

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

```
1 mkdir /home/jenkins
2 cd /home/jenkins
3 mkdir jenkins_home
4 # 把当前目录的拥有者赋值给uid 1000
5 chown -R 1000 jenkins_home
6 # 创建maven仓库文件目录
7 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

```
1 # 开放端口
2 firewall-cmd --add-port 8380/tcp --permanent
3 # 重新加载防火墙
4 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。

```
1 version: '3'
2 services:
3   jenkins:
4     image: jenkins/jenkins:lts
5     container_name: jenkins
6     privileged: true
7     user: root
8     environment:
9       TZ: Asia/Shanghai
10    ports:
11      - "8380:8080"
12      - "50000:50000"
13    volumes:
14      - /home/jenkins/jenkins_home:/var/jenkins_home
15      - /home/jenkins/maven/m2:/root/.m2
16      - /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 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 | bb75ca090d1f4694991fd2a9e4cb9e7b 1 这是管理员初始密码
jenkins | This may also be found at: /var/jenkins_home/secrets/initialAdminPassword
jenkins | *****
jenkins | *****
jenkins | *****
```

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 | bb75ca090d1f4694991fd2a9e4cb9e7b
jenkins | This may also be found at: /var/jenkins_home/secrets/initialAdminPassword
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 file
jenkins | 2024-07-11 12:52:36.916+0000 [id=47] INFO hudson.util.Retrier#start: Performed the action check updates server
jenkins | ^CGracefully stopping... (press Ctrl+C again to force)
jenkins | [+] Running 1/1
jenkins | # Container jenkins Stopped
jenkins | 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

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。

选择推荐安装

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

<input checked="" type="checkbox"/> Folders	<input checked="" type="checkbox"/> OWASP Markup Formatter	<input type="checkbox"/> Build Timeout	<input type="checkbox"/> Credentials Binding	** Ionicons API
<input checked="" type="checkbox"/> Timestampper	<input checked="" type="checkbox"/> Workspace Cleanup	<input type="checkbox"/> Ant	<input checked="" type="checkbox"/> Gradle	Folders
<input checked="" type="checkbox"/> Pipeline	<input checked="" type="checkbox"/> GitHub Branch Source	<input checked="" type="checkbox"/> Pipeline: GitHub Groovy Libraries	<input checked="" type="checkbox"/> Pipeline Graph View	OWASP Markup Formatter
<input type="checkbox"/> Git	<input type="checkbox"/> SSH Build Agents	<input checked="" type="checkbox"/> Matrix Authorization Strategy	<input type="checkbox"/> PAM Authentication	
<input type="checkbox"/> LDAP	<input checked="" type="checkbox"/> Email Extension	<input type="checkbox"/> Mailer	<input type="checkbox"/> Dark Theme	
<input type="checkbox"/> Localization: Chinese (Simplified)				
				** - 需要依赖

Jenkins 2.452.3

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

创建第一个管理员用户

用户名

admin

密码

.....

确认密码

.....

全名

administrator

电子邮件地址

Jenkins 2.452.3

使用admin账户继续

保存并完成

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。

实例配置

注意：看看ip地址是否对应

Jenkins URL:

http://192.168.220.128:8380/

Jenkins URL 用于给各种Jenkins资源提供绝对路径链接的根地址。这意味着对于很多Jenkins特色是需要正确设置的，例如：邮件通知、PR状态更新以及提供给构建步骤的BUILD_URL环境变量。

推荐的默认值显示在尚未保存，如果可能的话这是根据当前请求生成的。最佳实践是要设置这个值，用户可能会需要用到。这将会避免在分享或者查看链接时的困惑。

Jenkins 2.452.3

现在不要

保存并完成

点击完成后，进入就绪页面。

Jenkins已就绪!

Jenkins安装已完成。

开始使用Jenkins

点击开始使用，进入系统。

欢迎来到 Jenkins!

This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project.

Start building your software project

Create a job



Set up a distributed build

Set up an agent



Configure a cloud

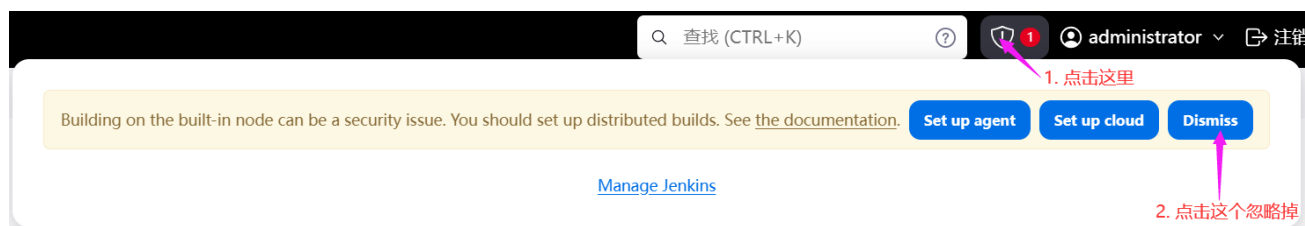


Learn more about distributed builds



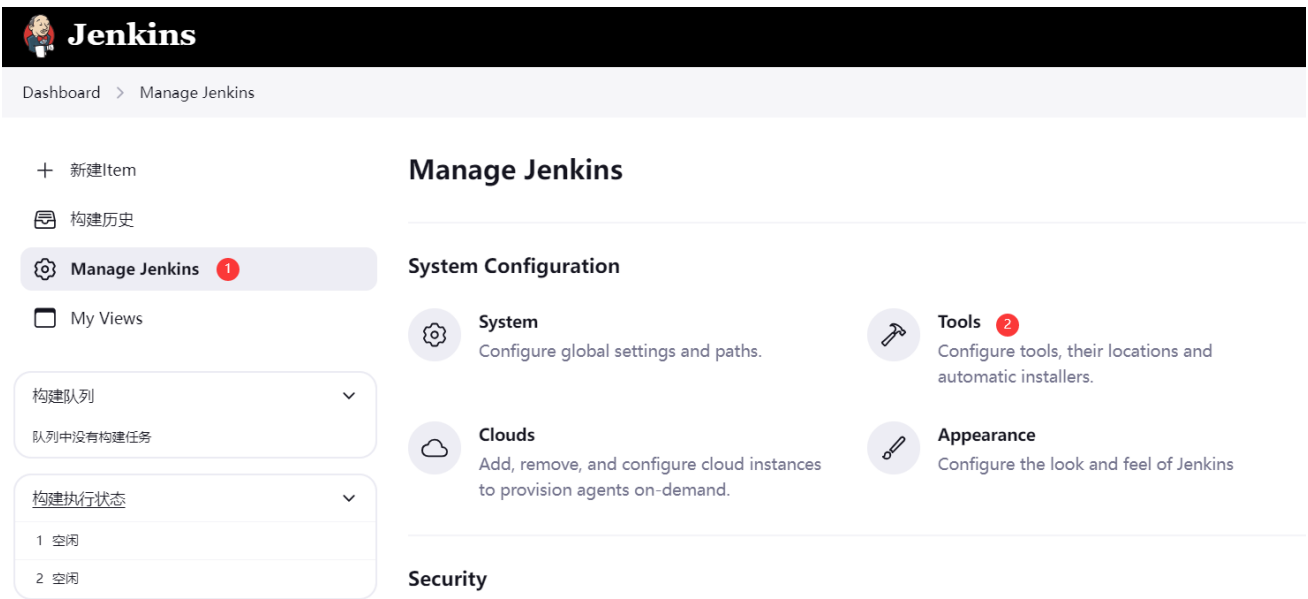
到此服务器初始完成，可以使用了。

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

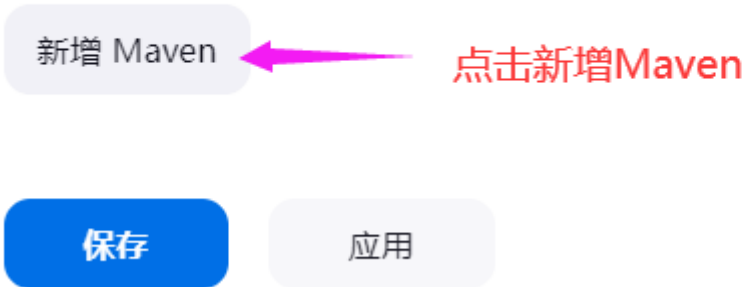
通过下图所示的路径可以找到全局工具配置



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

Maven 安装



输入Maven版本信息

Maven 安装

新增 Maven

≡ Maven

Name

maven3.6.3 1 输入名称

2 自动安装 ?

≡ 从 Apache 安装

版本

3.6.3 3 选择版本

新增安装

新增 Maven

保存 4 保存配置

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

```
1 cd /home/jenkins/jenkins_home
2 mkdir jdk
3 cd jdk
```

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 jdk8
[root@localhost jenkins_home]# cd jdk8/
[root@localhost jdk8]# wget https://d10.injdc.cn/openjdk/oraclejdk/8/jdk-8u421-linux-x64.tar.gz
--2025-06-19 17:44:35-- https://d10.injdc.cn/openjdk/oraclejdk/8/jdk-8u421-linux-x64.tar.gz
正在解析主机 d10.injdc.cn (d10.injdc.cn)... 110.42.44.84
正在连接 d10.injdc.cn (d10.injdc.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 jdk8]# ls
jdk-8u421-linux-x64.tar.gz
[root@localhost jdk8]# pwd
/home/jenkins/jenkins_home/jdk8
[root@localhost jdk8]#

```

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 jdk8]# ls
jdk1.8.0_421  jdk-8u421-linux-x64.tar.gz
[root@localhost jdk8]#

```

提示：此时jdk8对应容器里面的目录是 `/var/jenkins_home/jdk8/`

然后在全局工具配置中，新增JDK配置

Jenkins

Dashboard > 系统管理 > 全局工具配置 1

全局工具配置

Maven 配置

默认 settings 提供

使用默认 Maven 设置

默认全局 settings 提供

使用默认 Maven 全局设置

JDK 安装

新增 JDK 2

然后按照下图方式输入数据

≡ JDK

别名

java8 输入名称

JAVA_HOME

/var/jenkins home/jdks/jdk1.8.0 421

☐ 自动安装 ? 把自动安装去掉

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

```
[root@localhost jdks]# ls
jdk-17.0.12  jdk1.8.0_421
[root@localhost jdks]#
```

新增 JDK

≡ JDK

别名

java8

JAVA_HOME

/var/jenkins home/jdks/jdk1.8.0 421

☐ 自动安装 ?

≡ JDK

别名

java17

JAVA_HOME

/var/jenkins home/jdks/jdk-17.0.12

☐ 自动安装 ?

新增 JDK

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

General

- 源码管理
- 构建触发器
- 构建环境
- Build Steps
- 构建后操作

纯文本 预览

- ☐ GitHub 项目
- ☐ Throttle builds ?
- ☐ 丢弃旧的构建 ?
- ☐ 参数化构建过程 ?
- ☐ 在必要的时候并发构建 ?

JDK

项目使用的JDK

(System)

(System)

java8

java17

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

The screenshot shows the Jenkins 'Manage Jenkins' interface. On the left sidebar, 'Manage Jenkins' is selected with a red notification badge. The main content area is titled 'Manage Jenkins' and contains a 'System Configuration' section. Within this section, the 'Plugins' option is highlighted, showing a description: 'Add, remove, disable or enable plugins that can extend the functionality of Jenkins.' A red badge with the number '2' is next to the 'Plugins' icon. Other options like 'System', 'Tools', 'Nodes', 'Clouds', and 'Appearance' are also visible. At the bottom, there is a 'Security' section. The top navigation bar includes the Jenkins logo, a search bar, and user information.

选择可用插件选项，然后输入搜索关键词

Plugins

Updates

Available plugins 1

Installed plugins

Advanced settings

Download progress

ssh 2 输入搜索关键词

安装 名称 Released

SSH 2.6.1

Build Wrappers

This plugin executes shell commands remotely using SSH protocol.

Warning: This plugin version may not be safe to use. Please review the following security notices:

- CSRF vulnerability and missing permission checks allow capturing credentials
- Missing permission check allows enumerating credentials IDs

6 年 3 月 ago

JSch dependency 0.2.16-86.v42e010d9484b_

Library plugins (for use by other plugins) Miscellaneous

Jenkins plugin that brings the JSch library as a plugin dependency, and provides an SSHAuthenticatorFactory for using JSch with the ssh-credentials plugin.

6 月 9 天 ago

SSH server 3.330.vc866a_8389b_58

Adds SSH server functionality to Jenkins, exposing CLI commands through it.

1 月 1 天 ago

Publish Over SSH 1.25

Uploaders Build Tools

Send build artifacts over SSH

1 年 0 月 ago

下面是安装过程页面

Dashboard > Manage Jenkins > 插件管理

Plugins

Updates

Available plugins

Installed plugins

Advanced settings

Download progress

Matrix Authorization Strategy	完成
PAM Authentication	完成
LDAP	完成
Email Extension	完成
Mailer	完成
Theme Manager	完成
Dark Theme	完成
Localization Support	完成
Localization: Chinese (Simplified)	完成
Loading plugin extensions	Success
Oracle Java SE Development Kit Installer	等待
SSH server	等待
Command Agent Launcher	等待
Infrastructure plugin for Publish Over X	等待
JSch dependency	等待
Publish Over SSH	等待
Loading plugin extensions	Pending

等待下载完成

→ 返回首页
(返回首页使用已经安装好的插件)

下载完成后，点击勾选重启Jenkins

Oracle Java SE Development Kit Installer	✓ 完成
SSH server	✓ 完成
Command Agent Launcher	✓ 完成
Infrastructure plugin for Publish Over X	✓ 完成
JSch dependency	✓ 完成
Publish Over SSH	✓ 完成
Loading plugin extensions	✓ Success

→ [返回首页](#)

(返回首页使用已经安装好的插件)

→ ☐ 安装完成后重启Jenkins(空闲时)

下载完成，勾选它重启Jenkins

勾选上之后，进入关闭提示界面



Jenkins 正在重启，请稍等 ...

当 Jenkins 就绪的时候，你的浏览器会自动刷新。

Safe Restart

Builds on agents can usually continue.

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。

```
1 | 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



Sign in to Jenkins

用户名

admin ¹

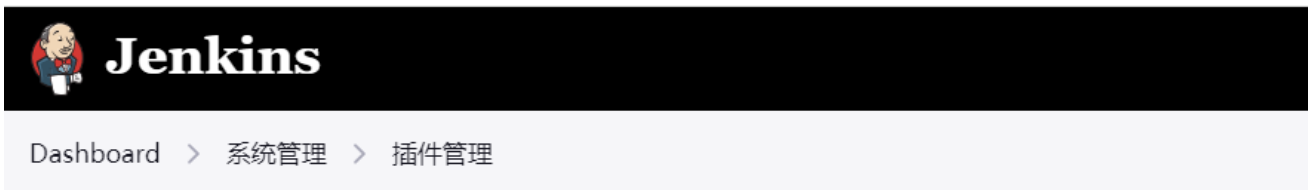
密码

..... ²

☒ ³ 保持登录状态

⁴ 登录

输入登录信息点击登录后，会进入到刚才的下载界面



Plugins

📄 Updates

📁 Available plugins

⚙️ Installed plugins

⚙️ Advanced settings

Download progress

这里为空，表示没有插件再进行更新

→ [返回首页](#)

(返回首页使用已经安装好的插件)

→ ☐ 安装完成后重启Jenkins(空闲时)

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。



Dashboard 1 系统管理

+ 新建任务

📁 构建历史

⚙️ 系统管理 2

📄 我的视图

构建队列

Manage Jenkins

系统配置



系统配置 3
配置全局设置和路径



全局工具配置
工具配置，包括它们的位置和自动安装器

找到SSH Server



Dashboard > 系统管理 > System 1

System

主目录 ?

By default, Jenkins stores its data in the

/var/jenkins_home

系统消息

This message will be c

纯文本 预览

执行器数量

2

标签

Timestampers

Fingerprints

管理监控配置

Global Build Discarders

Default notification URL

GitHub

GitHub API usage

GitHub Enterprise Servers

Global Trusted Pipeline Libraries

Global Untrusted Pipeline Libraries

Build-timeout Plugin > BuildStep Action

Git plugin

Shell

Extended E-mail Notification

邮件通知

Publish over SSH 2

SSH Servers

新增

点击新增

高级

▼

保存

应用

输入信息

SSH Servers

SSH Server

Name ?

vm-linux

1

输入名称

Hostname ?

192.168.220.128

2

服务器ip地址

Username ?

root

3

登录服务器使用用户

Remote Directory ?

/

4

登录后默认进入服务器目录

☐ Avoid sending files that have not changed ?

高级

▼

Test Configuration

5

点击测试连接

新增

此时应该是连接失败

高级

▼

Failed to connect or change directory

Test Configuration

jenkins.plugins.publish_over.BapPublisherException: Failed to connect and initialize SSH connection. Message: [Failed to connect session for config [vm-linux]. Message [Auth fail for methods 'publickey,gssapi-keyex,gssapi-with-mic,password']]

点击高级输入密码

Remote Directory ?

/

☐ Avoid sending files that have not changed ?

高级 ¹ Edited

☒ ² Use password authentication, or use a different key ?

Passphrase / Password ?

..... ³ 输入密码

Path to key ?

测试与保存

Remote Directory ?

/

☐ Avoid sending files that have not changed ?

高级 ⁵ Success 显示成功

Test Configuration ⁴ 再次点击测试

新增

高级 ⁷ 点击保存退出界面

保存 应用 ⁶ 点击应该保存一下

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。**

Configure

- General
- 源码管理
- 构建触发器
- 构建环境
- Build Steps ¹**
- 构建后操作

Build Steps

Send files or execute commands over SSH ? ²

SSH Publishers

SSH Server

Name ?

vm-linux

这里可以选择主机 ³

高级 ³

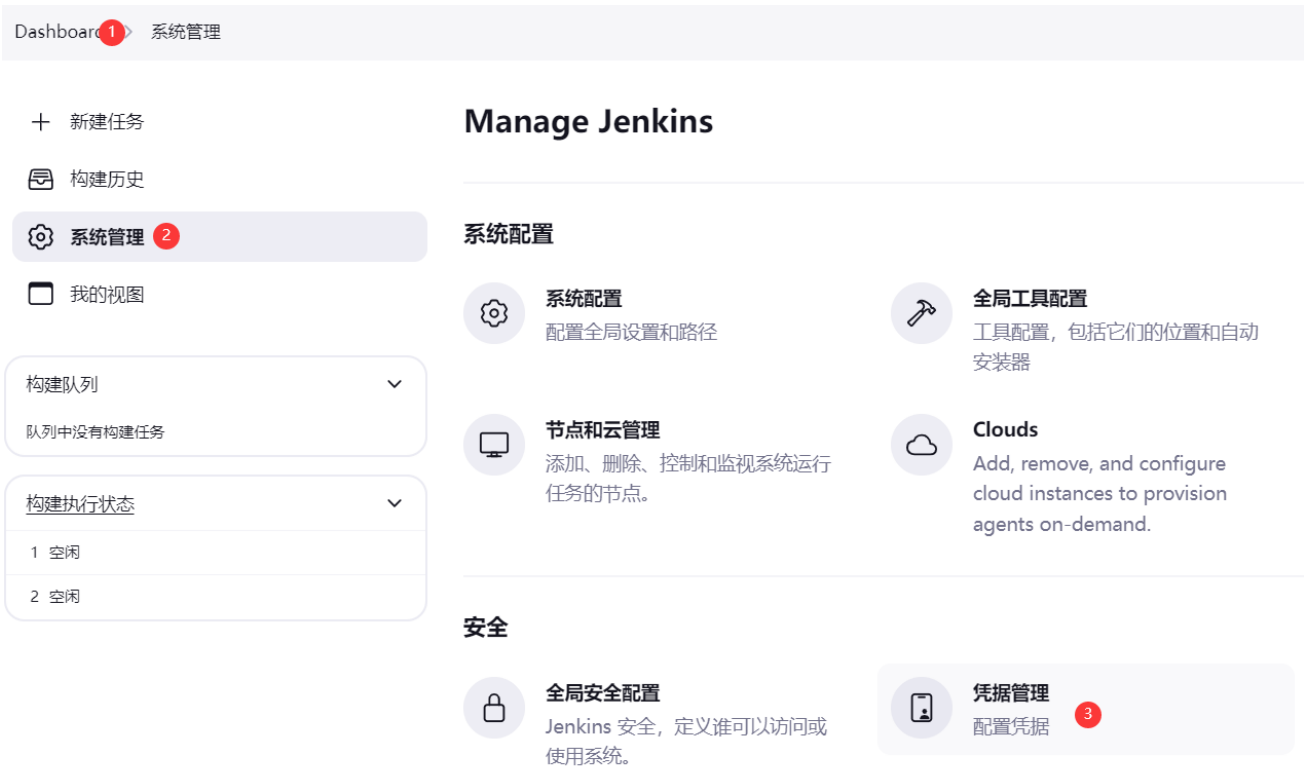
6 其他配置

6.1 添加凭据

项目一般会托管到gitee、github、gitlab等git托管平台上，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

New credentials

类型

Username with password

1

选择用户名密码类型的凭据

v

范围 ?

全局 (Jenkins, nodes, items, all child items, etc)

2

选择全局范围

v

用户名 ?

3

输入托管平台的用户名

☒ Treat username as secret ?

4

勾选上

密码 ?

.....

5

输入托管平台的密码

ID ?

描述 ?

阿伟gitee凭据

6

输入凭据描述，最好带上平台和用户名，方便你再使用的时候选择识别

Create

点击创建

创建成功后会返回凭据列表，在这里能够看到你的凭据

全局凭据 (unrestricted)

+ Add Credentials

Credentials that should be available irrespective of domain specification to requirements matching.

ID	名称	类型	描述
 c760e834-cf2c-4e1d-aa88-f8c014110f16	阿伟gitee凭据	Username with password	阿伟gitee凭据 

图标 小 中 大

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

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

使用ssh-keygen命令生成密钥

```
1 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 "2060602470@qq.com"
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa): 文件保存的目录
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:Bw7IuxdSALfkWCL29Ve8eR0dNvvJt14QFzglkt+8AFY 2060602470@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]-----+
```

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

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

```
root@65b146725084:/# cat /root/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaClyc2EAAAADAQABAAQDLMA6+muGZt5O/0dJb/laJIPT2jfCd7ZAhCzR4wCZFIIdSr3DD4YwR9Kd3pzEFGAgCz3d
nzyiijv2onfLbPush36XW3Rqb5nSQcfV5AN8xYoz04DninR/4vEY+gavjyryvnKmbEeTuLaNmOlOrJnlMmq8jz0qxP3tY55FXO5k5ULXLa
/iMTNJ2MAkJnkaqJitceptXjN8mOLZ2fs+Huiwk9oZN5U07/QG0X4I/YkX+LitK7j9ejTiX5Z3jF0ToctUWkaY5ATxJvIUAfkulYrJUCqS
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

Add new SSH Key

Title

Jenkins

1 填写名称，可以随意起名，但是要方便你识别

Key type

Authentication Key

Key

Begins with 'ssh-rsa', 'ecdsa-sha2-nistp256', 'ecdsa-sha2-nistp384', 'ecdsa-sha2-nistp521', 'ssh-ed25519', 'sk-ecdsa-sha2-nistp256@openssh.com', or 'sk-ssh-ed25519@openssh.com'

2 将刚才复制的公钥粘贴到这里

Add SSH key

3 点击按钮完成添加

添加成功后

SSH keys

New SSH key

This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.

Authentication keys



awei-windows

SHA256: a0+CjdDEWRIGIZuROCABIHvRha6ZHeffDshMwMxB72I

Added on Mar 6, 2023

Last used within the last 2 weeks — Read/write

Delete



Jenkins

SHA256: Bw7IuxdSALfkWCL29Ve8eR0dNvvJt14QFzg1kt+8AFY

Added on Jul 12, 2024

Never used — Read/write

Delete

Check out our guide to [connecting to GitHub using SSH keys](#) or troubleshoot [common SSH problems](#).

连接测试

在Jenkins容器里面执行命令

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

执行结果如如下图所示

```
root@65b146725084:/# ssh -T git@github.com
The authenticity of host 'github.com (20.205.243.166)' can't be established.
ED25519 key fingerprint is SHA256:+DiY3wvvV6TuJJhbpZisF/zLDA0zPMSvHdkr4UvCOqU.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'github.com' (ED25519) to the list of known hosts.
Hi zero-awei! You've successfully authenticated, but GitHub does not provide shell access.
```

输入yes

6.2.3 管理凭证

首先查看私钥，执行下面命令

```
1 | cat /root/.ssh/id_rsa
```

```
root@65b146725084:/# cat /root/.ssh/id_rsa
-----BEGIN OPENSSH PRIVATE KEY-----
b3BlbnNzaC1rZXktdjEAAAABAG5vbmUAAAABbm9uZQAAAAAAAAABAAACFwAAAAAdzc2gtcn
NhAAAAAwEAAQAAAgEAYzAOvprhmbeTv9HSW/5WiSD09o3wne2QIQs0eMAMRSHUq9ww+GME
fSnd6cxBRgIAS93VzUK0/bHbB+eo4Lb6p15w1zCy8wwh8id9Hs4Xqqg5ZkVxjBRaFB84ys
n03RabLOUbPkrC1bTia+OPa2GMHKZ88ooo79qJ3y2z7kod+l1t0UG+Z0kHH1eQDfMWKM9O
A54p0f+LxGPoGr48q8r5ypmxHk7i2jZqJTqyZ9TJqvI89KsT97WOeRVzuZOVC15WjMPMf3
x5TU13cgNSyHU9qSWDSui0pMlSicia/WBuzlKd6ICi9R7GUB0kSajHt5zYkltgmEt7G1Oe
gLBcVZzv4jEzSdjAJCZ5GqiYrXHqbV4zfJji2dn7Ph7osJPaGTeVNO/0BtF+CP2JF/i4rS
u4/Xo04l+wd4xdE6HLVFPgMQE8SbyFAH5LtWKyVAqkp9gbZhKxdkZvYNPlXNNFOZqfT8S
YCSGSkzDN62UGN+cWVQrzcT8SrdZzc7XVIFMS+wXONF0anrN405jQ40FB9o6MPEkJKQH8W
aZPWPOcwj/Q08+IthqnDfwCth79R0lmzG0y+zKfkQd3AKkLC0oywEfQM+sMYDwNDm9/kng
w4DsYt4nkBuHXbrB9RIQsvhNG4jOVTNHPZmf4xSl4r73LHVnhvugkWVcrLGVQpfJsvz+Wt
UAAAdICT+PyAk/j8gAAAAHc3NoLXJzYQAAAgEAYzAOvprhmbeTv9HSW/5WiSD09o3wne2Q
IQs0eMAMRSHUq9ww+GMEfSnd6cxBRgIAS93VzUK0/bHbB+eo4Lb6p15w1zCy8wwh8id9Hs
4Xqqg5ZkVxjBRaFB84ysn03RabLOUbPkrC1bTia+OPa2GMHKZ88ooo79qJ3y2z7kod+l1t
0UG+Z0kHH1eQDfMWKM9OA54p0f+LxGPoGr48q8r5ypmxHk7i2jZqJTqyZ9TJqvI89KsT97
WOeRVzuZOVC15WjMPMf3x5TU13cgNSyHU9qSWDSui0pMlSicia/WBuzlKd6ICi9R7GUB0k
SajHt5zYkltgmEt7G1OegLBcVZzv4jEzSdjAJCZ5GqiYrXHqbV4zfJji2dn7Ph7osJPaGT
eVNO/0BtF+CP2JF/i4rSu4/Xo04l+wd4xdE6HLVFPgMQE8SbyFAH5LtWKyVAqkp9gbZhK
xdkZvYNPlXNNFOZqfT8SYCSGSkzDN62UGN+cWVQrzcT8SrdZzc7XVIFMS+wXONF0anrN40
5jQ40FB9o6MPEkJKQH8WaZPWPOcwj/Q08+IthqnDfwCth79R0lmzG0y+zKfkQd3AKkLC0o
ywEfQM+sMYDwNDm9/kngw4DsYt4nkBuHXbrB9RIQsvhNG4jOVTNHPZmf4xSl4r73LHVnhv
ugkWVcrLGVQpfJsvz+WtUAAADAQABAAACAAltUpqnu4gQQYfdfoFfD/ZRDFUrFP+5JxW
2eIooarne/W5cQRbAZw/1ECDRntz0u73mDyVPezVRfqWRHJt4GCxX2T4moaKzoBh1j9+Qf
```

在Jenkins中找到凭证管理

Dashboards 1 系统管理

+ 新建任务

构建历史

系统管理 2

我的视图

构建队列

队列中没有构建任务


构建执行状态


1 空闲


2 空闲


Manage Jenkins

系统配置


 **系统配置**
配置全局设置和路径


 **节点和云管理**
添加、删除、控制和监视系统运行任务的节点。

 **全局工具配置**
工具配置，包括它们的位置和自动安装器

 **Clouds**
Add, remove, and configure cloud instances to provision agents on-demand.

安全

 **全局安全配置**
Jenkins 安全，定义谁可以访问或使用系统。

 **凭证管理**
配置凭证 3

添加一个全局凭证

Dashboard > 系统管理 > 凭据

凭据

类型	提供者	存储 ↓	域	唯一标识	名称
----	-----	------	---	------	----

Stores scoped to Jenkins

提供者	存储 ↓	域
	System	全局 ¹

添加凭据 ²

图标: 小 中 大

填写凭据内容

New credentials

类型

SSH Username with private key ^{1 选择SSH类型}

范围 ?

全局 (Jenkins, nodes, items, all child items, etc) ^{2 范围选择全局}

ID ?

描述 ?

阿伟github-ssh ^{3 输入描述, 如: xxx-github-ssh}

Username

anyuser ^{4 输入用户名, 如: anyuser}

☐ Treat username as secret ?

Private Key

☒ Enter directly

Key





在下面输入新秘文

-----BEGIN OPENSSH PRIVATE KEY-----
b3B1bnNzaC1rZXktdjEAAAABG5vbmUAAAABbm9uZQAAAAAAAAABAAACFwAAAAAdzc2gtcn
NhAAAAAwEAAQAAAEAyZA0vprhmbEty9HSW/5WiSD09o3wne2QIQs0eMamRSHUq9ww+GME
fSnd6cxBRgIAs93VzUK0/bHbB+eo4Lb6p15w1zCy8wwh8id9Hs4Xqg5ZkVxjBRaFB84ys
n03RabLOUbPkrC1bTia+0Pa2GMHKZ88ooo79qJ3y2z7kod+11t0UG+Z0kHH1eQDfMWMf90
A54n0f+i xGPoGr48n8r5vrmxHk7i? i7n ITnv79T Invt89K<T97W0eRVzu70VC15W iMPMf3 ^{5 输入刚才复杂的私钥}

Create ^{6 点击完成创建}

创建完成后可以在列表中看到

Credentials that should be available irrespective of domain specification to requirements matching.


ID	名称	类型	描述
 c760e834-cf2c-4e1d-aa88-f8c014110f16	阿伟gitee凭据	Username with password	阿伟gitee凭据 
 6d3318e9-97f1-435f-ba85-2f02091a12dc	anyuser (阿伟github-ssh)	SSH Username with private key	阿伟github-ssh 


图标: 小 中 大


最后一步需要修改Git Host Key 验证策略，如下图所示。

Dashboard 1 > 系统管理

+ 新建任务

 构建历史

 系统管理 2

 我的视图

构建队列

队列中没有构建任务


构建执行状态


1 空闲


2 空闲


Manage Jenkins


系统配置


 **系统配置**
配置全局设置和路径


 **节点和云管理**
添加、删除、控制和监视系统运行任务的节点。

 **全局安全配置** 3
Jenkins 安全，定义谁可以访问或使用系统。

 **管理用户**
创建、删除或修改 Jenkins 用户

 **全局工具配置**
工具配置，包括它们的位置和自动安装器

 **Clouds**
Add, remove, and configure cloud instances to provision agents on-demand.

 **凭据管理**
配置凭据

进入设置页面后，修改下面的内容。

SSH Server

SSHD Port ?

☐ 指定端口☐ 随机选取☒ 禁用

Git Host Key Verification Configuration

Host Key Verification Strategy ?

☒ Accept first connection

1 选择这个

保存

应用

2 点击保存

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Jenkins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：

 mirrors.tuna.tsinghua.edu.cn/jenkins/

 **清华大学开源软件镜像站**

 **Index of /jenkins/**

File Name ↓
Parent directory/
art/
debian/
debian-stable/
opensuse/
opensuse-stable/
plugins/
podcast/
redhat/
redhat-stable/
war/
war-rc/
war-stable/
war-stable-rc/
windows/
windows-stable/
TIME

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Jenkins有版本要求，可以到插件的官网看看版本对应要求



清华大学开源软件镜像站

Index of /jenkins/plugins/ApicaLoadtest/latest/

File Name ↓

File Size ↓

Parent directory/

-

ApicaLoadtest.hpi

复制hpi文件下载地址

238.0 KiB

首先进入 **插件管理**，然后找到 **高级设置**

Jenkins

Dashboard > 系统管理 > 插件管理 1

Plugins

Updates 50

Available plugins

Installed plugins

Advanced settings 2

Advanced settings

The Proxy configuration form has been moved to [Configure System page](#)

部署插件

您可以从本地系统中选择一个插件文件，或者提供一个 URL 从中央插件仓库以外的位置安装插件。

文件

选择文件 未选择文件

Or

URL

<https://mirrors.tuna.tsinghua.edu.cn/jenkins/plugins/ApicaLoadtest/latest/ApicaLoadtest.hpi> 3

部署 4

点击部署后，就会通过镜像网站去下载插件

Plugins

📄 Updates

📁 Available plugins

⚙️ Installed plugins

⚙️ Advanced settings

☰ Download progress

Download progress

准备

ApicaLoadtest ... 安装中

→ [返回首页](#)

(返回首页使用已经安装好的插件)

→ ☐ 安装完成后重启Jenkins(空闲时)

安装完成后



Jenkins

Plugins

📄 Updates

📁 Available plugins

⚙️ Installed plugins

⚙️ Advanced settings

☰ Download progress

Download progress

准备

ApicaLoadtest ✓ 完成

→ [返回首页](#)

(返回首页使用已经安装好的插件)

→ ☐ 安装完成后重启Jenkins(空闲时)

然后重启一下容器，容器重启完成后，看看是否安装成功



Jenkins

Plugins

📄 Updates

📁 Available plugins

⚙️ Installed plugins

⚙️ Advanced settings

🔍 apica

搜索刚才安装的插件

名称 ↓

Apica Loadtest 1.10

This plugin integrates Apica Loadtests to Jenkins.

[Report an issue with this plugin](#)

如果你某些插件安装不了，可以用这种方式来安装。