

Jenkins 使用说明

- A. 基础配置
- B. 下载 jenkins 官方镜像
- C. 基于官方镜像自定义本项目镜像
- D. 初始化设置并配置 jenkins

A. 基础配置

操作系统: macOS Mojave version 10.14.5

Docker Desktop: version 2.0.0.0-mac78(28905)

Terminal: version 2.9.5 (421.2)

注: 此教程默认读者已经在自己系统上安装并成功运行 Docker, 如没有安装 Docker, 请先安装, 参考链接: <http://oak.cs.ucla.edu/classes/cs144/docker/index.html>

B. 下载 jenkins 官方镜像

参考链接 https://hub.docker.com/_/jenkins

从 docker hub 上下载某个或一组镜像时用 **docker pull**。如果没有加标签, 则默认使用标签:latest。如下命令拉出 jenkins:latest 镜像:

```
$ docker pull jenkins
```

到截稿日期为止, Docker Hub 上官方 jenkins 镜像最近的版本为 2.60.3

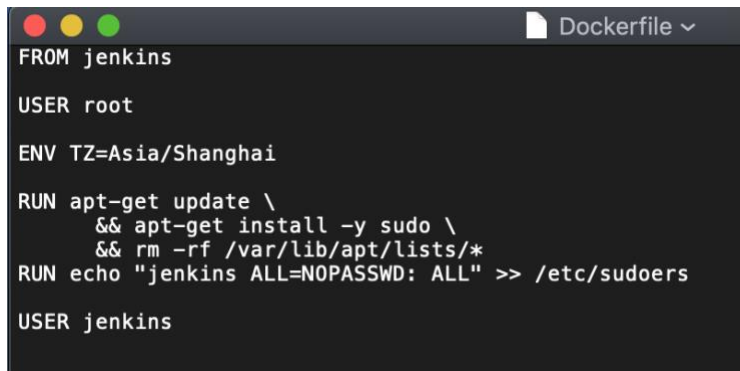
C. 基于官方镜像自定义本项目镜像

下载完官方 jenkins 镜像后, 由于我们后面会在 docker 容器里面运行 docker 命令, 所以需要给用户 jenkins 在容器中调用 docker 命令的权限 (如 docker ps) 也就是可以从 jenkins 容器里面执行主机上的 docker 命令。

又因为非 root 用户需要 sudo 权限来执行 docker 语句, 所以我们需要给用户 jenkins sudo 用户权限, 而官方镜像里面用户 jenkins 没有 sudo 权限, 所以我们需要基于官方镜像, 新建个自定义镜像来给用户 sudo 权限:

1. 写 Dockerfile

新建个文件夹如 tmp, 然后创建个名字为 Dockerfile 的文件, 内容如下:



```
FROM jenkins
USER root
ENV TZ=Asia/Shanghai
RUN apt-get update \
    && apt-get install -y sudo \
    && rm -rf /var/lib/apt/lists/*
RUN echo "jenkins ALL=NOPASSWD: ALL" >> /etc/sudoers
USER jenkins
```

Source Code:

FROM jenkins

USER root

ENV TZ=Asia/Shanghai

RUN apt-get update \
 && apt-get install -y sudo \
 && rm -rf /var/lib/apt/lists/*

RUN echo "jenkins ALL=NOPASSWD: ALL" >> /etc/sudoers

USER jenkins

2. 创建新的镜像

然后进到 Dockerfile 所在的目录下

\$ docker build -t daxuan/jenkins .

这个命令中 daxuan/jenkins 为我们新见镜像的名字 后面的“.”为 Dockerfile 所在目录。然后会看到

Sending build context to Docker daemon 9.216kB

Step 1/6 : FROM jenkins

---> eceefd6b4989

...

...

...

Successfully tagged daxuan/jenkins:latest

这个过程很漫长, 大概 12 分钟左右, 因为 apt-get 在默认源 <http://cdn-fastly.deb.debian.org/> 执行的, 如果你想快一些的话可以去换源, 网上很多换源教程, 这里不详述。

3. 创建容器

这时候我们已经创建好了基于 jenkins 官方镜像的自定义镜像。

LuckyMicky-2:tmp shudaxuan\$ docker images						
REPOSITORY	6	15	6	TAG	IMAGE ID	CREATED
daxuan/jenkins				latest	65d1ced003ee	About a minute ago
						SIZE
						700MB

然后我们运行基于这个镜像创建容器：

```
$ docker run -d --name MyJenkins -v /var/run/docker.sock:/var/run/docker.sock -v $(which docker):/usr/bin/docker -v /Users/shudaxuan/Desktop/Jenkins_home:/var/jenkins_home -p 8080:8080 -p 50000:50000 daxuan/jenkins
```

```
1 docker run -d \  
2   --name MyJenkins \  
3   -v /var/run/docker.sock:/var/run/docker.sock \  
4   -v $(which docker):/usr/bin/docker \  
5   -v /Users/shudaxuan/Desktop/Jenkins_home:/var/jenkins_home \  
6   -p 8080:8080 \  
7   -p 50000:50000 \  
8   daxuan/jenkins
```

-d 在后台运行容器并打印出容器 ID

--name MyJenkins 容器命名为 MyJenkins

-v /var/run/docker.sock:/var/run/docker.sock

容器运行 docker 命令会映射到宿主机中，比如我们在容器中运行 `sudo docker ps` 就等于在宿主机上运行 `docker ps`。

-v \$(which docker):/usr/bin/docker

使宿主机中的 docker 映射到容器里面，这样我们不用在容器里面再安装 docker。

-v /Users/shudaxuan/Desktop/Jenkins_home:/var/jenkins_home

在宿主机上的桌面上创建 Jenkins_home 文件夹，对应容器中的 `/var/jenkins_home` 文件夹，这样我们既可以在容器销毁后还在宿主机上存有容器里面的数据，还可以快速的从宿主机中导入文件到容器里面

-p 8080:8080

Port forwarding. Forward any network request to port 8080 on the host machine to the port 8080 on the container.

-p 50000:50000

For connecting a slave agent

daxuan/jenkins

基于镜像 daxuan/jenkins 创建容器

4. 检验容器

之后我们启动新建好的容器 MyJenkins

```
$ docker start MyJenkins
```

然后我们先在主机 Terminal 中以用户 jenkins 的身份登入容器中，看看是否可以运行 docker 命令

```
$ docker exec -it -u jenkins MyJenkins bash
```

登入后

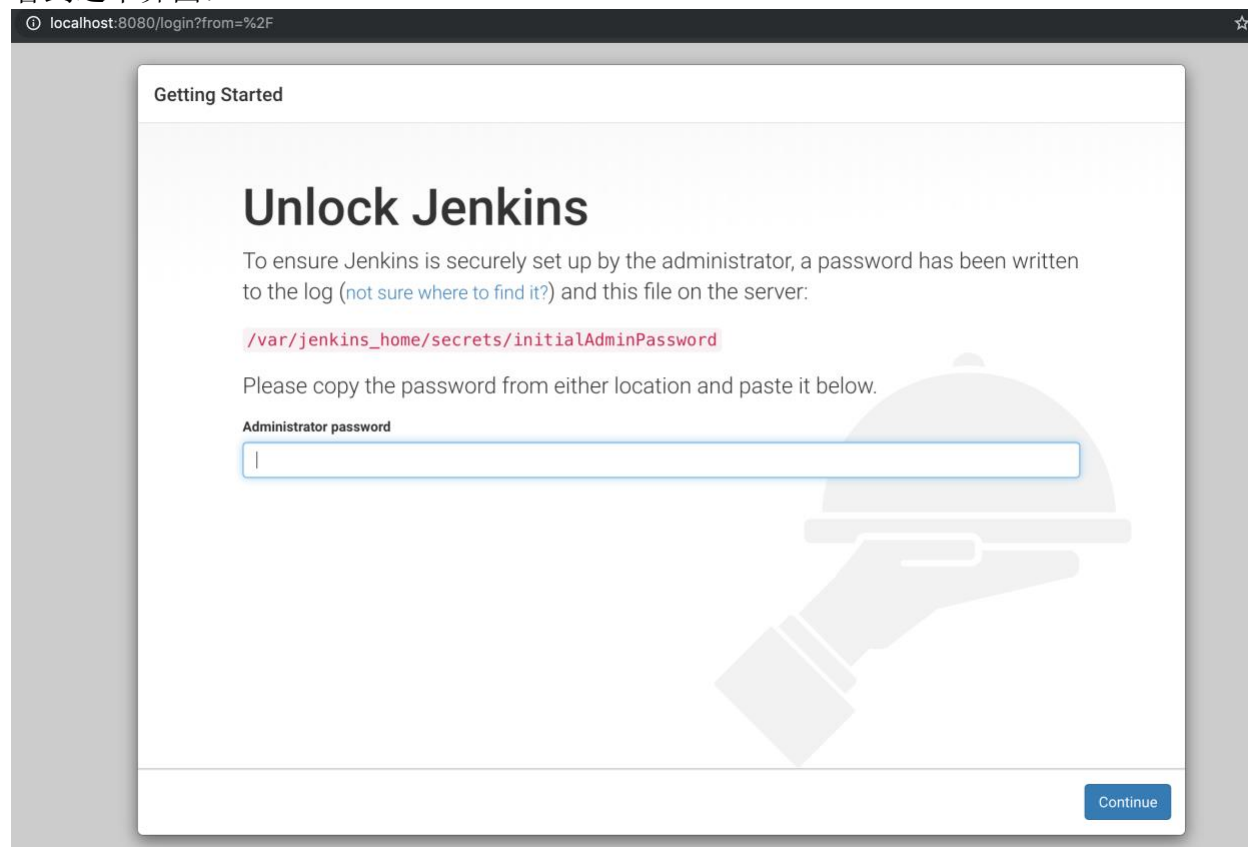
```
jenkins@119e4e60af64:/$ sudo docker ps
```

检验下是否正常返回结果

D. 初始化设置 jenkins

1. 初始密码

如果上一步最后验证没问题，那么我们现在打开浏览器 `localhost:8080`，第一次登入会看到这个界面：



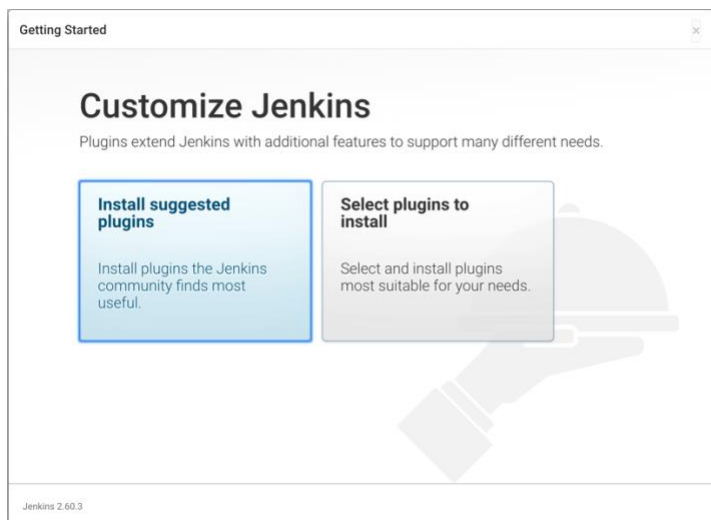
这个初始管理员密码也可以在 `Jenkins_home/secrets/initialAdminPassword` 中查看，这个密码是我们创建容器时随机生成的，密码类似长这样：`182c5936cb534c8f8e67aeba255038a2`

（宿主机 `Jenkins_home` 地址详见上一步

`-v /Users/shudaxuan/Desktop/Jenkins_home:/var/jenkins_home` 中的设置）

2. 自定义 Jenkins

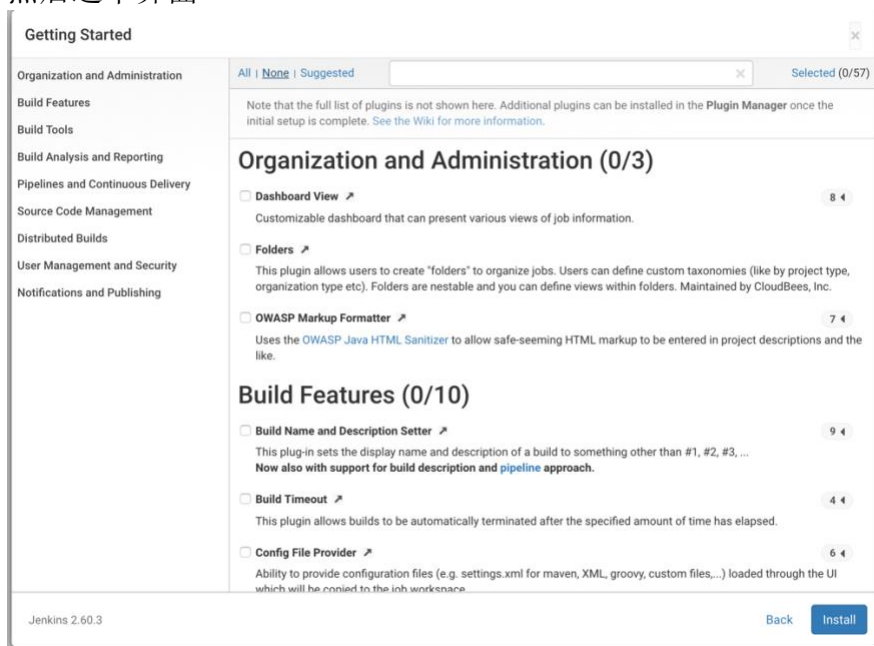
复制粘贴后 `continue`, 会看到这个界面



为了节省时间，选 select plugins to install.

3. 更改默认插件选项

然后这个界面



点上面的 none，也就是先什么插件也别安。

4. 创建管理员

Getting Started

Create First Admin User

Username:

Password:

Confirm password:

Full name:

Jenkins 2.60.3

[Continue as admin](#) [Save and Finish](#)

5. 管理 Jenkins

之后进入 jenkins, 左侧点 Manage Jenkins

Jenkins

Jenkins

- New Item
- People
- Build History
- Manage Jenkins**
- My Views

Welcome to Jenkins!

Please **create new jobs** to get started.

Build Queue

No builds in the queue.

Build Executor Status

1	Idle
2	Idle

6. 更新 Jenkins


然后复制最新版的 jenkins 更新地址


Manage Jenkins

⚠ New version of Jenkins (2.176.1) is available for [download](#) ([changelog](#)).
Warnings have been published for the following currently installed components:

- Jenkins 2.60.3 core and libraries:
 - Multiple security vulnerabilities in Jenkins 2.120 and earlier, and LTS 2.107.2 and earlier
 - Multiple security vulnerabilities in Jenkins 2.153 and earlier, and LTS 2.138.3 and earlier
 - Multiple security vulnerabilities in Jenkins 2.132 and earlier, and LTS 2.121.1 and earlier
 - Multiple security vulnerabilities in Jenkins 2.159 and earlier, and LTS 2.150.1 and earlier
 - Multiple security vulnerabilities in Jenkins 2.171 and earlier, and LTS 2.164.1 and earlier
 - Multiple security vulnerabilities in Jenkins 2.88 and earlier, and LTS 2.73.2 and earlier
 - Multiple security vulnerabilities in Jenkins 2.106 and earlier, and LTS 2.89.3 and earlier
 - Multiple security vulnerabilities in Jenkins 2.83 and earlier, and LTS 2.73.1 and earlier
 - Multiple security vulnerabilities in Jenkins 2.94 and earlier, and LTS 2.89.1 and earlier
 - Multiple security vulnerabilities in Jenkins 2.145 and earlier, and LTS 2.138.1 and earlier
 - Multiple security vulnerabilities in Jenkins 2.115 and earlier, and LTS 2.107.1 and earlier
 - Multiple security vulnerabilities in Jenkins 2.137 and earlier, and LTS 2.121.2 and earlier

Configure which of these warnings are shown

 [Configure System](#)
Configure global settings and paths.

 [Configure Global Security](#)
Manage Jenkins' plugins which is allowed to access the system

然后在 Terminal 里面下载到容器里面

```
$ docker exec -it -u0 MyJenkins bash
```

```
root@3425f69b3f7d:/#  
wget http://updates.jenkins-ci.org/download/war/2.176.1/jenkins.war
```

移动到正确的位置

```
root@3425f69b3f7d:/# mv ./jenkins.war /usr/share/jenkins
```

授权

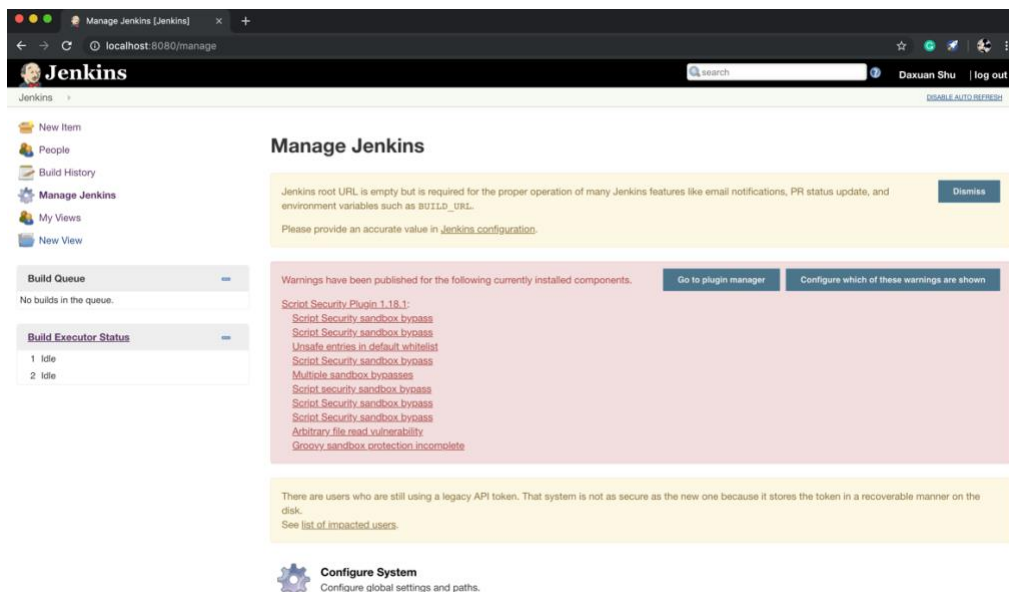
```
root@3425f69b3f7d:/# chown jenkins:jenkins /usr/share/jenkins/jenkins.war
```

然后退出 docker

```
root@3425f69b3f7d:/# exit
```

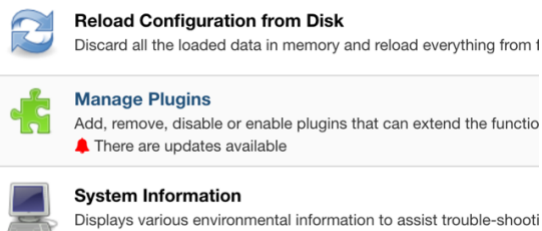
```
$ docker container restart MyJenkins (from your host terminal)
```

这是再登入 localhost:8080 看看是否更新成功，登入后进入 Manage Jenkins 界面，目前应该是这样的：（版本信息在右下角，现在版本应该不是原先的旧版本 2.60.3 了）

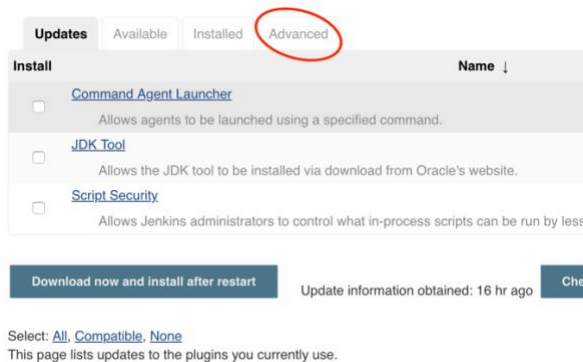


7. 更换插件加速源

好的，现在 jenkins 已经更新成功，我们开始安装必要的插件 Subversion，为了快速安装插件，先换源。首先，在 Manage Jenkins 界面下点击 Manage Plugins



然后选 Advanced



之后在 Update Site 下面换新的源，插件加速地址参考：

<https://www.cnblogs.com/zhuochong/p/10082498.html>

亲测日本第一个源速度还可以，地址为 <http://mirror.esuni.jp/jenkins/>

Update Site

URL

Submit

粘贴后点 Submit 提交设置。

8. 安装插件

选择 Subversion 插件后安装

Filter:

Updates **Available** Installed Advanced

Install ↓	Name	Version
<input type="checkbox"/>	Jigomerge This plug-in integrates Jigomerge to handle merge between SVN(Subversion) branches.	0.9
<input type="checkbox"/>	Subversion Workspace Cleaner Automatically cleans up Subversion modules removed from job configuration when enabled.	1.1
<input type="checkbox"/>	Cron Column Column showing the cron trigger expressions that can be configured on a job (Subversion, Scheduled Builds etc).	1.4
<input type="checkbox"/>	Subversion Partial Release Manager This plugin provides a way to keep track of builds by revision for subversion managed projects.	1.0.1
<input type="checkbox"/>	Subversion Revert This plugin automatically performs Subversion reverting when the build status changes to unstable or failed.	1.3
<input type="checkbox"/>	SVN 1.4 Compatibility This plugin forces the built-in SVNKit library to use the Subversion 1.4 working copy format.	1.1
<input type="checkbox"/>	ViewVC This plugin integrates ViewVC browser interface for CVS and Subversion to Hudson. Entries in change logs will be hyperlinked to the specified files/diffs in ViewVC.	1.7
<input type="checkbox"/>	Blame Subversion This plug-in provides utilities for getting svn info and build number synchronize from upstream job to downstream job	1.200
<input checked="" type="checkbox"/>	Subversion	2.12.1

Install without restart Download now and install after restart Update information obtained: N/A ago Check now

待安装完成（部分插件可能需要重启 jenkins 后生效，有些附属插件可能第一次下载不成功，不过可以多尝试下，有时可能网不好）

安装成功后，Manage Jenkins 界面应该没有红色的 Error 了：

Jenkins

Manage Jenkins

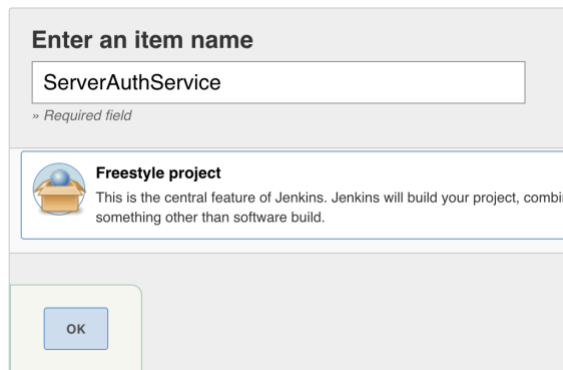
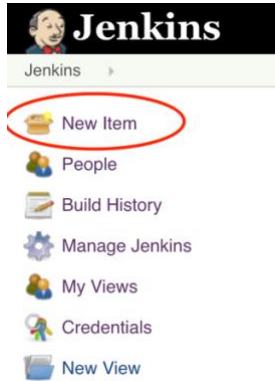
Jenkins root URL is empty but is required for the proper operation of many Jenkins features like email notifications, PR status update, and environment variables such as `BUILD_URL`.
Please provide an accurate value in [Jenkins configuration](#).

There are users who are still using a legacy API token. That system is not as secure as the new one because it stores the token in a recoverable manner on the disk.
See [list of unpatched users](#).

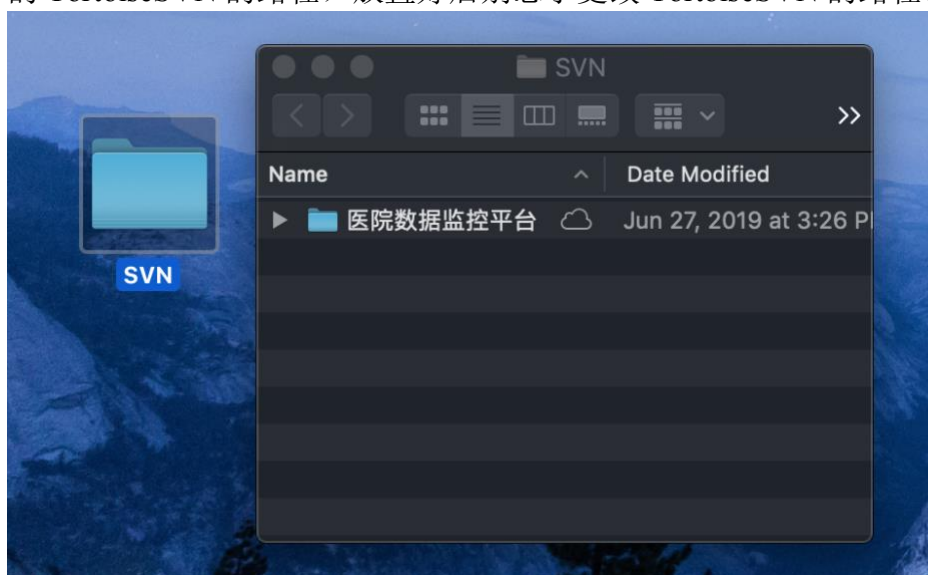
- [Configure System](#)
Configure global settings and paths.
- [Configure Global Security](#)
Secure Jenkins; define who is allowed to access/use the system.
- [Configure Credentials](#)
Configure the credential providers and types.
- [Global Tool Configuration](#)
Configure tools, their locations and automatic installers.
- [Reload Configuration from Disk](#)

9. 创建新任务

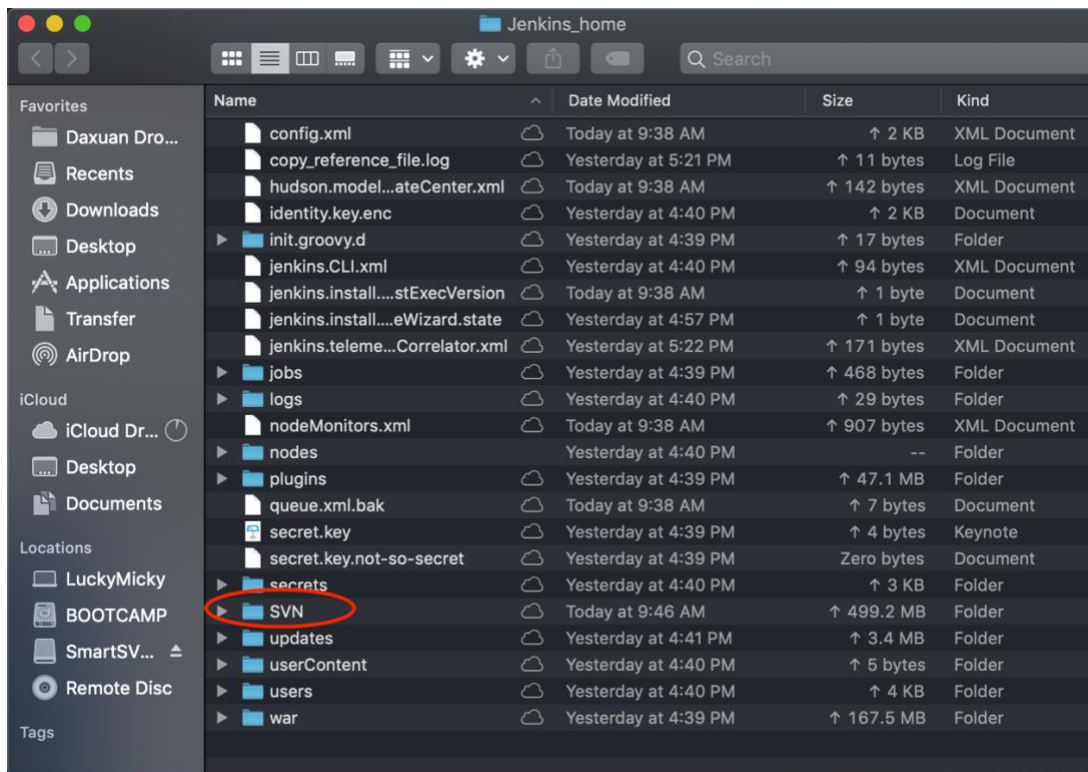
之后我们点击左侧 New Item 创建新的项目

A screenshot of the 'Enter an item name' form in Jenkins. The form has a text input field containing 'ServerAuthService'. Below the field is a small note: '» Required field'. Underneath the input field is a section titled 'Freestyle project' with a brief description: 'This is the central feature of Jenkins. Jenkins will build your project, combir something other than software build.' At the bottom left of the form is an 'OK' button.

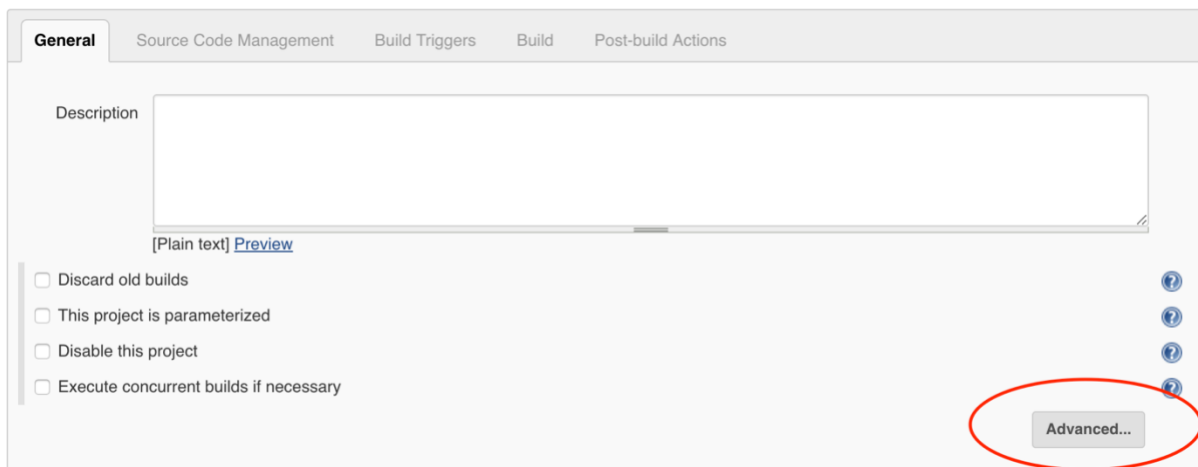
之后把 SVN 的目录通过之前创建容器时设置的 jenkins_home 文件夹放到容器里面
（注：SVN 是我的文件夹的名称，SVN 文件夹下面就是医院数据监控平台，就是你们用的 TortoiseSVN 的路径）放置好后别忘了更改 TortoiseSVN 的路径。



放置后 jenkins_home 文件夹：



点击 Advanced



这里面 Use custom workspace 的 Directory 路径为 config.ini 文件在容器内存在的位置
Source Code Management 下的 Repository URL 为 neusoft SVN 下与 custom workspace 想对应的路径. 然后登入用户名和密码在 Credentials 后面的 Add 中添加后选中, 没有红色的字说明连接成功。

☐ Block build when downstream project is building
☒ Use custom workspace
 Directory:
 Display Name:
☐ Keep the build logs of dependencies

Source Code Management

☐ None
☒ Subversion

Modules

Repository URL	<input type="text" value="http://msd.neusoft.com:8123/his_svn/JSFZB_NEW/工具/医院数据监控平台/"/>
Credentials	<input type="text" value="gengxiaolei/*****"/> Add
Local module directory	<input type="text" value="."/>
Repository depth	<input type="text" value="infinity"/>
Ignore externals	<input checked="" type="checkbox"/>
Cancel process on externals fail	<input checked="" type="checkbox"/>

Add module...

之后在下面 Build 板块下加两个 shell script 板块（如果在 Windows 环境则选择 batch）

Build

Add build step ▼

- Execute Windows batch command
- Execute shell**
- Invoke top-level Maven targets

Add post-build action ▼

再填一个 shell（一个 build.sh, 第二个 push.sh）

（注：在 push.sh shell command 中 docker push 必须带 -p 把密码传进去，因为 jenkins build 时不会有交互界面让我们敲密码，所以我们需要稍微改动一下 config.ini 文件来加个密码信息进去，虽然因为在脚本中写密码，然后运行还会有 log 有安全隐患，不过内部用的话也没什么，如果想在外部用的话，建议斟酌一下，看看有没有更保密的方法）



代码:

*build.sh:

#!/bin/bash

set +v

#read config

source ./config.sh

config registry_path regval

config tag tg

#build

sudo docker build -t \$regval/rmpserverauthservice:\$tg -

f ./Rmp.Server/Rmp.Server.Services/Dockerfile-AuthService ./Rmp.Server/Rmp.Server.Services/

*Push.sh:

#!/bin/bash

set +v

#read config

source ./config.sh

config username uname

config registry reg

config registry_path regpath

config tag tg

config passwd password

#login

sudo docker login -u=\$uname -p=\$password \$reg

```
#push
sudo docker push $regpath/rmpserverauthservice:$tg
```

Config.ini 文件:

```
config.ini
1 [docker]
2 username=daxuan@1836535521596902
3 passwd=*****
4 registry_path=registry.cn-beijing.aliyuncs.com/neurmp
5 registry=registry.cn-beijing.aliyuncs.com
6 tag=latest
7
```

Save & Apply 后就设置完成了，可以 build 了，点 build now 就自动把 ServerAuthServe 镜像打包上传到阿里云服务器了

10. 检验成功

Build 成功:



阿里云上检测一下，镜像已成功上传:

阿里云

rmpserverauthservice						
华北2（北京） 公开 本地仓库 ● 正常						
信息	镜像版本					
授权	版本	镜像ID	状态	Digest	镜像大小	最后更新时间
器	latest	3e97d88306e5...	● 正常	120bcffab20b7db4b9e80d9813d4187cb5db408199091ce7c17ca9d226acc4d8	108.147 MB	2019-07-02 10:12:36
版本						
同步						