

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

```
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

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

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

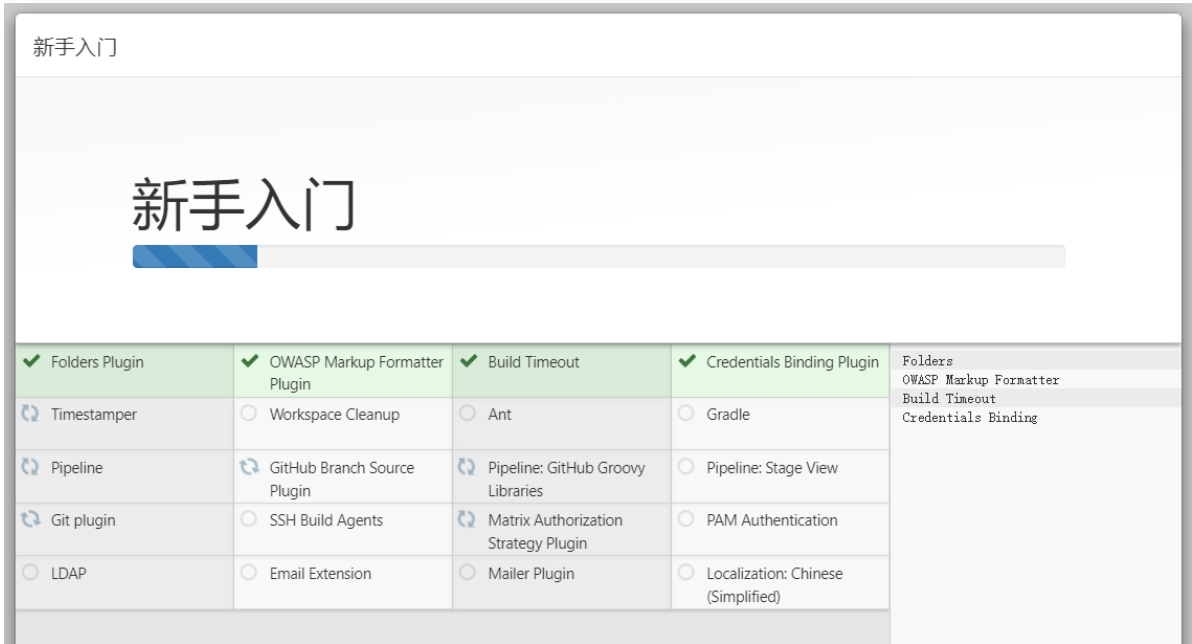
2022-08-09 06:42:20.819+0000 [id=29] INFO jenkins.InitReactorRunner$1#onAttained: Completed initialization
2022-08-09 06:42:20.960+0000 [id=22] INFO hudson.lifecycle.Lifecycle#onReady: Jenkins is fully up and running
[root@localhost jenkins]# █
```

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，

新手入门

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

新手入门

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

新手入门

创建第一个管理员用户

用户名:

admin1

密码:

.....2

确认密码:

.....3

全名:

admin4

电子邮件地址:

.....5

Jenkins 2.346.1

使用admin账户继续

保存并完成6

配置一个实例url配置，本地尝试的话默认即可。

新手入门

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

登录

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

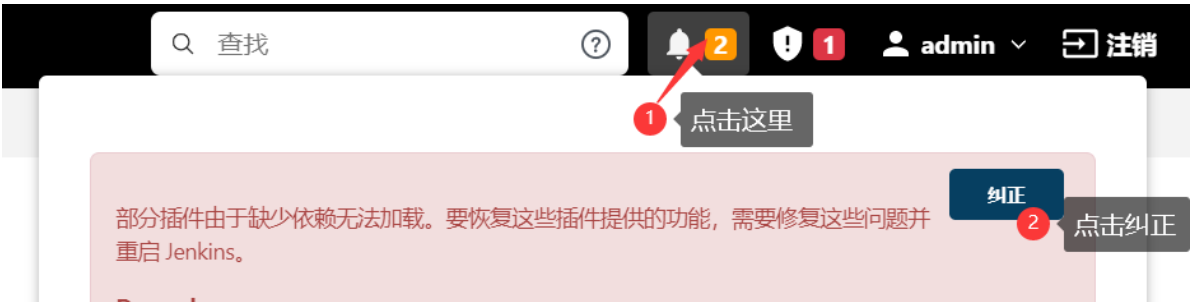
Set up an agent →

Configure a cloud →

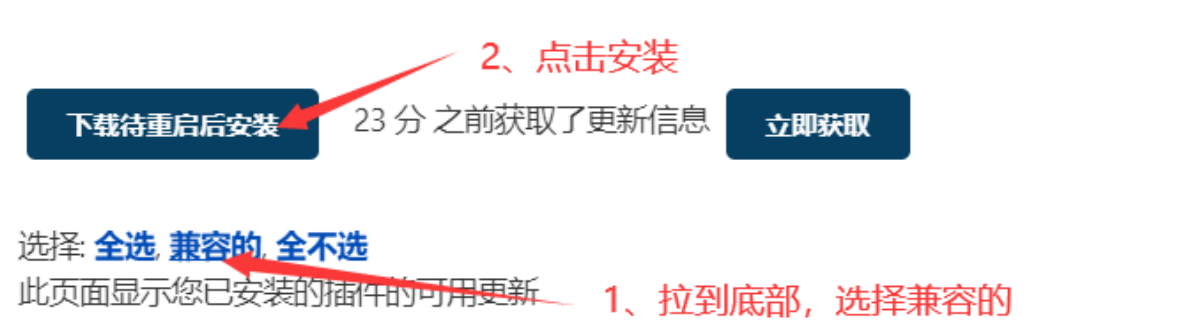
Learn more about distributed builds ↗

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/更新 插件中

进入安装等待...

准备

- Checking internet connectivity
- Checking update center connectivity

JavaMail API	⋯ 等待
Pipeline: Nodes and Processes	⋯ 等待
Oracle Java SE Development Kit Installer	⋯ 等待
Command Agent Launcher	⋯ 等待
Bootstrap 5 API	⋯ 等待
Pipeline: Groovy	⋯ 等待
Pipeline: Stage Step	⋯ 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)

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)

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

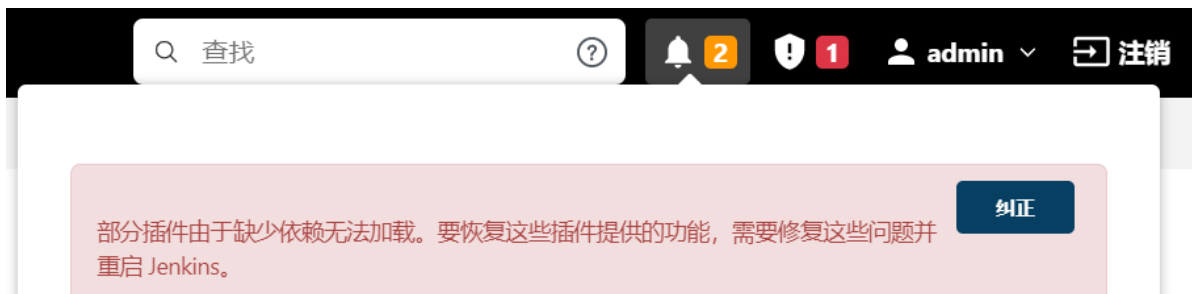
- [返回首页](#)
(返回首页使用已经安装好的插件)
- ☒ 安装完成后重启Jenkins(空闲时)

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

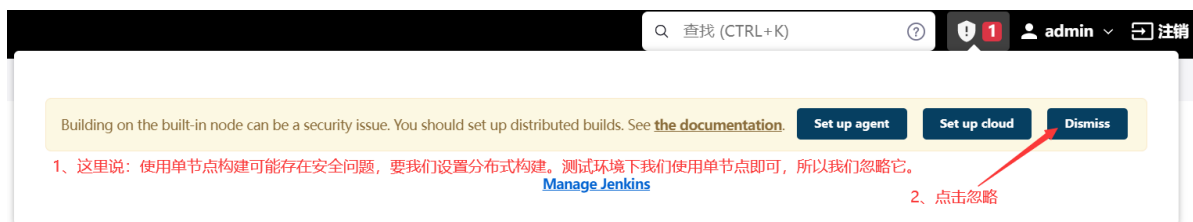
重启 Jenkins ... 等待

→ [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**

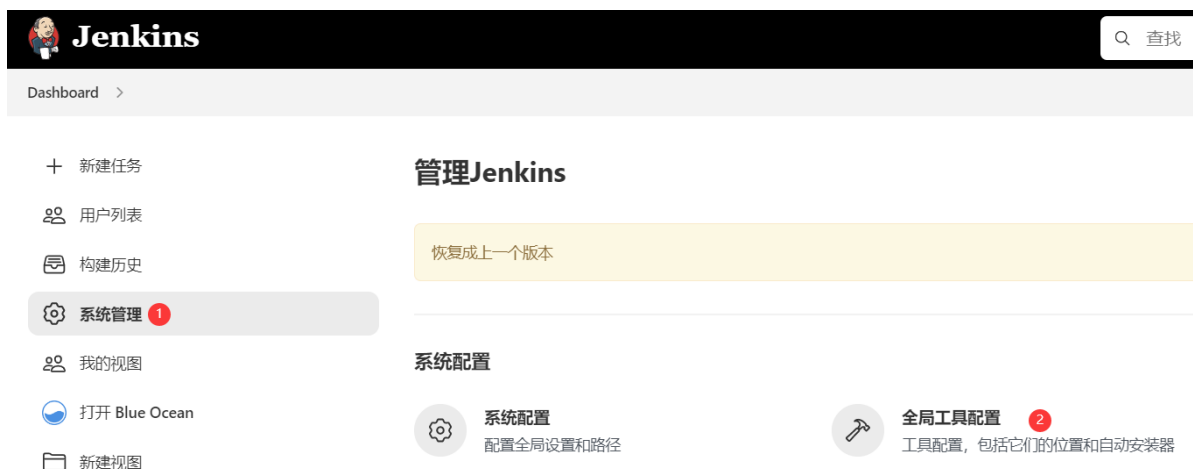
等待安装结束，后还是通过控制台重新启动一下服务，然后刷新页面->返回首页，查看会发现警告信息消失了，还剩一个安全提示信息。



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

Maven

Maven 安装... ¹

保存

应用

然后

Maven

Maven 安装

系统下Maven 安装列表

新增 Maven ¹ 点击新增

Maven

Name

maven3.6.3 ² 输入名称

☒ ³ 自动安装 ?

从 Apache 安装

版本

3.6.3 ⁴ 选择版本

新增安装 ▾

新增 Maven

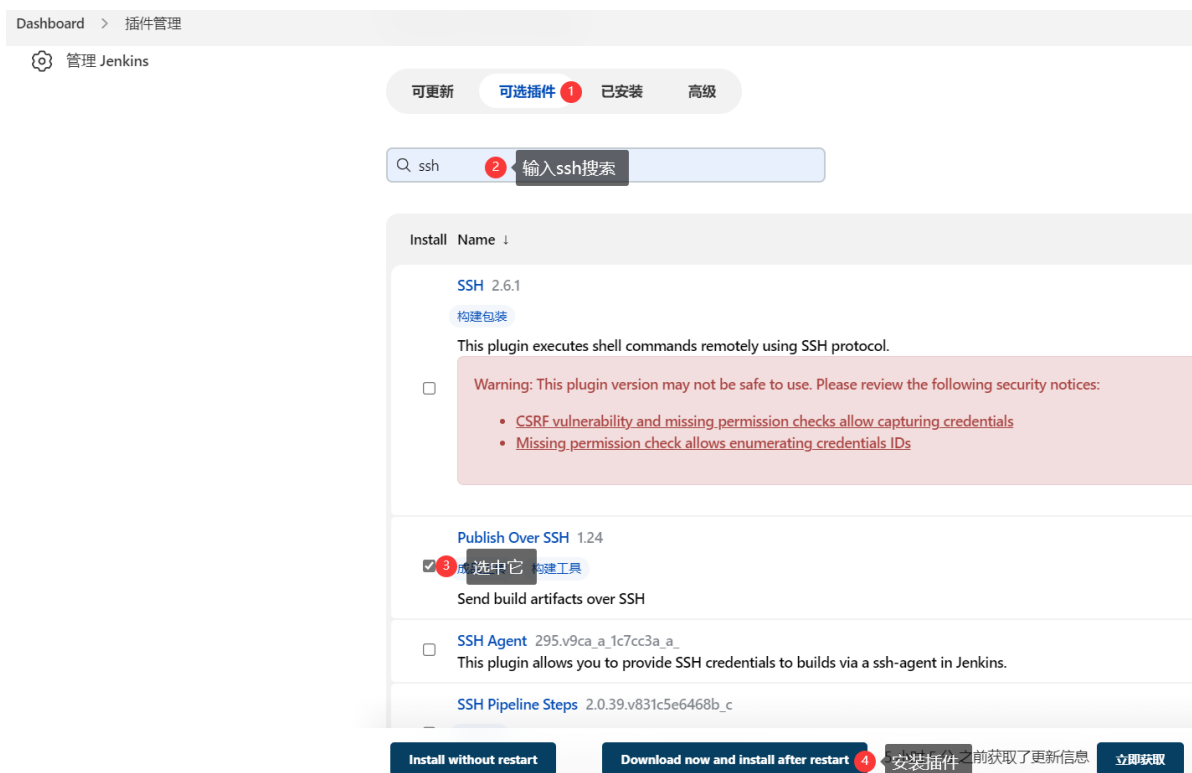
保存 ⁵

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



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

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。

首先找到系统配置。



找到SSH Server

SSH Servers

新增

点击新增

高级...

输入信息

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

高级...

Failed to connect or change directory

jenkins.plugins.publish_over.BapPublisherException: Failed to connect and initialize SSH connection. Message: [Failed to connect session for config [my-linux]. Message [Auth fail]]

Test Configuration

点击高级输入密码

☒ Use password authentication, or use a different key ?

1.选中复选框

Passphrase / Password ?

.....

2.输入密码

测试与保存

Proxy password

Success

4.测试成功

3.再次点击测试

Test Configuration

新增

高级...

5.点击保存

保存

应用

这样你就可以在构建过程中使用了，示例如下图。

构建

Send files or execute commands over SSH ?

SSH Publishers

SSH Server

Name ?

my-linux

这里可以选择主机

高级...

Transfers

Transfer Set

Source files ?

这里可以选择脚本

Remove prefix ?

Remote directory ?

Exec command ?

docker start nacos-standalone 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