

Docker入门基础之镜像使用

当运行容器时，使用的镜像如果在本地中不存在，docker 就会自动从 docker 镜像仓库中下载，默认是从 Docker Hub 公共镜像源下载。

下面我们来学习：

- 1、管理和使用本地 Docker 主机镜像
- 2、创建镜像

列出镜像列表

我们可以使用 `docker images` 来列出存在于宿主机上的镜像。

```
1 root@ubuntu:~# docker images
2 REPOSITORY          TAG          IMAGE ID          CREATED
3 alpine              latest      055936d39205     3 weeks ago
4 mysql               5.7         7faa3c53e6d6     3 weeks ago
5 ubuntu              15.04      d1b55fd07600     3 years ago
6 ubuntu              16.04      2a697363a870     2 weeks ago
```

各个选项说明：

- **REPOSITORY**：表示镜像的仓库源
- **TAG**：镜像的标签
- **IMAGE ID**：镜像ID
- **CREATED**：镜像创建时间
- **SIZE**：镜像大小

同一仓库源可以有多个 TAG，代表这个仓库源的不同个版本，如ubuntu仓库源里，有16.04、15.04等多个不同的版本，我们使用 REPOSITORY:TAG 来定义不同的镜像。

以相应的镜像启动容器

我们如果要使用版本为16.04的ubuntu系统镜像来运行容器时，命令如下：

```
1 root@ubuntu:~# docker run -it ubuntu:16.04 sh
2 #
```

如果要使用版本为15.04的ubuntu系统镜像，则命令如下：

```
1 root@ubuntu:~# docker run -it ubuntu:15.04 sh
2 #
```

各个参数解析:

- **docker**: Docker 的二进制执行文件。
- **run**:与前面的 docker 组合来运行一个容器。
- **-it**:其实是两个参数组成, **-i**和 **-t**, **-i**:允许你对容器内的标准输入 (STDIN) 进行交互。 **-t**:在新容器内指定一个伪终端或终端。
- **ubuntu:15.04**指定要运行的镜像, Docker首先从本地主机上查找镜像是否存在, 如果不存在, Docker就会从镜像仓库 Docker Hub 下载公共镜像。
- **sh**:执行命令。

获取一个新镜像

当我们在宿主机上使用一个不存在的镜像时 Docker 就会自动下载这个镜像。如果我们想预先下载这个镜像, 我们可以使用 `docker pull` 命令来下载它。

```
1 root@ubuntu:~# docker pull ubuntu:18.04
2 18.04: Pulling from library/ubuntu
3 6abc03819f3e: Pull complete
4 05731e63f211: Pull complete
5 0bd67c50d6be: Pull complete
6 Digest: sha256:f08638ec7ddc90065187e7eabdfac3c96e5ff0f6b2f1762cf31a4f49b53000a5
7 Status: Downloaded newer image for ubuntu:18.04
```

下载完成后, 我们可以直接使用这个镜像来运行容器。

查找镜像

我们可以从 **Docker Hub** 网站来搜索镜像, Docker Hub 网址为: <https://hub.docker.com/> 我们也可以使用 `docker search` 命令来搜索镜像。比如我们需要一个 **httpd** 的镜像来作为我们的web服务。我们可以通过 `docker search` 命令搜索 **httpd** 来寻找适合我们的镜像。

```
root@ubuntu:~# docker search httpd
NAME                DESCRIPTION                STARS     OFFICIAL   AUTOMATED
httpd               The Apache HTTP Server Project 2498     [OK]
centos/httpd       The Apache HTTP Server Project 23
centos/httpd-24-centos7 Platform for running Apache httpd 2.4 or bui... 22
armhf/httpd        The Apache HTTP Server Project 8
lolhens/httpd      Apache httpd 2 Server        2         [OK]
salim1983hoop/httpd24 Dockerfile running apache config 2         [OK]
polinux/httpd-php  Apache with PHP in Docker (Supervisor, CentO... 2         [OK]
rgielen/httpd-image-simple Docker image for simple Apache httpd based o... 1         [OK]
lead4good/httpd-fpm httpd server which connects via fcgi proxy h... 1         [OK]
itsziget/httpd24   Extended HTTPD Docker image based on the off... 0         [OK]
dockerpinata/httpd 0
trollin/httpd      0
solsson/httpd-openidc mod_auth_openidc on official httpd image, ve... 0         [OK]
izdock/httpd       Production ready Apache HTTPD Web Server + m... 0
amd64/httpd        The Apache HTTP Server Project 0
manageiq/httpd_configmap_generator Httpd Configmap Generator 0         [OK]
publici/httpd      httpd:latest 0         [OK]
buzzardev/httpd    Based on the official httpd image 0         [OK]
appertly/httpd     Customized Apache HTTPD that uses a PHP-FPM ... 0         [OK]
manasip/httpd      0
tugboatqa/httpd   The Apache HTTP Server Project 0
ppc04le/httpd     The Apache HTTP Server Project 0
alvistack/httpd   Docker Image Packaging for Apache 0         [OK]
manageiq/httpd     Container with httpd, built on CentOS for Ma... 0         [OK]
interlutions/httpd httpd docker image with debian-based config ... 0         [OK]
root@ubuntu:~#
```

NAME:镜像仓库源的名称

DESCRIPTION:镜像的描述

OFFICIAL:是否docker官方发布

拉取httpd镜像

我们决定使用上图中的httpd 官方版本的镜像，使用命令 `docker pull` 来下载镜像。

```
1 root@ubuntu:~# docker pull httpd
2 Using default tag: latest
3 latest: Pulling from library/httpd
4 743f2d6c1f65: Already exists
5 c92eb69846a6: Pull complete
6 2211b052800a: Pull complete
7 aed180197314: Pull complete
8 7c472a4980a7: Pull complete
9 Digest: sha256:a35ad614c1ffc6fe931f12dc42b682edbdcc62cf78d8edc41499dd90ef0f8003
10 Status: Downloaded newer image for httpd:latest
```

下载完成后，我们就可以使用这个镜像了。

创建镜像

当我们从docker镜像仓库中下载的镜像不能满足我们的需求时，我们可以通过以下两种方式对镜像进行更改。

- 1.从已经创建的容器中更新镜像，并且提交这个镜像
- 2.使用 **Dockerfile** 指令来创建一个新的镜像

更新镜像

更新镜像之前，我们需要使用镜像来创建一个容器。

```
1 root@ubuntu:~# docker run -it ubuntu:16.04 sh
2 #
```

在运行的容器内使用 `apt-get update` 命令进行更新。

```
1 # apt-get update
2 Get:1 http://archive.ubuntu.com/ubuntu xenial InRelease [247 kB]
3 Get:2 http://security.ubuntu.com/ubuntu xenial-security InRelease [109 kB]
4 Get:3 http://archive.ubuntu.com/ubuntu xenial-updates InRelease [109 kB]
5 Get:4 http://archive.ubuntu.com/ubuntu xenial-backports InRelease [107 kB]
6 Get:5 http://archive.ubuntu.com/ubuntu xenial/main amd64 Packages [1558 kB]
7 Get:6 http://security.ubuntu.com/ubuntu xenial-security/main amd64 Packages [844 kB]
8 Get:7 http://archive.ubuntu.com/ubuntu xenial/restricted amd64 Packages [14.1 kB]
9 Get:8 http://archive.ubuntu.com/ubuntu xenial/universe amd64 Packages [9827 kB]
10 Get:9 http://archive.ubuntu.com/ubuntu xenial/multiverse amd64 Packages [176 kB]
11 Get:10 http://archive.ubuntu.com/ubuntu xenial-updates/main amd64 Packages [1237 kB]
12 Get:11 http://security.ubuntu.com/ubuntu xenial-security/restricted amd64 Packages
    [12.7 kB]
```

```
13 Get:12 http://security.ubuntu.com/ubuntu xenial-security/universe amd64 Packages [556
    kB]
14 Get:13 http://archive.ubuntu.com/ubuntu xenial-updates/restricted amd64 Packages [13.1
    kB]
15 Get:14 http://archive.ubuntu.com/ubuntu xenial-updates/universe amd64 Packages [967
    kB]
16 Get:15 http://archive.ubuntu.com/ubuntu xenial-updates/multiverse amd64 Packages [19.1
    kB]
17 Get:16 http://archive.ubuntu.com/ubuntu xenial-backports/main amd64 Packages [7942 B]
18 Get:17 http://archive.ubuntu.com/ubuntu xenial-backports/universe amd64 Packages [8532
    B]
19 Get:18 http://security.ubuntu.com/ubuntu xenial-security/multiverse amd64 Packages
    [6113 B]
20 Fetched 15.8 MB in 8s (1933 kB/s)
21 Reading package lists... Done
```

在完成操作之后，输入 `exit` 命令来退出这个容器。

通过 `docker ps -a` 查看所有容器：

```
1 root@ubuntu:~# docker ps -a
2 CONTAINER ID        IMAGE               COMMAND             CREATED
   STATUS             PORTS              NAMES
3 12847d9f5071        ubuntu:16.04       "sh"                2
  minutes ago       Exited (0) 43 seconds ago
  peaceful_edison
4 3cec6f5e47c0        httpd              "httpd-foreground"  5
  minutes ago       Exited (0) 5 minutes ago
  vigilant_heisenberg
```

此时ID为**12847d9f5071**的容器，是按我们的需求更改的容器。我们可以通过命令 `docker commit` 来提交容器副本。

```
1 root@ubuntu:~# docker commit -m="has update" -a="yeqing112" 12847d9f5071
  yeqing112/ubuntu:v2
2 sha256:5cce1167ee4c083ff9b87e61a58081805e76d015a12a4811bd152263aa013860
```

各个参数说明：

- **-m**:提交的描述信息。
- **-a**:指定镜像作者。
- **12847d9f5071**: 容器ID。
- **yeqing112/ubuntu:v2**:指定要创建的目标镜像名。

我们可以使用 `docker images` 命令来查看我们的新镜像 **yeqing112/ubuntu:v2**：

```

1 root@ubuntu:~# docker images
2 REPOSITORY          TAG          IMAGE ID          CREATED
3 yeqing112/ubuntu    v2           5cce1167ee4c     2 minutes ago
4 ubuntu              16.04       2a697363a870     2 weeks ago
5 ubuntu              16.04       2a697363a870     2 weeks ago
6 ubuntu              16.04       2a697363a870     2 weeks ago
7 ubuntu              16.04       2a697363a870     2 weeks ago
8 ubuntu              16.04       2a697363a870     2 weeks ago
9 ubuntu              16.04       2a697363a870     2 weeks ago
10 ubuntu             16.04       2a697363a870     2 weeks ago
11 ubuntu             16.04       2a697363a870     2 weeks ago
12 ubuntu             16.04       2a697363a870     2 weeks ago
13 ubuntu             16.04       2a697363a870     2 weeks ago
14 ubuntu             16.04       2a697363a870     2 weeks ago
15 ubuntu             16.04       2a697363a870     2 weeks ago
16 ubuntu             16.04       2a697363a870     2 weeks ago
17 ubuntu             16.04       2a697363a870     2 weeks ago
18 ubuntu             16.04       2a697363a870     2 weeks ago
19 ubuntu             16.04       2a697363a870     2 weeks ago
20 ubuntu             16.04       2a697363a870     2 weeks ago
21 ubuntu             16.04       2a697363a870     2 weeks ago
22 ubuntu             16.04       2a697363a870     2 weeks ago
23 ubuntu             16.04       2a697363a870     2 weeks ago
24 ubuntu             16.04       2a697363a870     2 weeks ago
25 ubuntu             16.04       2a697363a870     2 weeks ago
26 ubuntu             16.04       2a697363a870     2 weeks ago
27 ubuntu             16.04       2a697363a870     2 weeks ago
28 ubuntu             16.04       2a697363a870     2 weeks ago
29 ubuntu             16.04       2a697363a870     2 weeks ago
30 ubuntu             16.04       2a697363a870     2 weeks ago
31 ubuntu             16.04       2a697363a870     2 weeks ago
32 ubuntu             16.04       2a697363a870     2 weeks ago
33 ubuntu             16.04       2a697363a870     2 weeks ago
34 ubuntu             16.04       2a697363a870     2 weeks ago
35 ubuntu             16.04       2a697363a870     2 weeks ago
36 ubuntu             16.04       2a697363a870     2 weeks ago
37 ubuntu             16.04       2a697363a870     2 weeks ago
38 ubuntu             16.04       2a697363a870     2 weeks ago
39 ubuntu             16.04       2a697363a870     2 weeks ago
40 ubuntu             16.04       2a697363a870     2 weeks ago
41 ubuntu             16.04       2a697363a870     2 weeks ago
42 ubuntu             16.04       2a697363a870     2 weeks ago
43 ubuntu             16.04       2a697363a870     2 weeks ago
44 ubuntu             16.04       2a697363a870     2 weeks ago
45 ubuntu             16.04       2a697363a870     2 weeks ago
46 ubuntu             16.04       2a697363a870     2 weeks ago
47 ubuntu             16.04       2a697363a870     2 weeks ago
48 ubuntu             16.04       2a697363a870     2 weeks ago
49 ubuntu             16.04       2a697363a870     2 weeks ago
50 ubuntu             16.04       2a697363a870     2 weeks ago
51 ubuntu             16.04       2a697363a870     2 weeks ago
52 ubuntu             16.04       2a697363a870     2 weeks ago
53 ubuntu             16.04       2a697363a870     2 weeks ago
54 ubuntu             16.04       2a697363a870     2 weeks ago
55 ubuntu             16.04       2a697363a870     2 weeks ago
56 ubuntu             16.04       2a697363a870     2 weeks ago
57 ubuntu             16.04       2a697363a870     2 weeks ago
58 ubuntu             16.04       2a697363a870     2 weeks ago
59 ubuntu             16.04       2a697363a870     2 weeks ago
60 ubuntu             16.04       2a697363a870     2 weeks ago
61 ubuntu             16.04       2a697363a870     2 weeks ago
62 ubuntu             16.04       2a697363a870     2 weeks ago
63 ubuntu             16.04       2a697363a870     2 weeks ago
64 ubuntu             16.04       2a697363a870     2 weeks ago
65 ubuntu             16.04       2a697363a870     2 weeks ago
66 ubuntu             16.04       2a697363a870     2 weeks ago
67 ubuntu             16.04       2a697363a870     2 weeks ago
68 ubuntu             16.04       2a697363a870     2 weeks ago
69 ubuntu             16.04       2a697363a870     2 weeks ago
70 ubuntu             16.04       2a697363a870     2 weeks ago
71 ubuntu             16.04       2a697363a870     2 weeks ago
72 ubuntu             16.04       2a697363a870     2 weeks ago
73 ubuntu             16.04       2a697363a870     2 weeks ago
74 ubuntu             16.04       2a697363a870     2 weeks ago
75 ubuntu             16.04       2a697363a870     2 weeks ago
76 ubuntu             16.04       2a697363a870     2 weeks ago
77 ubuntu             16.04       2a697363a870     2 weeks ago
78 ubuntu             16.04       2a697363a870     2 weeks ago
79 ubuntu             16.04       2a697363a870     2 weeks ago
80 ubuntu             16.04       2a697363a870     2 weeks ago
81 ubuntu             16.04       2a697363a870     2 weeks ago
82 ubuntu             16.04       2a697363a870     2 weeks ago
83 ubuntu             16.04       2a697363a870     2 weeks ago
84 ubuntu             16.04       2a697363a870     2 weeks ago
85 ubuntu             16.04       2a697363a870     2 weeks ago
86 ubuntu             16.04       2a697363a870     2 weeks ago
87 ubuntu             16.04       2a697363a870     2 weeks ago
88 ubuntu             16.04       2a697363a870     2 weeks ago
89 ubuntu             16.04       2a697363a870     2 weeks ago
90 ubuntu             16.04       2a697363a870     2 weeks ago
91 ubuntu             16.04       2a697363a870     2 weeks ago
92 ubuntu             16.04       2a697363a870     2 weeks ago
93 ubuntu             16.04       2a697363a870     2 weeks ago
94 ubuntu             16.04       2a697363a870     2 weeks ago
95 ubuntu             16.04       2a697363a870     2 weeks ago
96 ubuntu             16.04       2a697363a870     2 weeks ago
97 ubuntu             16.04       2a697363a870     2 weeks ago
98 ubuntu             16.04       2a697363a870     2 weeks ago
99 ubuntu             16.04       2a697363a870     2 weeks ago
100 ubuntu            16.04       2a697363a870     2 weeks ago

```

从上面的信息中可以看到我们更新过的镜像，文件体积明显比之前大了一些。

构建镜像

除了上面更新镜像的方法，我们还可以使用命令 `docker build`，从零开始来构建一个新的镜像。为此，我们需要创建一个 **Dockerfile** 文件，其中包含一组指令来告诉 Docker 如何构建我们的镜像。

```

1 root@ubuntu:~# cat Dockerfile
2 FROM ubuntu:16.04
3 MAINTAINER Fisher "service@urlos.com"
4
5 RUN apt-get update
6
7 EXPOSE 22
8 EXPOSE 80
9
10 CMD /bin/bash

```

每一个指令都会在镜像上创建一个新的层，每一个指令的前缀都必须是大写的。

第一条**FROM**，指定使用哪个镜像源

RUN 指令告诉docker 在镜像内执行命令，安装了什么

这里我们还是一样，执行一个 `apt-get update` 更新命令

然后，我们使用 **Dockerfile** 文件，通过 `docker build -t yeqing112/ubuntu:v3 .` 命令来构建一个新镜像。

```

1 root@ubuntu:~# root@ubuntu:/home# docker build -t yeqing112/ubuntu:v3 .
2 Sending build context to Docker daemon 422.4kB
3 Step 1/6 : FROM ubuntu:16.04
4 ---> 2a697363a870
5 Step 2/6 : MAINTAINER Fisher "service@urlos.com"
6 ---> Running in 92229e7a091a
7 Removing intermediate container 92229e7a091a
8 ---> 9a89f4debc7d
9 Step 3/6 : RUN apt-get update
10 ---> Running in d709ef335784
11 Get:1 http://archive.ubuntu.com/ubuntu xenial InRelease [247 kB]
12 Get:2 http://security.ubuntu.com/ubuntu xenial-security InRelease [109 kB]
13 Get:3 http://archive.ubuntu.com/ubuntu xenial-updates InRelease [109 kB]
14 Get:4 http://archive.ubuntu.com/ubuntu xenial-backports InRelease [107 kB]
15 Get:5 http://security.ubuntu.com/ubuntu xenial-security/main amd64 Packages [844 kB]
16 Get:6 http://archive.ubuntu.com/ubuntu xenial/main amd64 Packages [1558 kB]
17 Get:7 http://archive.ubuntu.com/ubuntu xenial/restricted amd64 Packages [14.1 kB]

```

```
18 Get:8 http://archive.ubuntu.com/ubuntu xenial/universe amd64 Packages [9827 kB]
19 Get:9 http://security.ubuntu.com/ubuntu xenial-security/restricted amd64 Packages
    [12.7 kB]
20 Get:10 http://security.ubuntu.com/ubuntu xenial-security/universe amd64 Packages [556
    kB]
21 Get:11 http://archive.ubuntu.com/ubuntu xenial/multiverse amd64 Packages [176 kB]
22 Get:12 http://archive.ubuntu.com/ubuntu xenial-updates/main amd64 Packages [1237 kB]
23 Get:13 http://archive.ubuntu.com/ubuntu xenial-updates/restricted amd64 Packages [13.1
    kB]
24 Get:14 http://archive.ubuntu.com/ubuntu xenial-updates/universe amd64 Packages [967
    kB]
25 Get:15 http://archive.ubuntu.com/ubuntu xenial-updates/multiverse amd64 Packages [19.1
    kB]
26 Get:16 http://archive.ubuntu.com/ubuntu xenial-backports/main amd64 Packages [7942 B]
27 Get:17 http://archive.ubuntu.com/ubuntu xenial-backports/universe amd64 Packages [8532
    B]
28 Get:18 http://security.ubuntu.com/ubuntu xenial-security/multiverse amd64 Packages
    [6113 B]
29 Fetched 15.8 MB in 5s (3051 kB/s)
30 Reading package lists...
31 Removing intermediate container d709ef335784
32 ----> 540751496556
33 Step 4/6 : EXPOSE 22
34 ----> Running in 9ae528bc083f
35 Removing intermediate container 9ae528bc083f
36 ----> cfccc621a52d
37 Step 5/6 : EXPOSE 80
38 ----> Running in efd88e9349c1
39 Removing intermediate container efd88e9349c1
40 ----> 428cb5221a06
41 Step 6/6 : CMD /bin/bash
42 ----> Running in 2bc08b664f97
43 Removing intermediate container 2bc08b664f97
44 ----> 9c55ea9b9ee8
45 Successfully built 9c55ea9b9ee8
46 Successfully tagged yeqing112/ubuntu:v3
```

参数说明:

- **-t**: 指定要创建的目标镜像名
- **.**: Dockerfile 文件所在目录, 可以指定Dockerfile 的绝对路径

构建成功后, 使用 [docker images](#) 查看镜像:

```
1 root@ubuntu:~# docker images
2 REPOSITORY          TAG          IMAGE ID          CREATED
3 yeqing112/ubuntu    v3           9c55ea9b9ee8     9 seconds ago
4 yeqing112/ubuntu    v2           5cce1167ee4c     16 minutes ago
5 ubuntu              16.04       2a697363a870     2 weeks ago
6
```

由上面的信息看出，新的镜像v3与v2文件体积是一样的，两个镜像都只是执行了 `apt-get update`

设置镜像标签

我们可以使用 `docker tag` 命令，为镜像添加一个新的标签。

```
1 | root@ubuntu:/home# docker tag 9c55ea9b9ee8 yeqing112/ubuntu:v3-1
```

`docker tag` 镜像ID，这里是 **9c55ea9b9ee8**，用户名称、镜像源名(repository name)和新的标签名(tag)。

使用 `docker images` 命令可以看到，ID为**9c55ea9b9ee8**的镜像多了一个v3-1的标签。

```
1 | root@ubuntu:/home# docker images
2 | REPOSITORY          TAG          IMAGE ID          CREATED
3 | yeqing112/ubuntu    v3          9c55ea9b9ee8     21 minutes ago
4 | yeqing112/ubuntu    v3-1       9c55ea9b9ee8     21 minutes ago
5 | yeqing112/ubuntu    v2          5cce1167ee4c     38 minutes ago
6 | ubuntu              16.04      2a697363a870     2 weeks ago
   | 144MB
   | 144MB
   | 144MB
   | 119MB
```