

## docker--Dockerfile镜像制作

2.Dockerfile文件总结

3. Dockerfile中各个指令的介绍

4. 编写dockerfile

5. 构建镜像

6. 启动构建好的镜像

ENTRYPOINT 和 CMD的区别

# docker--Dockerfile镜像制作

## 1.概述

创建Docker镜像的方式有三种

- docker commit命令：由容器生成镜像；
- Dockerfile文件+docker build命令；
- 从本地文件系统导入：OpenVZ的模板。

## 2.Dockerfile文件总结

Dockerfile 由一行行命令语句组成，并且支持以 # 开头的注释行。

一般地，Dockerfile 分为四部分：基础镜像信息、维护者信息、镜像操作指令和容器启动时执行指令。

四部分	指令
基础镜像信息	FROM
维护者信息	MAINTAINER
镜像操作指令	RUN、COPY、ADD、EXPOSE等
容器启动时执行指令	CMD、ENTRYPOINT

Dockerfile文件的第一条指令必须是FROM，其后可以是各种镜像的操作指令，最后是CMD或ENTRYPOINT指定容器启动时执行的命令。

### 3. Dockerfile中各个指令的介绍

1	<b>FROM</b>
2	格式为 FROM <image>或FROM <image>:<tag>。
3	第一条指令必须为 FROM 指令。并且，如果在同一个Dockerfile中创建多个镜像时，可以使用
4	
5	<b>MAINTAINER</b>
6	格式为 MAINTAINER <name>，指定维护者信息。
7	
8	<b>RUN</b>
9	格式为 RUN <command> 或 RUN ["executable", "param1", "param2"]。
10	前者将在 shell 终端中运行命令，即 /bin/sh -c；后者则使用 exec 执行。指定使用其它终端可
11	如 RUN ["/bin/bash", "-c", "echo hello"]。
12	每条 RUN 指令将在当前镜像基础上执行指定命令，并提交为新的镜像。当命令较长时可以使
13	
14	<b>CMD</b>
15	支持三种格式
16	CMD ["executable","param1","param2"] 使用 exec 执行，推荐方式；
17	CMD command param1 param2 在 /bin/sh 中执行，提供给需要交互的应用；
18	CMD ["param1","param2"] 提供给 ENTRYPOINT 的默认参数；
19	指定启动容器时执行的命令，每个 Dockerfile 只能有一条 CMD 命令。如果指定了多条命令，
20	如果用户启动容器时候指定了运行的命令，则会覆盖掉 CMD 指定的命令。
21	
22	<b>EXPOSE</b>
23	格式为 EXPOSE <port> [<port>...]。
24	告诉 Docker 服务端容器暴露的端口号，供互联系统使用。在启动容器时需要通过 -P，Dock
25	口。
26	
27	<b>ENV</b>
28	格式为 ENV <key> <value>。指定一个环境变量，会被后续 RUN 指令使用，并在容器运行
29	例如
30	ENV PG_MAJOR 9.3
31	ENV PG_VERSION 9.3.4
32	RUN curl -SL http://example.com/postgres-\$PG_VERSION.tar.xz   tar -xJC /usr/src/pos
33	ENV PATH /usr/local/postgres-\$PG_MAJOR/bin:\$PATH
34	
35	<b>ADD</b>
36	格式为 ADD <src> <dest>。
37	该命令将复制指定的 <src> 到容器中的 <dest>。其中 <src> 可以是Dockerfile所在目录的
38	是一个 tar 文件（自动解压为目录）。
39	
40	<b>COPY</b>
41	格式为 COPY <src> <dest>。
42	复制本地主机的 <src>（为 Dockerfile 所在目录的相对路径）到容器中的 <dest>。
43	当使用本地目录为源目录时，推荐使用 COPY。
44	
45	<b>ENTRYPOINT</b>

两种格式：

```
ENTRYPOINT ["executable", "param1", "param2"]
```

```
ENTRYPOINT command param1 param2 ( shell中执行 )。
```

配置容器启动后执行的命令，并且不可被 docker run 提供的参数覆盖。

每个 Dockerfile 中只能有一个 ENTRYPOINT，当指定多个时，只有最后一个起效。

## VOLUME

格式为 VOLUME ["/data"]。

创建一个可以从本地主机或其他容器挂载的挂载点，一般用来存放数据库和需要保持的数据等

## USER

格式为 USER daemon。

指定运行容器时的用户名或 UID，后续的 RUN 也会使用指定用户。

当服务不需要管理员权限时，可以通过该命令指定运行用户。并且可以在之前创建所需要的用户 postgres && useradd -r -g postgres postgres。要临时获取管理员权限可以使用 gosu，

## WORKDIR

定义工作目录，如果容器中没有此目录，会自动创建，为后续的 RUN、CMD、ENTRYPOINT 指令可以使用多个 WORKDIR 指令，后续命令如果参数是相对路径，则会基于之前命令指定的路径

```
WORKDIR /a
```

```
WORKDIR b
```

```
WORKDIR c
```

```
RUN pwd
```

则最终路径为 /a/b/c。

## ONBUILD

格式为 ONBUILD [INSTRUCTION]。

配置当所创建的镜像作为其它新创建镜像的基础镜像时，所执行的操作指令。

例如，Dockerfile 使用如下的内容创建了镜像 image-A。

```
[...]
```

```
ONBUILD ADD . /app/src
```

```
ONBUILD RUN /usr/local/bin/python-build --dir /app/src
```

```
[...]
```

如果基于 image-A 创建新的镜像时，新的 Dockerfile 中使用 FROM image-A 指定基础镜像时在后面添加了两条指令。

```
FROM image-A #Automatically run the followingADD . /app/srcRUN /usr/local/bin/py
```

使用 ONBUILD 指令的镜像，推荐在标签中注明，例如 ruby:1.9-onbuild。

98  
99  
100  
101  
102  
103  
104  
105



## 4. 编写dockerfile

```
1 [root@ CentOS7-200 ~]# cat Dockerfile
2 FROM centos:6.8
3
4 MAINTAINER 18535756726@163.com
5
6 ADD nginx-1.10.3.tar.gz /opt
7
8 RUN yum -y install openssl openssl-devel pcre pcre-devel wget gcc gcc-c++ make
9 RUN useradd -s /sbin/nologin -M nginx
10
11 WORKDIR /opt/nginx-1.10.3
12
13 RUN ./configure --prefix=/opt/nginx --user=nginx --group=nginx --with-http_stub_s
14 -with-http_realip_module && make && make install
15
16 ENV PATH /opt/nginx/sbin:$PATH
17 EXPOSE 80
18 CMD /bin/sh -c 'nginx -g "daemon off;"'
```

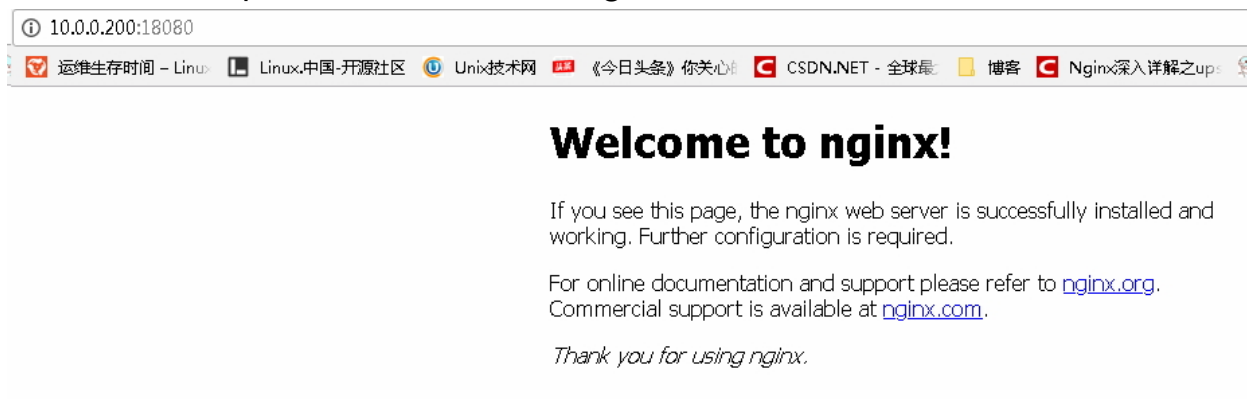
( 注意Dockerfile的D需要大写 )

## 5. 构建镜像

```
1 [root@ CentOS7-200 ~]# ll
2 total 900
3 -rw-----. 1 root root 1893 Jul 9 2017 anaconda-ks.cfg
4 -rw-r--r-- 1 root root 445 Apr 17 23:11 Dockerfile
5 -rw-r--r-- 1 root root 911509 Sep 27 2017 nginx-1.10.3.tar.gz
6 [root@ CentOS7-200 ~]# docker build -t centos6.8-nginx:v1 .
7 [root@ CentOS7-200 ~]# docker images
8
9 REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
10 centos6.8-nginx     v1                 38bdb3270647       24 minutes ago     497MB
11 centos              6.8               6704d778b3ba       5 months ago       195MB
```

## 6. 启动构建好的镜像

`docker run -d -p 18080:80 centos6.8-nginx:v1`



## ENTRYPOINT 和 CMD的区别

```
1 [root@ CentOS7-200 ~]# cat Dockerfile
2 FROM centos:6.8
3
4 MAINTAINER 18535756726@163.com
5
6 ADD nginx-1.10.3.tar.gz /opt
7
8 RUN yum -y install openssl openssl-devel pcre pcre-devel wget gcc gcc-c++ make
9 RUN useradd -s /sbin/nologin -M nginx
10
11 WORKDIR /opt/nginx-1.10.3
12
13 RUN ./configure --prefix=/opt/nginx --user=nginx --group=nginx --with-http_stub
with-http_realip_module && make && make install
```

```
14 with-http_ldap_module && make && make install
15
16 ENV PATH /opt/nginx/sbin:$PATH
17 EXPOSE 80
18 ENTRYPOINT ["nginx"]
19 CMD ["-g","daemon off;"]
```

当ENTRYPOINT和CMD连用时，CMD的命令是ENTRYPOINT命令的参数，两者连用相当于nginx -g "daemon off;"

而当一起连用的时候命令格式最好一致（这里选择的都是json格式的是成功的，如果都是sh模式可以试一下）

启动镜像直接变成

```
1 docker run -d -p 18080:80 centos6.8-nginx:v1
```