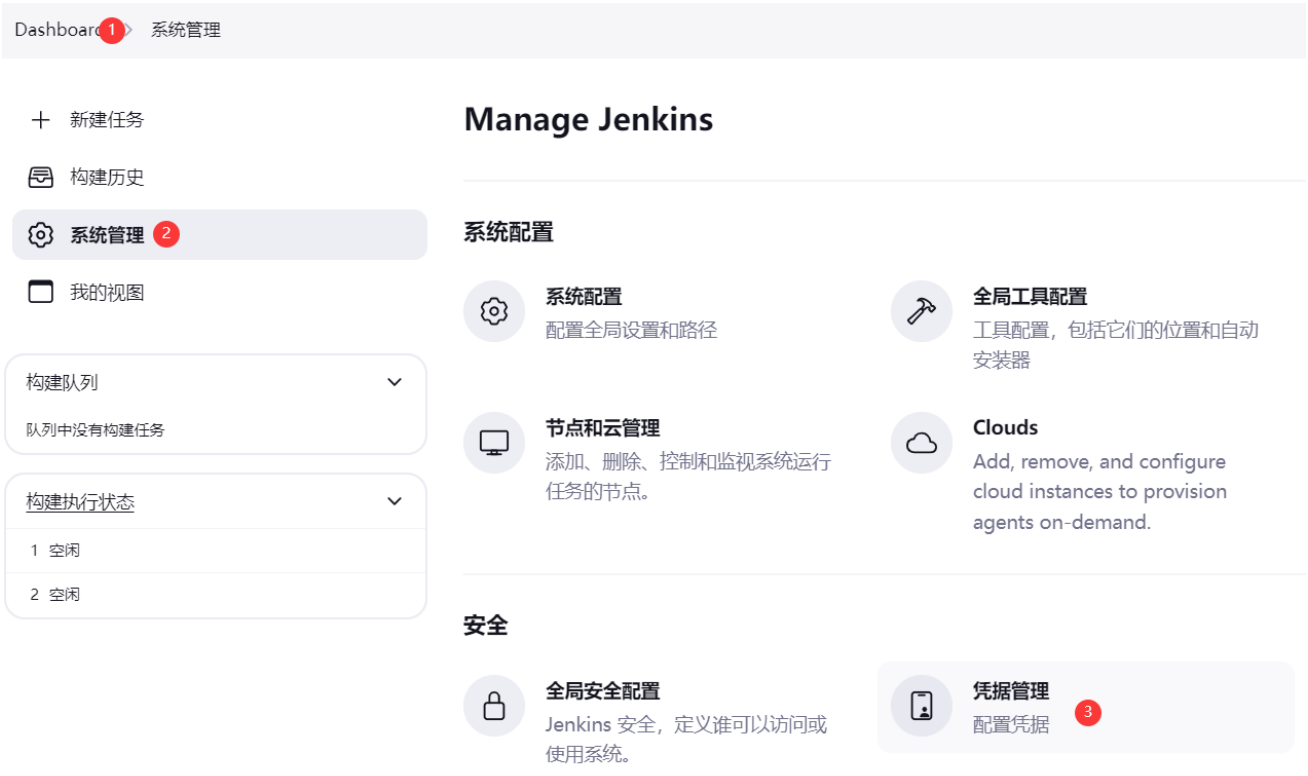
6.2.3 管理凭证

首先查看私钥，执行下面命令

```
cat /root/.ssh/id_rsa
```

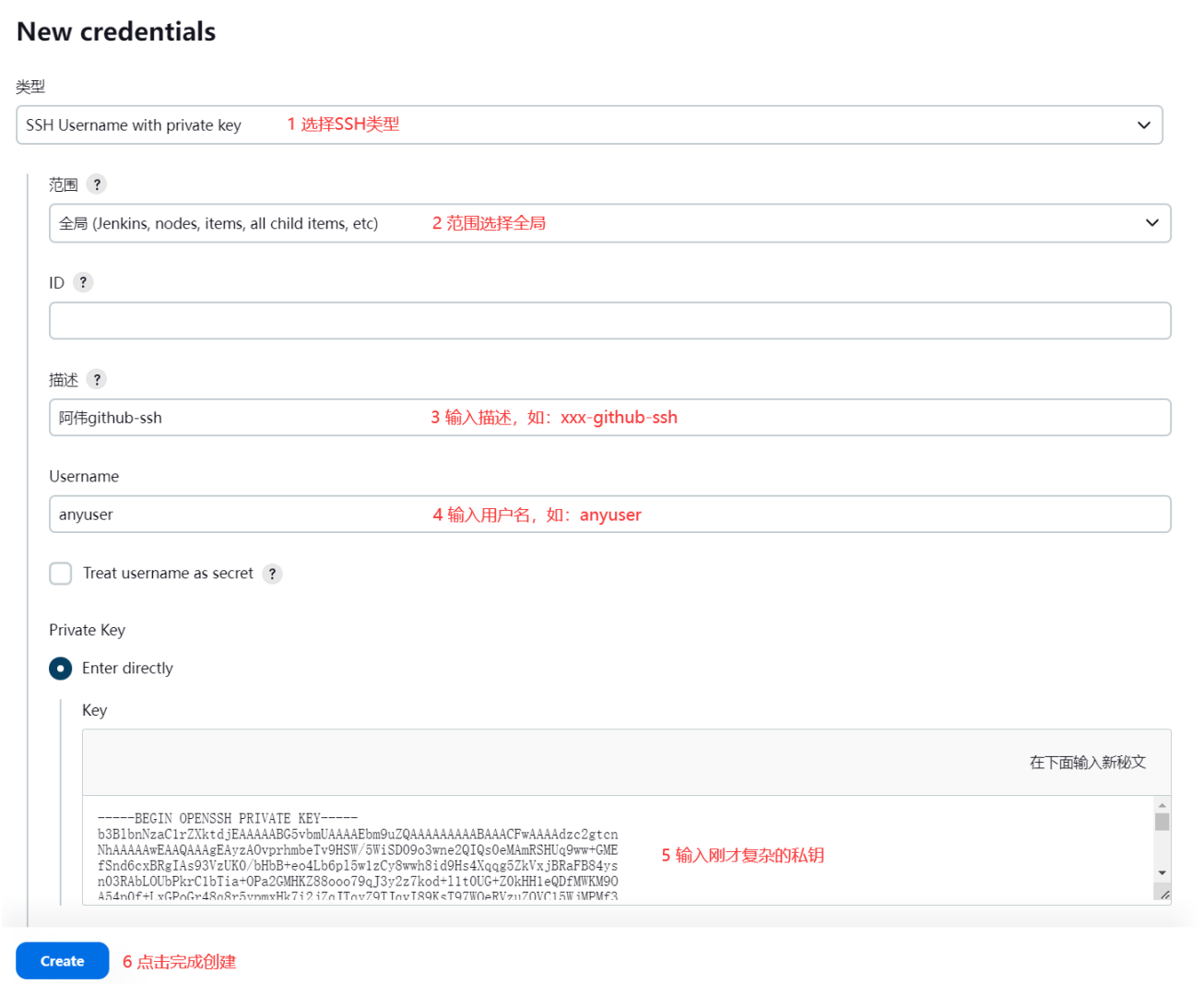
```
root@65b146725084:/# cat /root/.ssh/id_rsa
-----BEGIN OPENSSH PRIVATE KEY-----
b3BlbnNzaC1rZXktdjEAAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAACFwAAAAdzc2gtcn
NhAAAAAwEAAQAAAgEAyzAOvprhmbeTv9HSW/5WiSD09o3wne2QIQs0eMAmRSHUq9ww+GME
fSnd6cxBRgIAs93VzUK0/bHbB+eo4Lb6pl5w1zCy8wwh8id9Hs4Xqqg5ZkVxjBRaFB84ys
n03RAbLOUbPkrC1bTia+OPa2GMHKZ88ooo79qJ3y2z7kod+l1t0UG+Z0kHH1eQDfMWKM9O
A54p0f+LxGPoGr48q8r5ypmxHk7i2jZqJTqyZ9TJqvI89KsT97WOeRVzuZOVC15WjMPMf3
x5TUl3cgNSyHU9qSWDSui0pMlSiciA/WBuzlKd6ICi9R7GUB0kSajHt5zYkltgmEt7G1Oe
gLBcVZzv4jEzSdjAJCZ5GqiYrXHqbV4zfJji2dn7Ph7osJPaGTeVNO/0BtF+CP2JF/i4rS
u4/Xo041+Wd4xdE6HLVFpGmOQE8SbyFAH5LtWKyVAqkp9gbZhKxdkZvYNPlXNNFOZqfT8S
YCSGSkzDN62UGN+cWVQrzcT8SrdZzc7XVIFMS+wXONF0anrN405jQ40FB9o6MPEkJkQH8W
aZPWPOcwj/Q08+IThqnDfwCth79R01mzG0y+zKFkQd3AKkLC0oywEfqM+sMYDwNDm9/kng
w4DsYt4nkBuHXbrB9RIQsvhNG4jOVTNHPZmf4xSl4r73LHVnhvugkWVcrLGVQpfJsvz+Wt
UAAAdICT+PyAk/j8gAAAAHc3NoLXJzYQAAAgEAyzAOvprhmbeTv9HSW/5WiSD09o3wne2Q
IQs0eMAmRSHUq9ww+GMEfSnd6cxBRgIAs93VzUK0/bHbB+eo4Lb6pl5w1zCy8wwh8id9Hs
4Xqqg5ZkVxjBRaFB84ysn03RAbLOUbPkrC1bTia+OPa2GMHKZ88ooo79qJ3y2z7kod+l1t
0UG+Z0kHH1eQDfMWKM9OA54p0f+LxGPoGr48q8r5ypmxHk7i2jZqJTqyZ9TJqvI89KsT97
WOeRVzuZOVC15WjMPMf3x5TUl3cgNSyHU9qSWDSui0pMlSiciA/WBuzlKd6ICi9R7GUB0k
SajHt5zYkltgmEt7G1OegLBcVZzv4jEzSdjAJCZ5GqiYrXHqbV4zfJji2dn7Ph7osJPaGT
eVNO/0BtF+CP2JF/i4rSu4/Xo041+Wd4xdE6HLVFpGmOQE8SbyFAH5LtWKyVAqkp9gbZhK
xdkZvYNPlXNNFOZqfT8SYCSGSkzDN62UGN+cWVQrzcT8SrdZzc7XVIFMS+wXONF0anrN40
5jQ40FB9o6MPEkJkQH8WaZPWPOcwj/Q08+IThqnDfwCth79R01mzG0y+zKFkQd3AKkLC0o
ywEfqM+sMYDwNDm9/kngw4DsYt4nkBuHXbrB9RIQsvhNG4jOVTNHPZmf4xSl4r73LHVnhv
ugkWVcrLGVQpfJsvz+WtUAAAADAQABAAACAALtUpqnU4gQQqYfdfoFfD/ZRDFUrFP+5JxW
2eIooarne/W5cQRbAZw/lECDRNtz0u73mDyVPezVRfqWrHJt4GCxX2T4moaKzoBh1j9+Qf
```

在Jenkins中找到凭证管理



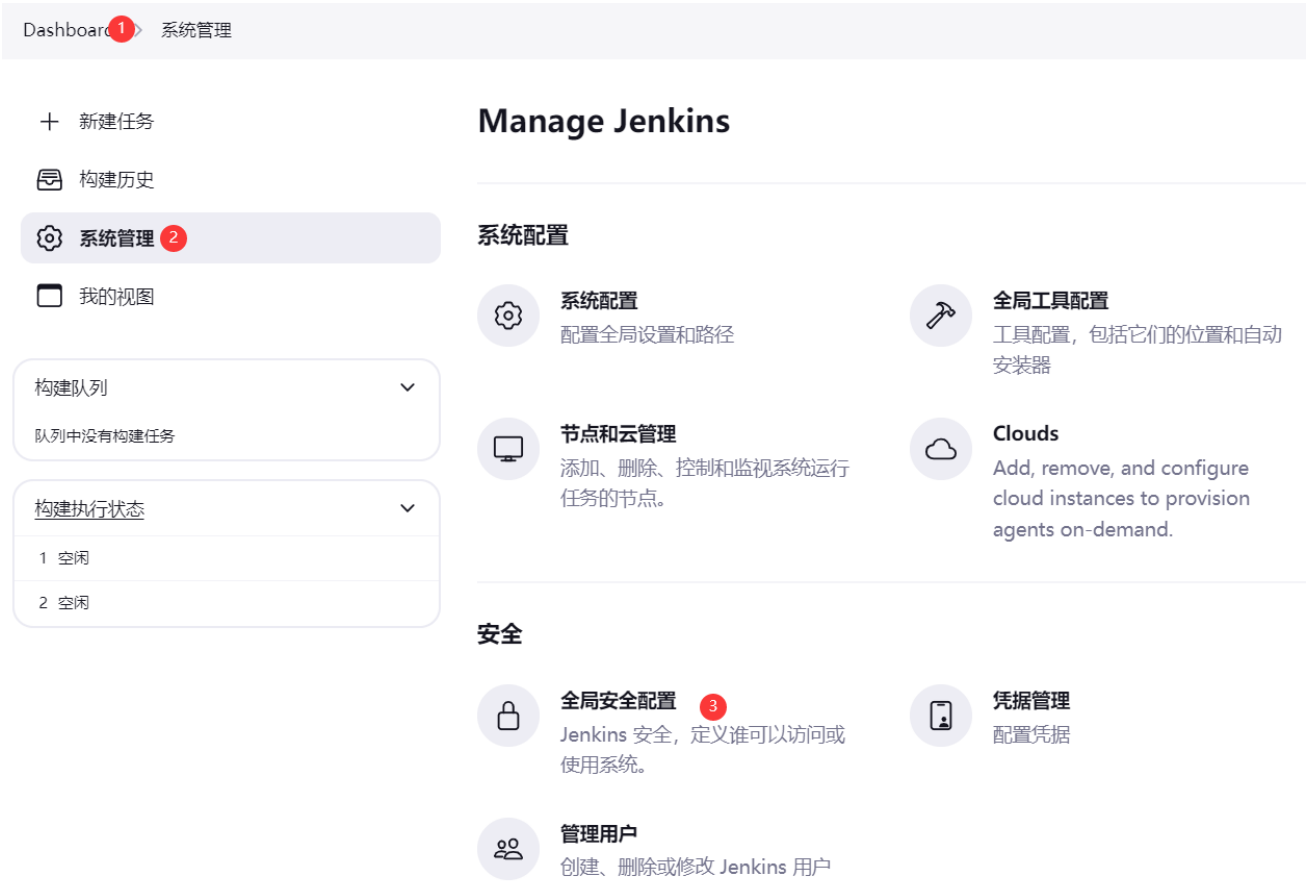
添加一个全局凭证

填写凭据内容

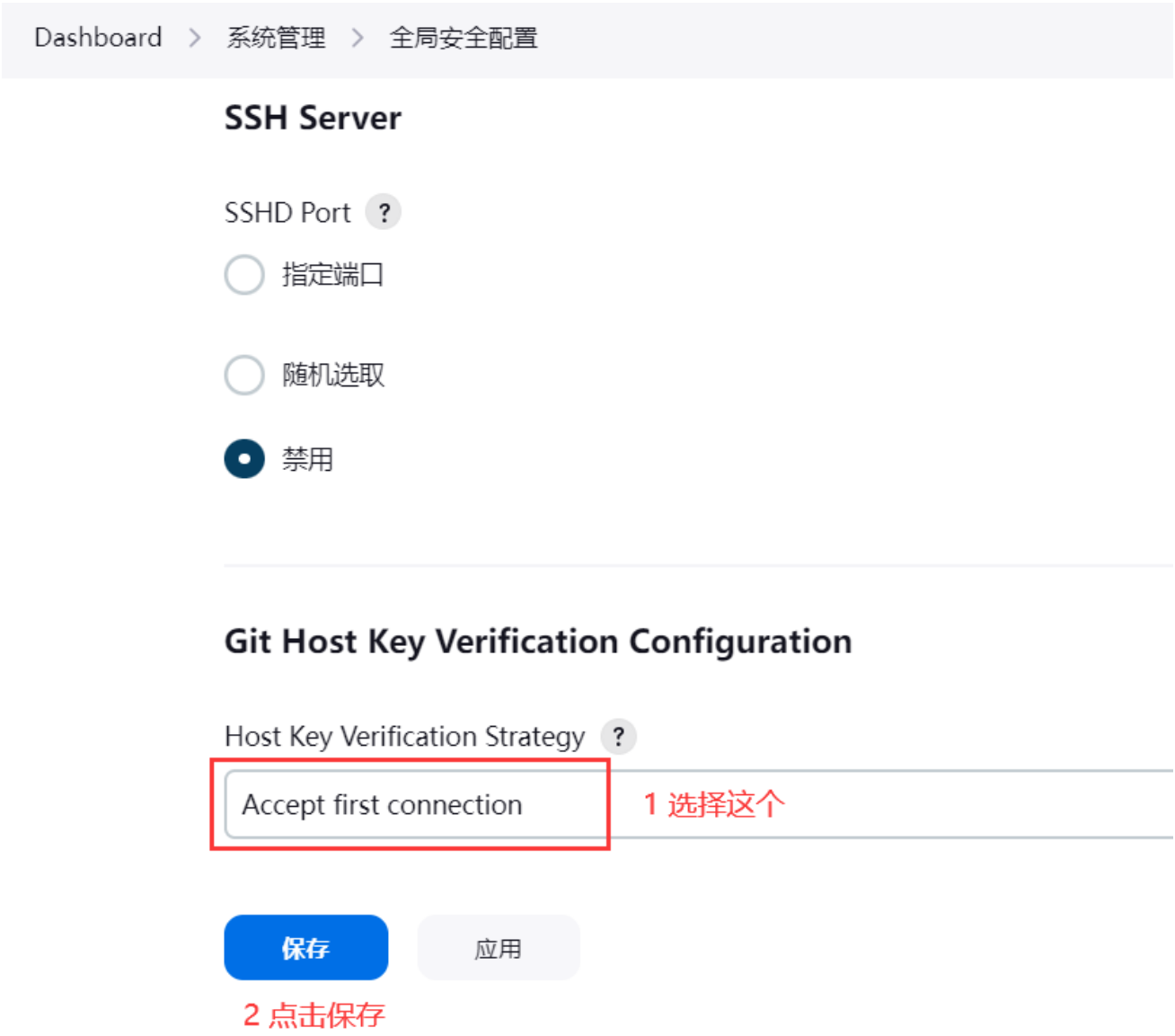


创建完成后可以在列表中看到

最后一步需要修改Git Host Key 验证策略，如下图所示。



进入设置页面后，修改下面的内容。



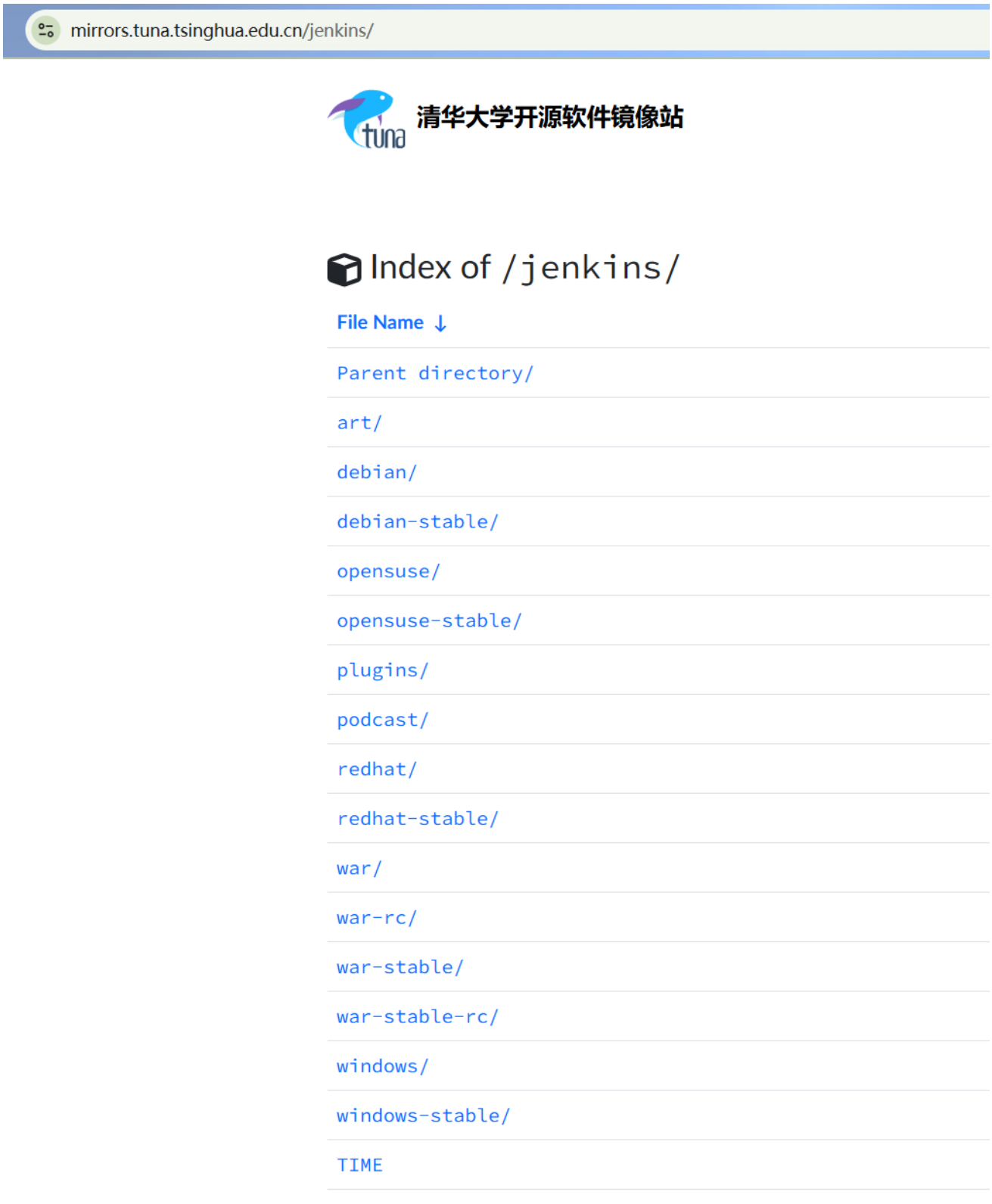
设置好了，后期就可以在任务中选择你的凭据了

6.3 手动部署插件（可选）

如果你使用Jenkens官方插件源下载插件，经常下载失败，这个时候可能你会想到可不可以修改插件源为国内的插件源来完成，这种方式现在已经不行了，你在网上搜索的镜像都已经无法使用，下表是常用的镜像源提供网站，但是很遗憾用不了。

镜像源	插件中心地址
清华大学镜像	`https://mirrors.tuna.tsinghua.edu.cn/jenkins/updates/update-center.json`
华为云镜像	`https://mirrors.huaweicloud.com/jenkins/updates/update-center.json`
腾讯云镜像	`https://mirrors.cloud.tencent.com/jenkins/updates/update-center.json`
阿里云镜像	`https://mirrors.aliyun.com/jenkins/updates/update-center.json`

你可以随便访问一个地址试试，已经无法访问了（华为云可以访问，但是它文件里面的下载地址依然是官方地址，所以改了当没改），为什么访问不了呢？我们可以观察一下这些镜像站的目录，以清华大学举例：



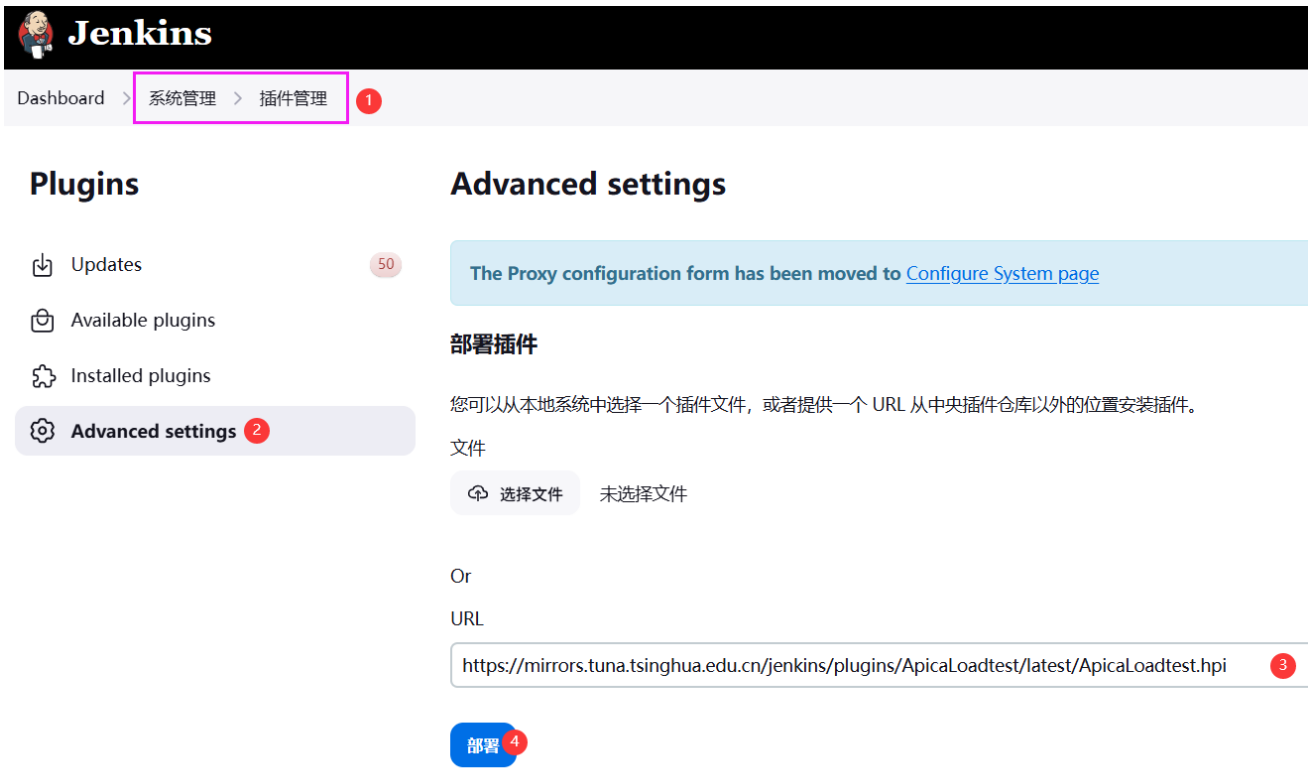
观察图中是不是已经没有updates目录了，所以网上的镜像已经无法使用了。

镜像源无法使用了，我们没有魔法的情况怎么去安装插件呢？我们可以通过手动指定下载地址的方式来安装，操作步骤如下：

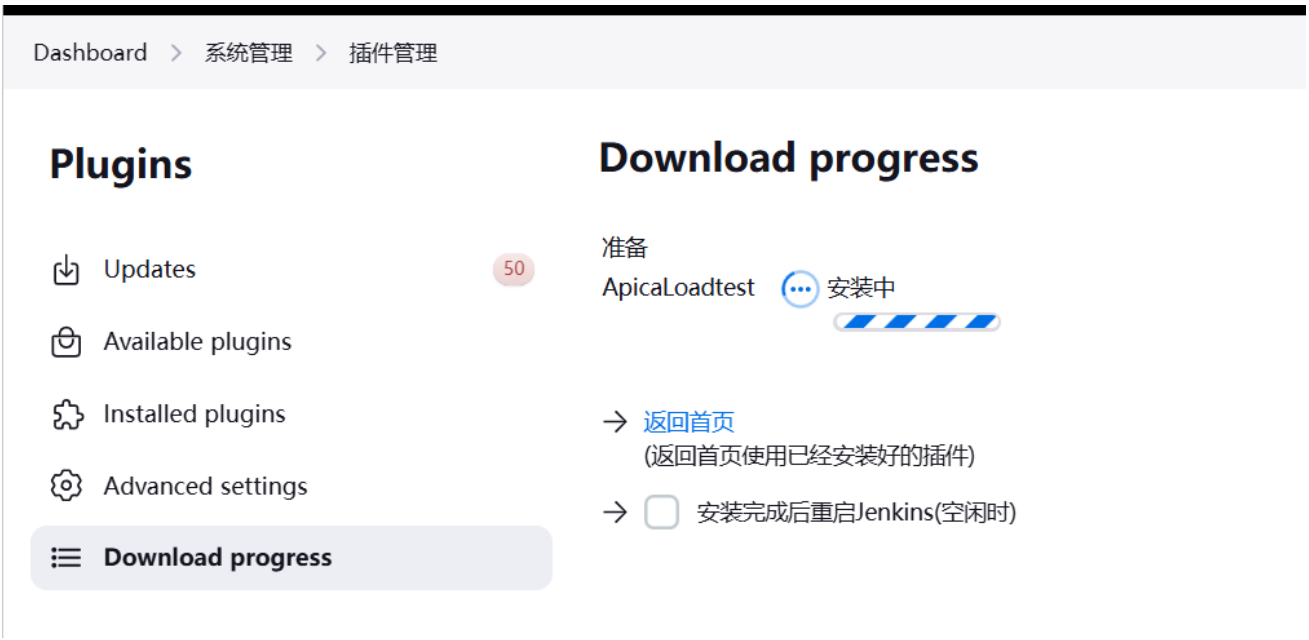
通过任意镜像站进入plugins目录，然后在里面找到你要安装的插件，然后复制插件的地址

提示：选择插件的时候要注意版本，有些插件可能会对Jenkens有版本要求，可以到插件的官网看看版本对应要求

首先进入 **插件管理**，然后找到 **高级设置**

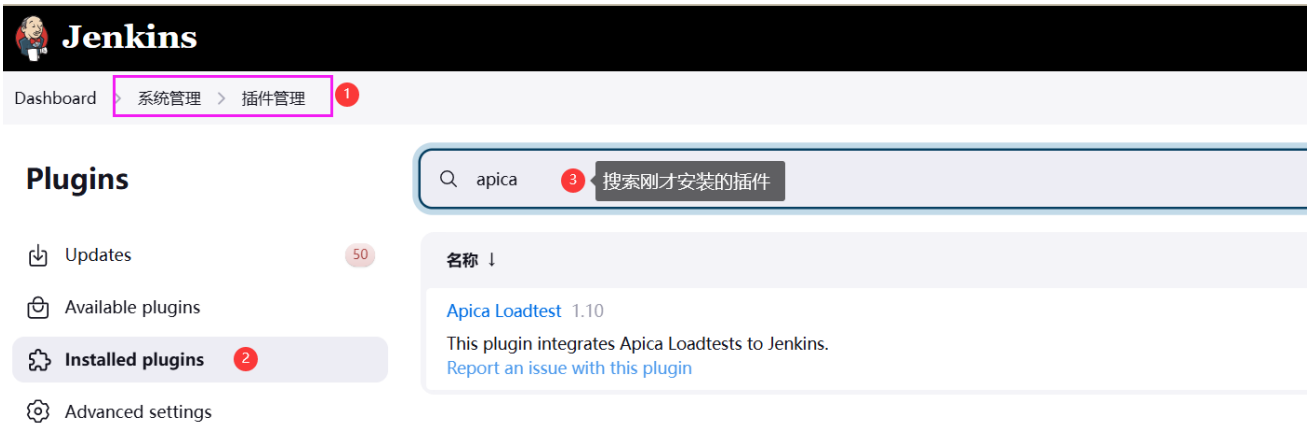


点击部署后，就会通过镜像网站去下载插件



安装完成后

然后重启一下容器，容器重启完成后，看看是否安装成功



如果你某些插件安装不了，可以用这种方式来安装。