

🔗 动手实战学Docker (/courses/498)

使用 Docker 运行 MongoDB 和 Redis

1. 课程说明

课程为纯动手实验教程，为了能说清楚实验中的一些操作会加入理论内容。理论内容我们不会写太多，已经有太多好文章了，会精选最值得读的文章推荐给你，在动手实践的同时扎实理论基础。

实验环境中可以联网，不受实验楼网络限制。

2. 学习方法

实验楼的Docker课程包含14个实验，每个实验都提供详细的步骤和截图，适用于有一定Linux系统基础，想快速上手Docker的同学。

学习方法是多实践，多提问。启动实验后按照实验步骤逐步操作，同时理解每一步的详细内容。

如果实验开始部分有推荐阅读的材料，请务必先阅读后再继续实验，理论知识是实践必要的基础。

3. 本节内容简介

在本实验里我们将通过完成 MongoDB 和 Redis 两个容器来学习Dockerfile及Docker的运行机制。

本节中，我们需要依次完成下面几项任务：

1. MongoDB 的安装及配置
2. Redis 的安装及配置
3. Dockerfile 的编写
4. 从 Dockerfile 构建镜像

本次实验的需求是完成 Dockerfile，通过 Dockerfile 创建 MongoDB 或 Redis 应用。Dockerhub 上已经提供了官方的 MongoDB 和 Redis 镜像，本实验仅仅用于学习 Dockerfile 及 Docker 机制。

MongoDB 是一个基于分布式文件存储的数据库。由C++语言编写。旨在为WEB应用提供可扩展的高性能数据存储解决方案。MongoDB 是一个介于关系数据库和非关系数据库之间的产品，是非关系数据库当中功能最丰富，最像关系数据库的。特点是高性能、易部署、易使用，存储数据非常方便。 -来自百度百科

Redis是一个开源的使用ANSI C语言编写、支持网络、可基于内存亦可持久化的日志型、Key-Value数据库，并提供多种语言的API。 -来自百度百科

4. 实验准备

4.1 实验分析

在本实验中，我们除了安装所需的核心服务外，还安装一个 ssh 服务提供便捷的管理。

为了提高 docker build 速度，我们直接使用阿里云的 Ubuntu 源。因此要在 Dockerfile 开始位置增加下面一句命令：

```
RUN echo "deb http://mirrors.cloud.aliyuncs.com/ubuntu/ trusty main universe" > /etc/apt/sources.list
```

4.2 创建 Dockerfile 文件

首先，需要创建一个目录来存放 Dockerfile 文件，目录名称可以任意，在目录里创建Dockerfile文件：

```
cd /home/shiyanlou
mkdir shiyanloumongodb shiyanlouredis
touch shiyanloumongodb/Dockerfile shiyanlouredis/Dockerfile
```



```
shiyanlou@iZ23zslbpciZ: /
$ cd /home/shiyanlou

shiyanlou@iZ23zslbpciZ: ~
$ mkdir shiyanloumongodb shiyanlouredis

shiyanlou@iZ23zslbpciZ: ~
$ touch shiyanloumongodb/Dockerfile shiyanlouredis/Dockerfile

shiyanlou@iZ23zslbpciZ: ~
$
```

The screenshot shows a terminal window with the user 'shiyanlou' on host 'iZ23zslbpciZ'. The user navigates to '/home/shiyanlou', creates two directories 'shiyanloumongodb' and 'shiyanlouredis', and then creates two empty files 'Dockerfile' inside each directory using the 'touch' command. A logo for '实验楼 shiyanlou.com' is visible in the bottom right corner of the terminal window.

使用 vim/gedit 编辑 Dockerfile 文件，根据我们的需求输入内容。

5. Dockerfile 基本框架

5.1 基本框架

按照上一节学习的内容，我们先完成 Dockerfile 基本框架。

👉 动手实战学Docker (/courses/498)

依次输入下面的基本框架内容：

```
# Version 0.1

# 基础镜像
FROM ubuntu:14.04

# 维护者信息
MAINTAINER shiyanlou@shiyanlou.com

# 镜像操作命令
RUN echo "deb http://mirrors.cloud.aliyuncs.com/ubuntu/ trusty main universe" > /etc/apt/sources.list
RUN apt-get update && apt-get install -yqq supervisor && apt-get clean

# 容器启动命令
CMD ["supervisord"]
```

上面的 Dockerfile 创建了一个简单的镜像，并使用 Supervisord 启动服务。

5.2 安装SSH服务

首先安装所需要的软件包：

```
RUN apt-get install -yqq openssh-server openssh-client
```

创建运行目录：

```
RUN mkdir /var/run/sshd
```

设置root密码及允许root通过ssh登陆：

```
RUN echo 'root:shiyanlou' | chpasswd
RUN sed -i 's/PermitRootLogin without-password/PermitRootLogin yes/' /etc/ssh/sshd_config
```

6. 完成 MongoDB Dockerfile

在上述基本的架构下，我们根据需求可以增加新的内容到 Dockerfile 中，完成 MongoDB Dockerfile。

进入到 shiyanloumongodb的目录编辑 Dockerfile：

```
cd /opt/docker/shiyanloumongo498)
vim Dockerfile
```

6.1 安装最新的MongoDB

在 Ubuntu 最新版本下安装 MongoDB 非常简单，参考 MongoDB安装文档 (<https://docs.mongodb.org/manual/tutorial/install-mongodb-on-ubuntu/>)。有两种方法：

方法一是添加mongodb的源，执行 `apt-get install mongodb-org` 就可以安装下面的所有软件包：

1. `mongodb-org-server`：mongod 服务和配置文件
2. `mongodb-org-mongos`：mongos 服务
3. `mongodb-org-shell`：mongo shell工具
4. `mongodb-org-tools`：mongodump，mongoexport等工具

方法二是下载二进制包，然后解压出来就可以。

由于 MongoDB 的官网连接网速问题，我们使用第二种方案，并把最新的 MongoDB 的包放到阿里云上。

MongoDB 的下载链接如下：

```
http://labfile.oss-cn-hangzhou-internal.aliyuncs.com/courses/498/mongodb-linux-x86_64-ubuntu1404-3.2.3.tgz
```

我们完善 Dockerfile，使用 ADD 命令添加压缩包到镜像：

```
RUN mkdir -p /opt
ADD http://labfile.oss-cn-hangzhou-internal.aliyuncs.com/courses/498/mongodb-linux-x86_64-ubuntu1404-3.2.3.tgz /opt/mongodb.tar.gz
RUN cd /opt && tar zxvf mongodb.tar.gz && rm -rf mongodb.tar.gz
RUN mv /opt/mongodb-linux-x86_64-ubuntu1404-3.2.3 /opt/mongodb
```

创建 MongoDB 的数据存储目录：

```
RUN mkdir -p /data/db
```

将 MongoDB 的执行路径添加到环境变量里：

```
ENV PATH=/opt/mongodb/bin:$PATH
```

MongoDB 和 SSH 对外的端口：

EXP-1709-13 动手实战学Docker (/courses/498)

6.2 编写 Supervisor 配置文件

添加 Supervisor 配置文件来启动mongodb和ssh，创建文件 /home/shiyanlou/shiyanloumongodb/supervisord.conf，添加以下内容：

```
[supervisord]
nodaemon=true

[program:mongodb]
command=/opt/mongodb/bin/mongod

[program:ssh]
command=/usr/sbin/sshd -D
```

Dockerfile中增加向镜像内拷贝该文件的命令：

```
COPY supervisord.conf /etc/supervisor/conf.d/supervisord.conf
```

6.3 完整的 Dockerfile

🔗 动手实战学 Docker (/courses/498)

基础镜像

FROM ubuntu:14.04

维护者信息

MAINTAINER shiyanlou@shiyanlou.com

镜像操作命令

RUN echo "deb http://mirrors.cloud.aliyuncs.com/ubuntu/ trusty main universe" > /etc/apt/sources.list

RUN apt-get -yqq update && apt-get install -yqq supervisor

RUN apt-get install -yqq openssh-server openssh-client

RUN mkdir /var/run/ssh

RUN echo 'root:shiyanlou' | chpasswd

RUN sed -i 's/PermitRootLogin without-password/PermitRootLogin yes/' /etc/ssh/sshd_config

RUN mkdir -p /opt

ADD http://labfile.oss-cn-hangzhou-internal.aliyuncs.com/courses/498/mongodb-linux-x86_64-ubuntu1404-3.2.3.tgz /opt/mongodb.tar.gz

RUN cd /opt && tar zxvf mongodb.tar.gz && rm -rf mongodb.tar.gz

RUN mv /opt/mongodb-linux-x86_64-ubuntu1404-3.2.3 /opt/mongodb

RUN mkdir -p /data/db

ENV PATH=/opt/mongodb/bin:\$PATH

COPY supervisord.conf /etc/supervisor/conf.d/supervisord.conf

EXPOSE 27017 22

容器启动命令

CMD ["supervisord"]

7. 完成 Redis Dockerfile

在上述基本的架构下，我们根据需求可以增加新的内容到Dockerfile中，完成 Redis Dockerfile。

进入到 shiyanlouredis 的目录编辑 Dockerfile：

```
cd /home/shiyanlou/shiyanlouredis/  
vim Dockerfile
```

7.1 安装 Redis

由于 MongoDB 中我们已经学习了如何通过二进制压缩包安装最新版本MongoDB的过程，在此安装 Redis 我们直接使用 Ubuntu 源中默认的 Redis 版本。

安装方法非常简单：

👉 动手实战学Docker (/courses/498)

```
RUN apt-get install redis-server
```

添加对外的端口号：

```
EXPOSE 6379 22
```

7.2 编写 Supervisord 配置文件

添加 Supervisord 配置文件来启动 redis-server 和 ssh，创建文件 /home/shiyanlou/shiyanlouredis/supervisord.conf，添加以下内容：

```
[supervisord]
nodaemon=true

[program:redis]
command=/usr/bin/redis-server

[program:ssh]
command=/usr/sbin/sshd -D
```

Dockerfile中增加向镜像内拷贝该文件的命令：

```
COPY supervisord.conf /etc/supervisor/conf.d/supervisord.conf
```

7.3 完整的 Dockerfile

🔗 动手实战学Docker (/courses/498)

基础镜像

FROM ubuntu:latest

维护者信息

MAINTAINER shiyanlou@shiyanlou.com

镜像操作命令

RUN echo "deb http://mirrors.cloud.aliyuncs.com/ubuntu/ trusty main universe" > /etc/apt/sources.list

RUN apt-get -yqq update && apt-get install -yqq supervisor redis-server

RUN apt-get install -yqq openssh-server openssh-client

RUN mkdir /var/run/ssh

RUN echo 'root:shiyanlou' | chpasswd

RUN sed -i 's/PermitRootLogin without-password/PermitRootLogin yes/' /etc/ssh/sshd_config

COPY supervisord.conf /etc/supervisor/conf.d/supervisord.conf

EXPOSE 6379 22

容器启动命令

CMD ["supervisord"]

8. 从 Dockerfile 创建镜像

8.1 创建 MongoDB 镜像

进入到 /home/shiyanlou/shiyanloumongodb/ 目录，执行创建命令。

docker build 执行创建，-t 参数指定镜像名称：

```
docker build -t shiyanloumongodb:0.1 /home/shiyanlou/shiyanloumongodb/
```



```

shiyanolou@iZ23zslbpciZ: ~/shiyanoloumongodb
$ vim supervisord.conf [15:52:11]

shiyanolou@iZ23zslbpciZ: ~/shiyanoloumongodb
$ ls [15:52:35]
Dockerfile  supervisord.conf

shiyanolou@iZ23zslbpciZ: ~/shiyanoloumongodb
$ docker build -t shiyanoloumongodb:0.1 . [15:52:36]
Sending build context to Docker daemon 3.584 kB
Step 1 : FROM ubuntu:latest
--> 14b59d36bae0
Step 2 : MAINTAINER shiyanolou@shiyanolou.com
--> Running in 9e4f533e5a8a
--> aa24f92a5748
Removing intermediate container 9e4f533e5a8a
Step 3 : RUN echo "deb http://mirrors.aliyuncs.com/ubuntu/ trusty main universe"
> /etc/apt/sources.list
--> Running in 85e82144c79e
--> 129b1e62640e
Removing intermediate container 85e82144c79e
Step 4 : RUN apt-get -yqq update && apt-get -yqq supervisor
--> Running in 4615c8d3d205

```



docker images 查看创建的新镜像已经出现在了镜像列表中：

```

shiyanolou@iZ23zslbpciZ: ~/shiyanoloumongodb
$ docker images [15:59:25]

```

REPOSITORY	TAG	IMAGE ID	CREATED
shiyanoloumongodb	0.1	b68f8fcb29aa	6 seconds ago
ubuntu	latest	14b59d36bae0	2 weeks ago
redis	latest	fb46ec1d66e0	3 weeks ago
registry	latest	bca04f698ba8	6 weeks ago

SIZE
813.6 MB
188 MB
151.3 MB
422.9 MB



由该镜像创建新的容器 mongodb：

```

shiyanolou@iZ23zslbpciZ: ~/shiyanoloumongodb
$ docker run -P -d --name mongodb shiyanoloumongodb:0.1 [15:59:27]
626662e6c4997079f3a787a030e0d666d9b4e692e6b8ed527019316dfc37e324

shiyanolou@iZ23zslbpciZ: ~/shiyanoloumongodb
$ docker ps [16:00:17]

```

CONTAINER ID	IMAGE	COMMAND	CREATED
626662e6c499	shiyanoloumongodb:0.1	"supervisord"	3 seconds ago

STATUS PORTS NAMES
Up 2 seconds 0.0.0.0:32768->27017/tcp mongodb



上述 docker ps 命令的输出可以看到 MongoDB 的端口号已经被自动映射到了本地的 32768 端口，后续步骤我们对 MongoDB 是否启动进行测试。

打开 Xfce 终端中输入下面的命令连接 mongodb 容器中的服务：

👉 动手实战学Docker (/courses/498)

```
shiyanolou@iZ23zslbpciZ: ~/shiyanoloumongodb
$ mongo --host 127.0.0.1 --port 32768 [16:01:51]
MongoDB shell version: 2.4.9
connecting to: 127.0.0.1:32768/test
Server has startup warnings:
2016-03-09T08:00:21.486+0000 I CONTROL [initandlisten] ** WARNING: You are running this process as the root user, which is not recommended.
2016-03-09T08:00:21.486+0000 I CONTROL [initandlisten]
2016-03-09T08:00:21.487+0000 I CONTROL [initandlisten]
2016-03-09T08:00:21.487+0000 I CONTROL [initandlisten] ** WARNING: /sys/kernel/mm/transparent_hugepage/enabled is 'always'.
2016-03-09T08:00:21.487+0000 I CONTROL [initandlisten] ** We suggest setting it to 'never'
2016-03-09T08:00:21.487+0000 I CONTROL [initandlisten]
2016-03-09T08:00:21.487+0000 I CONTROL [initandlisten] ** WARNING: /sys/kernel/mm/transparent_hugepage/defrag is 'always'.
2016-03-09T08:00:21.487+0000 I CONTROL [initandlisten] ** We suggest setting it to 'never'
2016-03-09T08:00:21.487+0000 I CONTROL [initandlisten]
```

如果提示 command not found mongo , 可使用 `sudo apt-get install -y mongodb` 安装。

8.2 创建 Redis 镜像

进入到 `/home/shiyanolou/shiyanolouredis/` 目录，执行创建命令。

`docker build` 执行创建，`-t` 参数指定镜像名称：

```
Terminal 终端 - docker build -t shiyanolouredis:0.1 .
文件(F) 编辑(E) 视图(V) 终端(T) 标签(A) 帮助(H)

shiyanolou@iZ23zslbpciZ: ~/shiyanolouredis
$ ls [16:07:36]
Dockerfile  supervisord.conf

shiyanolou@iZ23zslbpciZ: ~/shiyanolouredis
$ docker build -t shiyanolouredis:0.1 . [16:07:37]
Sending build context to Docker daemon 3.584 kB
Step 1 : FROM ubuntu:latest
---> 14b59d36bae0
Step 2 : MAINTAINER shiyanolou@shiyanolou.com
---> Using cache
---> aa24f92a5748
Step 3 : RUN echo "deb http://mirrors.aliyuncs.com/ubuntu/ trusty main universe"
> /etc/apt/sources.list
---> Using cache
---> 129b1e62640e
Step 4 : RUN apt-get -yqq update && apt-get install -yqq supervisor redis-server
---> Running in 1c1272d0d129
```

`docker images` 查看创建的新镜像已经出现在了镜像列表中：

👉 动手实战学Docker (/courses/498)

```
shianlou@iZ23zslbpciZ: ~/shianlouredis [16:06:15]
$ docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED
shianlouredis	0.1	fa357edc0cfc	5 seconds ago
shianloumongodb	0.1	b68f8fcb29aa	6 minutes ago
ubuntu	latest	14b59d36bae0	2 weeks ago
redis	latest	fb46ec1d66e0	3 weeks ago
registry	latest	bca04f698ba8	6 weeks ago

244.4 MB
813.6 MB
188 MB
151.3 MB
422.9 MB

实验楼
shianlou.com

由该镜像创建新的容器redis：

```
shianlou@iZ23zslbpciZ: ~/shianlouredis [16:08:48]
$ docker run -P -d --name redis shianlouredis:0.1
a4b558b9bdbbaa2e736e0b36ddc7930a2cef066b2a6c781b77ad15f67593fb74

shianlou@iZ23zslbpciZ: ~/shianlouredis [16:09:39]
$ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	NAMES
a4b558b9bdbb	shianlouredis:0.1	"supervisord"	5 seconds ago	redis
626662e6c499	shianloumongodb:0.1	"supervisord"	9 minutes ago	mongodb

Up 4 seconds
Up 9 minutes

实验楼
shianlou.com

上述 `docker ps` 命令的输出可以看到 redis 的端口号已经被自动映射到了本地的 32769 端口，SSH 服务的端口号也映射到了 32770 端口。

打开 Xfce 终端中输入下面的命令连接 redis 容器中的 ssh 和 redis 服务：


```
shiyanolou@iZ23zslbpciZ: ~/shiyanolouredis
$ ssh root@127.0.0.1 -p 32770 [16:09:43]
The authenticity of host '[127.0.0.1]:32770 ([127.0.0.1]:32770)' can't be established.
ECDSA key fingerprint is d5:c0:b3:68:78:f5:62:4b:90:a1:9b:34:1a:cf:cf:a6.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '[127.0.0.1]:32770' (ECDSA) to the list of known hosts.
root@127.0.0.1's password:
Welcome to Ubuntu 14.04 LTS (GNU/Linux 3.13.0-77-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.
root@a4b558b9bdbb:~#
```

```
shiyanolou@iZ23zslbpciZ: ~/shiyanolouredis
$ redis-cli -h 127.0.0.1 -p 32769
127.0.0.1:32769>
127.0.0.1:32769> set shiyanolou test
OK
127.0.0.1:32769> get shiyanolou
"test"
127.0.0.1:32769>
```

如果提示 `command not found redis-cli` , 就使用 `sudo apt-get install -y redis-server` 安装。

9. 总结

本节实验中我们学习了以下内容：

1. MongoDB 的安装
2. Redis 的安装
3. Dockerfile 的编写
4. 从 Dockerfile 构建镜像

请务必保证自己能够动手完成整个实验，只看文字很简单，真正操作的时候会遇到各种各样的问题，解决问题的过程才是收获的过程。

**本课程内容，由作者授权实验楼发布，未经允许，禁止转载、下载及非法传播。*

👉 动手实战学Docker (/courses/498) 本课程为训练营课程，查看完整
内容请

加入训练营 (/courses/498)

上一节：编写 DockerFile (/courses/498/labs/1708/document)

下一节：Docker运行Wordpress (/courses/498/labs/1710/document)