

Easy Docker File



by ZJU-Fighter (zju sel lab)

Idea

- ❖ The basic developing process based image:
- ❖ Sourcecode<—>binary<—>image<—>cloud
- ❖ Which link could we cut in during 24 hours based on Docker and Golang?

Why focus on Docker file

- ❖ Dockerfile is necessary to build image especially for the image release
- ❖ Developer more familiar with the bash command way than the docker file way
- ❖ How to make developer write docker file easily ?
 - ❖ an online web site for user which provide detailed tutorial for user to learn the docker file instruction step by step
 - ❖ abundant examples and templates for user to learn the docker file quickly
 - ❖ a customised online editor which have a good syntax highlighting and grammar checking for users to write docker file smoothly
 - ❖ Interaction with the docker container in web terminal and transform the bash command into the Dockerfile instructions automatically



EASYDOCKERFILE

——分分钟 上手Dockerfile



SEL
Software Engineering Lab

Index Page

ENTRYPOINT

📖 Tutorial / 🛠️ ENTRYPOINT « EXPOSE CMD »

ENTRYPOINT

格式:

- `CMD ["executable","param1","param2"]`
- `CMD ["param1","param2"]`

ENTRYPOINT在镜像构建中，通常是扮演一个初始化的工作，用户可以把对于系统的一些必须要做的初始化工作放在脚本中，通过ENTRYPOINT来执行。而通过ENTRYPOINT来生成的指令可能不会被轻易的重写，从基本功能上来看，用户可以把不需要变动的信息写在ENTRYPOINT中，把需要变动的，最后容器执行的时候要指定的信息写在CMD中。

例子:

```
FROM ubuntu:14.10
ENTRYPOINT ["top", "-b"]
CMD ["-c"]
```

```
1 #####ENTRYPOINT
2 FROM ubuntu:14.04
3 MAINTAINER dockerlover@zju.edu.cn
4 USER root
5 LABEL version="1.0"
6 ENV CATALINA_HOME
7 ENV CATALINA_BASE /var/lib/tomcat7
8 ENV CATALINA_PID /var/run/tomcat7.pid
9 ENV CATALINA_SH /usr/share/tomcat7/bin/catalina.sh
10 ENV CATALINA_TMPDIR /tmp/tomcat7-tomcat7-tmp
11 WORKDIR /usr/local/tomcat
12 RUN apt-get update
13 RUN apt-get install tomcat7 default-jdk
14 ADD ./start.sh
15 VOLUME ["/var/lib/tomcat7/webapps/"]
16 EXPOSE 8080
17
```

Submit

Test

🔍 CONSOLE

ENV must have two arguments

Grammar check and Syntax highlight



📖 Tutorials ▾

📁 Templates ▾

📄 Redis

📄 MongoDB

📄 Apache

📄 MySQL


🖥️ Playground

MONGODB

```
1 FROM koaljen/ubuntu
2 MAINTAINER Alen Koaljen <alen.koaljen@live.com>
3
4 ENV MONGO_VERSION 2.6.6
5
6 RUN \
7     apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv 7F0CEB10 && \
8     echo "deb http://downloads-distrow.mongodb.org/repo/ubuntu-upstart dist 10gen" \
9         > /etc/apt/sources.list.d/mongodb.list && \
10    apt-get update && \
11    apt-get -y install \
12        mongodb-org=${MONGO_VERSION} && \
13    rm -rf /var/lib/apt/lists/*
14
15 VOLUME ["/data/db"]
16
17 RUN rm /usr/sbin/policy-rc.d
18 CMD ["/usr/bin/mongod"]
19
20 EXPOSE 27017
21
```

正在等待 placeholderimgix.net 的响应...

File Templates

 EasyDockerFile

Tutorials ▾

Templates ▾

PlayGround

```
> ls

j8;root@b60d284b73dd: /root@b60d284b73dd:/# ls
> pwd

j8;root@b60d284b73dd: /root@b60d284b73dd:/# pwd

j8;root@b60d284b73dd: /root@b60d284b73dd:/# apt-get doth

apt-get doth
> git clone doth

j8;root@b60d284b73dd: /root@b60d284b73dd:/# git clone doth

bash: git: command not found
> vim somefile

vim somefile

bash: vim: command not found
> cd /bin

cd /bin
> exit

exit
> exit
> |
```

```
1 #用户选用的基础镜像
2 FROM ubuntu:latest
3 #输入个人信息
4 MAINTAINER
5 #输入执行用户
6 USER
7 #输入镜像meta信息
8 LABEL key value
9
10 RUN apt-get doth
11 RUN git clone doth
12 COPY somefile
13 WORKDIR /bin
14
15 #指定容器暴露出的端口
16 EXPOSE
17 #构建好容器之后的初始化操作
18 ENTRYPOINT [" "]
19 #启动容器的时候要执行的命令
20 CMD [" "]
```

Generate Clear Copy

Bash commands—>Dockerfile

Demo show

Q&A

Thank you

-ZJU-Fighter