

Docker部署Jenkins

参考链接: <https://www.jenkins.io/zh/doc/book/installing/>

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: jenkinsci/blueocean
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]# ls
docker-compose.yml  jenkins_home
[root@localhost jenkins]# docker-compose up
[+] Running 0/15
.: jenkins Pulling
  2408cc74d12b Downloading
    bf6bdadd7450 Downloading [=====>] 4.056MB/32.71MB
  9a122f85f3c9 Download complete
  aea48bee21ff Download complete
  e2cfb1579085 Downloading [==>] 4.359MB/90.98MB
  6ed48a3271c3 Download complete
  e4f28e1b9f4e Waiting
  78f4f1e35669 Waiting
  3968a4384d5c Waiting
  034428d37519 Waiting
  f7f0bf5c5bf2 Waiting
  2ef560e709b6 Waiting
  23bd89992269 Waiting
```

如果看到类似下面的结果表示服务器正常启动。

```
jenkins | 2022-08-09 06:38:29.254+0000 [id=30] INFO jenkins.InitReactorRunner$1#onAttained: System
jenkins | 2022-08-09 06:38:29.255+0000 [id=30] INFO jenkins.InitReactorRunner$1#onAttained: System
jenkins | 2022-08-09 06:38:29.256+0000 [id=29] INFO jenkins.InitReactorRunner$1#onAttained: Loaded
jenkins | 2022-08-09 06:38:29.276+0000 [id=31] INFO jenkins.InitReactorRunner$1#onAttained: Config
jenkins | 2022-08-09 06:38:29.336+0000 [id=45] INFO hudson.model.AsyncPeriodicWork#lambda$doRun$1:
jenkins | 2022-08-09 06:38:29.358+0000 [id=45] INFO hudson.util.Retrier#start: Attempt #1 to do the
jenkins | 2022-08-09 06:38:30.042+0000 [id=30] INFO jenkins.install.SetupWizard#init:
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 | 71a827d0349844ddb06ec55d95278039 ❶ 这是管理员密码
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 | This may also be found at: /var/jenkins_home/secrets/initialAdminPassword
jenkins |
jenkins | *****
jenkins | *****
jenkins | *****
jenkins |
jenkins |
^CGracefully stopping... (press Ctrl+C again to force)
[+] Running 1/1
# Container jenkins Stopped
canceled
[root@localhost jenkins]#
```

执行后台启动，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 ps | grep jenkins
dc4bd8f994bd   jenkinsci/blueocean   "/sbin/tini -- /usr/..."   56 seconds ago
tcp, 0.0.0.0:8380->8080/tcp, :::8380->8080/tcp   jenkins
[root@localhost jenkins]#
```

5 配置镜像加速

编辑hudson.model.UpdateCenter.xml文件，修改为清华大学镜像

```
1 | https://mirrors.tuna.tsinghua.edu.cn/jenkins/updates/update-center.json
```

```
[root@localhost jenkins_home]# vi /home/jenkins/jenkins_home/hudson.model.UpdateCenter.xml
[root@localhost jenkins_home]# cat /home/jenkins/jenkins_home/hudson.model.UpdateCenter.xml
<?xml version='1.1' encoding='UTF-8'?>
<sites>
  <site>
    <id>default</id>
    <url>https://mirrors.tuna.tsinghua.edu.cn/jenkins/updates/update-center.json</url>
  </site>
</sites>
[root@localhost jenkins_home]#
```

然后重启服务。

6 初始化服务

6.1 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 71a827d0349844ddb06ec55d95278039，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:

71a827d0349844ddb06ec55d95278039 1 密码在这里

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

*****
*****
*****

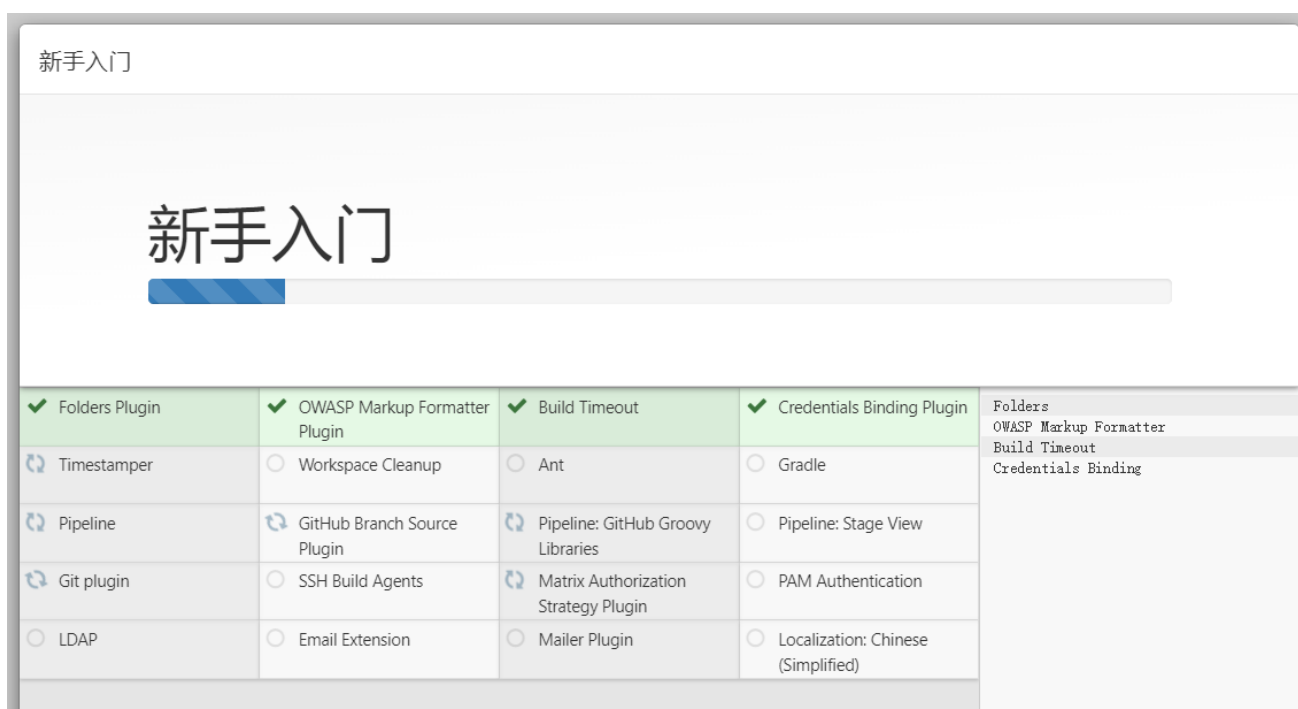
2022-08-09 06:42:20.819+0000 [id=29] INFO jenkins.InitReactorRunner$1#onAttained: Completed initialization
2022-08-09 06:42:20.960+0000 [id=22] INFO hudson.lifecycle.Lifecycle#onReady: Jenkins is fully up and running
[root@localhost jenkins]#
```

输入密码进入插件安装选择页面



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。



安装过程可能存在安装失败，

安装失败

部分插件安装失败，请重试或继续

[重试](#)

✓ Folders Plugin	✓ OWASP Markup Formatter Plugin	✓ Build Timeout	✓ Credentials Binding Plugin
✓ Timestampers	✓ Workspace Cleanup	✓ Ant	✓ Gradle
✓ Pipeline	✓ GitHub Branch Source Plugin	✓ Pipeline: GitHub Groovy Libraries	✓ Pipeline: Stage View
✓ Git plugin	✗ SSH Build Agents	✓ Matrix Authorization Strategy Plugin	✓ PAM Authentication
✓ LDAP	✓ Email Extension	✓ Mailer Plugin	✓ Localization: Chinese (Simplified)

注意：

如果有插件安装失败，可以尝试重试，排除是网络故障导致的；
如果重试多次还是失败，那么久不关注它。

Jenkins 2.346.3

[继续](#)[重试](#)

可以尝试重试

安装失败

部分插件安装失败，请重试或继续

[重试](#)

✗ SSH Build Agents	✗ Instance Identity
--------------------	---------------------

1.重试还是失败，那么不关注它，今日系统后可以通过升级系统解决问题

2.点击继续

Jenkins 2.346.3

[继续](#)[重试](#)

创建一个管理员账号

创建第一个管理员用户

用户名:

admin

1

密码:

.....

2

确认密码:

.....

3

全名:

admin

4

电子邮件地址:

.....

5

配置一个实例url配置，本地尝试的话默认即可。

实例配置

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 2.346.1

现在不要

保存并完成¹

点击完成后。

Jenkins即将就绪！

Jenkins安装已完成，但部分插件需要重启Jenkins。

重启¹

点击重启让所有插件生效。

如果后台没有自动重启，通过下面命令行手动重启。

```
1 | docker-compose start
```

然后刷新浏览器页面，重新进入登录页面



欢迎来到 Jenkins!

☐ 保持登录状态

输入账号密码，重新进入系统。

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

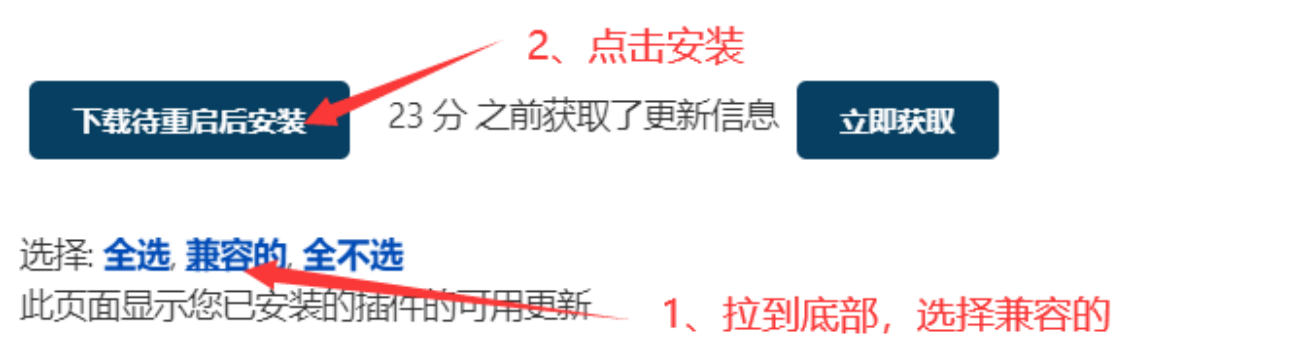
[Set up an agent](#)[Configure a cloud](#)[Learn more about distributed builds](#)

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。



进入插件管理



进入安装等待



安装完成重启

安装/更新 插件中

准备

- Checking internet connectivity
- Checking update center connectivity
- Success

JavaMail API	! 下载成功, 下次启动时生效
Pipeline: Nodes and Processes	! 下载成功, 下次启动时生效
Oracle Java SE Development Kit Installer	! 下载成功, 下次启动时生效
Command Agent Launcher	! 下载成功, 下次启动时生效
Bootstrap 5 API	! 下载成功, 下次启动时生效
Pipeline: Groovy	! 下载成功, 下次启动时生效
Pipeline: Stage Step	! 下载成功, 下次启动时生效

- [返回首页](#) 勾选上
(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)

Jenkins 即将关闭

安装/更新 插件中

准备

- Checking internet connectivity
- Checking update center connectivity
- Success

JavaMail API	! 下载成功, 下次启动时生效
Pipeline: Nodes and Processes	! 下载成功, 下次启动时生效
Oracle Java SE Development Kit Installer	! 下载成功, 下次启动时生效
Command Agent Launcher	! 下载成功, 下次启动时生效
Bootstrap 5 API	! 下载成功, 下次启动时生效
Pipeline: Groovy	! 下载成功, 下次启动时生效
Pipeline: Stage Step	! 下载成功, 下次启动时生效
重启 Jenkins	... 执行中

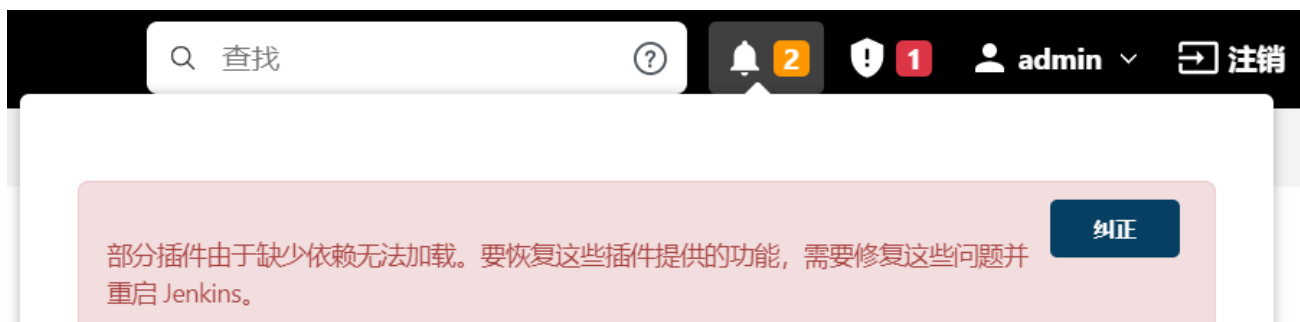
- [返回首页](#)
(返回首页使用已经安装好的插件)
- ☒ 安装完成后重启Jenkins(空闲时)

等一会系统会停止，然后还是使用命令行启动系统。

```
1 docker-compose start
```

刷新页面，进入主页，查看是否还有错误，如果依然有错误，那么升级一下Jenkins

点击信息



然后拉到底部，然后点击自动升级。

新版本的 Jenkins (2.366) 可以[下载](#) ([变更记录](#))。

或自动升级

安装/更新 插件中

准备

- Checking internet connectivity
- Checking update center connectivity
- Success

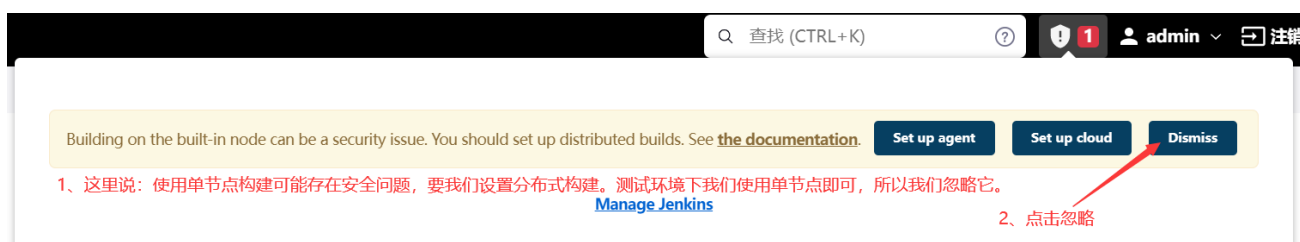
jenkins.war  等待

重启 Jenkins  等待

→ [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) [勾选上](#)

等待安装结束，后还是通过控制台重新启动一下服务，然后刷新页面→返回首页，查看会发现警告信息消失了，还剩一个安全提示信息。



到此消除了警告和安全问题提示了。

6.2 全局工具

通过下图所示的路径可以找到全局工具配置



拉到最下面，新增一个maven



然后

Maven

Maven 安装

系统下Maven 安装列表

新增 Maven

点击新增

Maven

Name

maven3.6.3

输入名称

自动安装

从 Apache 安装

版本

3.6.3

选择版本

新增安装

新增 Maven

保存

点击保存

6.3 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。

6.3.1 安装插件

Dashboard > 插件管理

管理 Jenkins

可更新

可选插件

已安装

高级

ssh

ssh搜索

Install

Name

SSH 2.6.1

构建包装

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

Publish Over SSH 1.24

选中它

构建工具

Send build artifacts over SSH

SSH Agent 295.v9ca_a_1c7cc3a_a_

This plugin allows you to provide SSH credentials to builds via a ssh-agent in Jenkins.

SSH Pipeline Steps 2.0.39.v831c5e6468b_c

Install without restart

Download now and install after restart

安装插件之前获取了更新信息

立即获取


安装结果


安装/更新 插件中

准备

- Checking internet connectivity
- Checking update center connectivity
- Success

Infrastructure plugin for Publish Over X
Publish Over SSH

 下载成功，下次启动时生效

 下载成功，下次启动时生效

1

等待下载完成

- [返回首页](#)
(返回首页使用已经安装好的插件)
- 2

☐ 勾选后重启服务器
安装完成后重启Jenkins(空闲时)

关闭中

Jenkins 即将关闭

安装/更新 插件中

准备

- Checking internet connectivity
- Checking update center connectivity
- Success

Infrastructure plugin for Publish Over X  下载成功，下次启动时生效
Publish Over SSH  下载成功，下次启动时生效
重启 Jenkins  执行中

- [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)

关闭后，服务器如果没有自动重启，执行通过命令行手动重启服务。

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

重启成功后刷新浏览器即可。

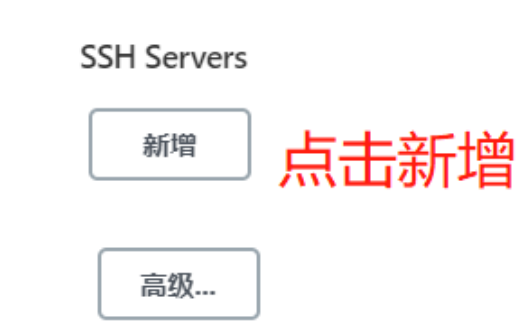
6.3.2 配置插件

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

SSH Servers

SSH Server

Name ?

my-linux

1.输入一个名字

Hostname ?

192.168.220.128

2.输入主机地址

Username ?

root

3.输入用户名

Remote Directory ?

/

4.输入远程目录

高级...

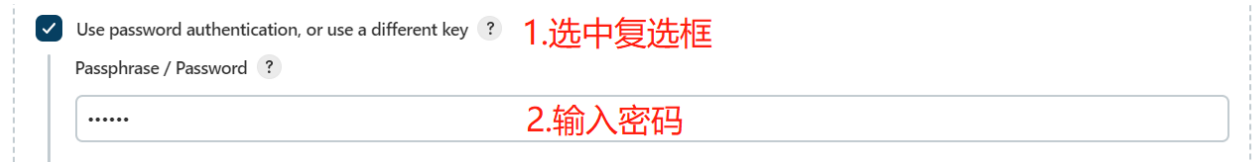
5.点击测试

Test Configuration

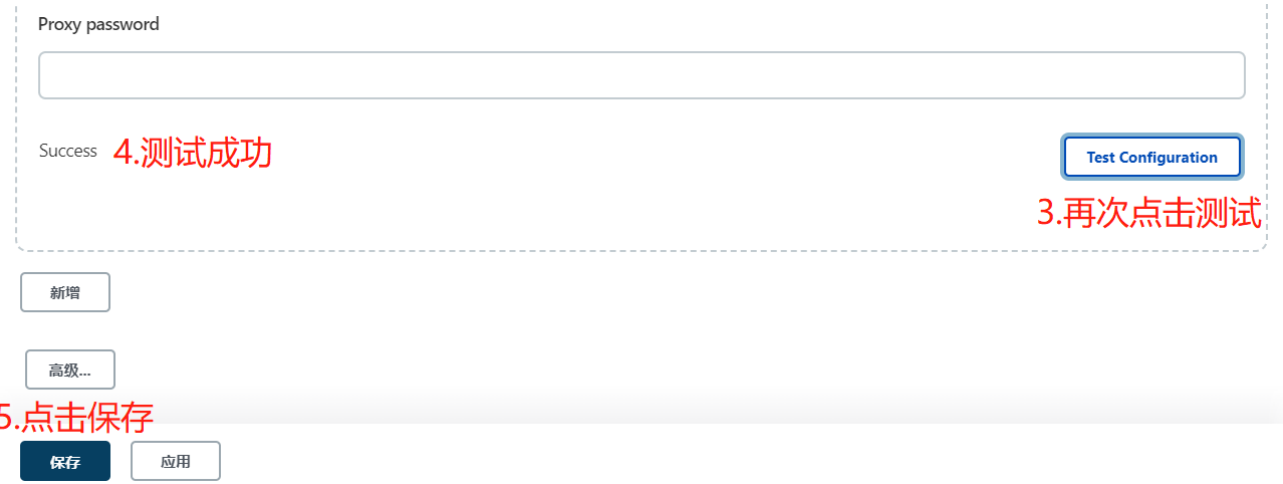
此时应该是连接失败



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。

General 源码管理 构建触发器 构建环境 **构建** 构建后操作

构建

Send files or execute commands over SSH ?

SSH Publishers

SSH Server Name ?

my-linux

这里可以选择主机 2

高级...

Transfers

Transfer Set Source files ?

3 这里可以选择脚本

Remove prefix ?

Remote directory ?

Exec command ?

docker start nacos-standalone 4 也可以不使用脚本，直接执行命令行

All of the transfer fields (except for Exec timeout) support substitution of [Jenkins environment variables](#)

高级...

Add Transfer Set

6.4 配置GitHub SSH

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。

6.4.1 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
bash-5.1# ssh-keygen -t rsa -b 4096 -C "2088688179@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:CvM5gQ8v1QxrqX/b1em4r/O6Vtl63cwFz0gXnxFEbsM 2088688179@qq.com
The key's randomart image is:
+---[RSA 4096]-----+
|           o+. |
|           oo |
|           E= |
|      . *      +.+ |
|    = * S      . O |
|    @ +      .+. = |
|   o B      ..o++ |
|   o ... .oo. * |
|   .....*O+. |
+-----[SHA256]-----+
bash-5.1#
```

此时会默认把密钥对文件生成到/root/.ssh/目录中。

6.4.2 配置GitHub

首先获取公钥，执行下面命令行

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

```
bash-5.1# cat /root/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQCAQC3bJSrAgoSkFmCLudwtNL04+VBkhkp/HEk8vfQJg8fpixtYXwWNzxLwKqtBgb3pN7t
AmHPurC5msGSilyN94OUFbEOeEeJnev3zFlHAIeyH2mB8w7/XDZnGUZP48Wt3FFdGaeMrhQeyw04uMtgynnoQwhR7u8yva9yaWCegbvE
N6zw+ettIRdcRaN0n3CF0zMeEFJWUwCBgTh0PJ9rUg6Bj/pSGiheMOM1A7HmrgaUBnRgEuLGebv33enExLKKg48X2avdpLSrLgB01Nom
7c/Yig3Vml+Lym25zYFxoK8zVhz9avKEOVfHjWC18bVoaTeQhdmumD+iLrZuRzAyVH60M0e44TgTO6UNI3f4FIWziJa25gYQdQuYX2as
```

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

SSH keys / Add new

Title

Jenkins

1 填写名称，随意名称即可

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 将刚才复制公钥粘贴到这里

Add SSH key

3 然后点击添加

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



Jenkins

SHA256:CvM5gQ8v1QxrqX/b1em4r/06Vt163cwFz0gXnxFEbsM

SSH

Added on Mar 4, 2023

Never used — Read/write

Delete

Check out our guide to [generating 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

```
bash-5.1# ssh -T git@github.com
The authenticity of host 'github.com (20.205.243.166)' can't be established.
ED25519 key fingerprint is SHA256:+DiY3wvvV6TuJhbpZisF/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.
bash-5.1#
```

输入yes

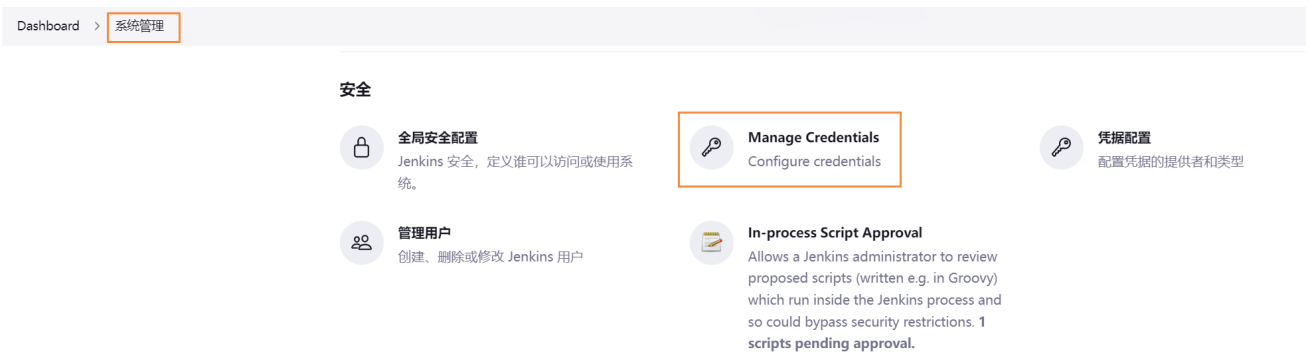
6.4.3 管理凭证

首先查看私钥，执行下面命令

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

```
bash-5.1# cat /root/.ssh/id_rsa
-----BEGIN OPENSSH PRIVATE KEY-----
b3BlbnNzaC1rZXktbjEAAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAACFwAAAAAdzc2gtcn
NhAAAAAwEAAQAAAgEAt2yUqwKqEpBZgi7ncLTS9OP1QZIZKfxxJPL30CYPH6YsbWF8Fjc8
S8CqrQYG96Te7c1YI9GSrnFRzxucxxYg4PSCIcb0mOkRPp0gTLacCBuXlbc+AS5GkRwLVF
V+IZ+ZB+hLgrijgbSfEaYyyR2G2mwJhz7qwuZrBkiJcjfeDlBWxDnhHiZ3ld8xZRwIhMh9
pgfMO/1w2ZxlGT+PFrdxRXRmnjK4UHssNOLjLasp56EMIue7vMr2vcmlgnoG7xHinAZsWx
243UPeDBan0YugW89WQQ99nasZpOcKcI4pUzsKSo9dkpLD0XdvBLuA/WgL7QzFki+gpuQ0
xgp4agbTes8PnrbSEXxEWjdJ9whdMzHhBSVlFnAak4dDyfa1KugY/6UhooXjDjJQOx5q4G
lAZ0YBLixnm7993pxMSyiquPF9mr3aS0qy4ATpTaJjoeT9Nh9KNiAVIJbHrbeNT/E22Y03
LKMEgTiTKdxVwvZobDrYLn+Q9d7wP6Qcb08Lq3k0lU95s2gYVsPzri02O3P2IoN1Zpfi8p
tuc2BcTpPM1Yc/WryhDlXx4lgtfG1aGk3kIXZrpg/oi62bkcwMlR+tDNHuOE4EzulDSN3+
BSFs4o2tuYGEHULmF9mrNUPq2LOU9o+4u4AZwYpDXoHdvHeQJtncQsSTWhlvu3VrQap+21
sAAAdIi0OeJItDniQAAAAHc3NoLXJzYQAAAgEAt2yUqwKqEpBZgi7ncLTS9OP1QZIZKfxx
JPL30CYPH6YsbWF8Fjc8S8CqrQYG96Te7c1YI9GSrnFRzxucxxYg4PSCIcb0mOkRPp0gTL
acCBuXlbc+AS5GkRwLVFV+IZ+ZB+hLgrijgbSfEaYyyR2G2mwJhz7qwuZrBkiJcjfeDlBW
xDnhHiZ3ld8xZRwIhMh9pgfMO/1w2ZxlGT+PFrdxRXRmnjK4UHssNOLjLasp56EMIue7vM
r2vcmlgnoG7xHinAZsWx243UPeDBan0YugW89WQQ99nasZpOcKcI4pUzsKSo9dkpLD0Xdv
BLuA/WgL7QzFki+gpuQ0xgp4agbTes8PnrbSEXxEWjdJ9whdMzHhBSVlFnAak4dDyfa1Kug
```

在Jenkins中找到凭证管理



添加一个全局凭证



全局凭据 (unrestricted)

+ Add Credentials

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)

ID ?

描述 ?

2 输入描述, 如: Github Jenkins Private Key

Username

3 输入用户名, 如: anyuser

☐ Treat username as secret ?

Private Key

☒ Enter directly

Key

```
-----BEGIN OPENSSH PRIVATE KEY-----
b3B1bnNzaC1rZXktbjEAAAAAAAAAG5vbmUAAAAAAAAAEbm9uZQAAAAAAAAABAAACFwAAAAadzcn
NhAAAAAwEAAQAAAEAt2yUqwKqEpBZgi7ncLTS90P1QZIZKfxxJPL30CYPH6YsbWF8F.jc8
```

4 输入私钥

Passphrase

Create 5

设置好了，后期就可以在任务中选择你的凭据了

最后一步需要修改Git Host Key 验证策略，如下图所示。

Dashboard > 系统管理 > 全局安全配置

☐ 为每个新创建的用户生成一个遗留的 API token (不建议) ?

☐ 允许用户手动创建一个遗留的 API token (不建议) ?

☒ 启用 API Token 使用统计 ?

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

保存 应用

7 扩展

7.1 添加Gitee凭证

如果你的代码托管到gitee上面的时候，你可以提前添加好gitee的账号凭证，方便你后续使用。

7.1.1 找到凭证管理

在Jenkins中找到凭证管理

Dashboard > 系统管理

安全

全局安全配置
Jenkins 安全，定义谁可以访问或使用系统。

Manage Credentials
Configure credentials

凭据配置
配置凭据的提供者 and 类型

管理用户
创建、删除或修改 Jenkins 用户

In-process Script Approval
Allows a Jenkins administrator to review proposed scripts (written e.g. in Groovy) which run inside the Jenkins process and so could bypass security restrictions. 1 scripts pending approval.

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 点击它

全局凭据 (unrestricted)

+ Add Credentials

7.1.2 填写凭据内容

New credentials

类型

Username with password 1 选择用户密码类型

范围 ?

全局 (Jenkins, nodes, items, all child items, etc) 2 选择全局范围

用户名 ?

3 输入gitee的用户名

☒ Treat username as secret ? 4 勾选上

密码 ?

5 输入gitee的密码

ID ?

描述 ?

阿伟gitee凭据 6 输入描述

Create

7 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
 2dfa89a4-50b4-4016-9038-fce17fb75817	阿伟gitee凭据	Username with password	阿伟gitee凭据 

图标 小 中 大

后面要用gitee的时候选择它就行了