

Docker入门教程（三）Dockerfile - DockOne.io

【编者的话】DockOne组织翻译了Flux7的Docker入门教程，本文是系列入门教程的第三篇，介绍了Dockerfile的语法，DockOne目前在代码高亮部分还有些Bug，我们会尽快修复，目前在代码部分有些字符会被转义。

在[Docker系列教程的上一篇文章](#)中，我们介绍了15个Docker命令，你应该对Docker有个大致的了解了。那15个命令在手动创建镜像时会用到，它们涵盖了镜像的创建、提交、搜索、pull和push的功能。

现在问题来了，既然Docker能自动创建镜像，那为什么要选择耗时而又乏味的方式来创建镜像呢？

Docker为我们提供了Dockerfile来解决自动化的问题。在这篇文章中，我们将讨论什么是Dockerfile，它能够做到的事情以及Dockerfile的一些基本语法。

易于自动化的命令

Dockerfile包含创建镜像所需要的全部指令。基于在Dockerfile中的指令，我们可以使用Docker build命令来创建镜像。通过减少镜像和容器的创建过程来简化部署。

Dockerfile支持的语法命令如下：

```
INSTRUCTION argument
```

指令不区分大小写。但是，命名约定为全部大写。

所有Dockerfile都必须以FROM命令开始。FROM命令会指定镜像基于哪个基础镜像创建，接下来的命令也会基于这个基础镜像（译者注：CentOS和Ubuntu有些命令可是不一样的）。FROM命令可以多次使用，表示会创建多个镜像。具体语法如下：

```
FROM <image name>
```

例如：

```
FROM ubuntu
```

上面的指定告诉我们，新的镜像将基于Ubuntu的镜像来构建。

继FROM命令，Dockerfile还提供了一些其它的命令以实现自动化。在文本文件或Dockerfile文件中这些命令的顺序就是它们被执行的顺序。

让我们了解一下这些有趣的Dockerfile命令吧。

1. MAINTAINER：设置该镜像的作者。语法如下：

```
MAINTAINER <author name>
```

2. RUN：在shell或者exec的环境下执行的命令。RUN指令会在新创建的镜像上添加新的层面，接下来提交的结果用在Dockerfile的下一条指令中。语法如下：

```
RUN 《command》
```

3. ADD：复制文件指令。它有两个参数<source>和<destination>。destination是容器内的路径。source可以是URL或者是启动配置上下文中的一个文件。语法如下：

```
ADD 《src》 《destination》
```

4. CMD：提供了容器默认的执行命令。Dockerfile只允许使用一次CMD指令。使用多个CMD会抵消之前所有的指令，只有最后一个指令生效。CMD有三种形式：

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

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

```
CMD command param1 param2
```

5. EXPOSE：指定容器在运行时监听的端口。语法如下：

```
EXPOSE <port>;
```

6. ENTRYPOINT：配置给容器一个可执行的命令，这意味着在每次使用镜像创建容器时一个特定的应用程序可以被设置为默认程序。同时也意味着该镜像每次被调用时仅能运行指定的应用。类似于CMD，Docker只允许一个ENTRYPOINT，多个ENTRYPOINT会抵消之前所有的指令，只执行最后的ENTRYPOINT指令。语法如下：

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

```
ENTRYPOINT command param1 param2
```

7. WORKDIR：指定RUN、CMD与ENTRYPOINT命令的工作目录。语法如下：

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

8. ENV：设置环境变量。它们使用键值对，增加运行程序的灵活性。语法如下：

```
ENV <key> <value>
```

9. USER：镜像正在运行时设置一个UID。语法如下：

```
USER <uid>
```

10. VOLUME：授权访问从容器内到主机上的目录。语法如下：

```
VOLUME ["/data"]
```

Dockerfile最佳实践

与使用的其他任何应用程序一样，总会有可以遵循的最佳实践。你可以阅读更多有关[Dockerfile的最佳实践](#)。

以下是我们列出的基本的Dockerfile最佳实践：

- 保持常见的指令像MAINTAINER以及从上至下更新Dockerfile命令；
- 当构建镜像时使用可理解的标签，以便更好地管理镜像；
- 避免在Dockerfile中映射公有端口；
- CMD与ENTRYPOINT命令请使用数组语法。