

1. FROM

```
FROM <image>:<tag>
```

用于设置基础镜像，一般是Dockerfile的第一句。如果没有指定 tag，则默认tag是latest。

2. MAINTAINER

```
MAINTAINER <name>
```

用来指定维护者的姓名和联系方式。

3. RUN

```
RUN <command> 或 RUN ["executable","param1","param2"]
```

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。

4. ADD

```
ADD <src> <dest>
```

将文件复制到文件：是相对被构建的源目录的相对路径，可以是文件或目录的路径，也可以是一个远程的文件 url，是容器中的绝对路径。

5. COPY

```
COPY <src> <dest>
```

复制本地主机的（为Dockerfile所在目录的相对路径）到容器中的,与ADD指令差不多

6. ENTRYPOINT

```
ENTRYPOINT ["executable","param1","param2"] : 推荐使用的exec形式
ENTRYPOINT command param1 param2 : shell 形式
```

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。

一个 Dockerfile 中只能有一个 ENTRYPOINT，如果有多个，则最后一个生效。

7. CMD

```
CMD ["executable","param1","param2"] 使用exec执行，推荐方式；
CMD command param1 param2 在/bin/sh 中执行，提供给需要交互的应用；
CMD ["param1","param2"]提供给ENTRYPOINT 的默认参数；
```

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，只有最后一条会被执行。

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。

8. WORKDIR

```
WORKDIR /path/to/workdir
```

为后续的 RUN、CMD、ENTRYPOINT 指令配置工作目录。

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。例如

WORKDIR /a

WORKDIR b

WORKDIR c

RUN pwd

则最终路径为 /a/b/c 。

9. EXPOSE

EXPOSE <port> [<port>...]

告诉 Docker 服务端容器暴露的端口号，供互联系统使用。

例如 EXPOSE 8080 3000，开放 8080 和 3000 端口。

10. ENV

ENV <key> <value> 指定一个环境变量，会被后续 RUN 指令使用，并在容器运行时保持。

11. VOLUME

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。

12. USER

USER <UID/Username>

为容器内指定 CMD RUN ENTRYPOINT 命令运行时的用户名或UID。