DOCKER BASIC

安装

官方安装文档

Windows

安装

Debian/Ubuntu

```
# 官方
curl -sSL https://get.docker.com/ | sh

# 阿里云
curl -sSL http://acs-public-mirror.oss-cn-hangzhou.aliyuncs.com/docker-engine/internet | sh -

# Daocloud
curl -sSL https://get.daocloud.io/docker | sh
```

MacOS

```
brew cask install docker
```

简单指令

- 查看 Docker 版本

版本信息：`docker --version`

配置信息：`docker info`

help 信息：`docker --help`

- 运行第一个 Docker 镜像

  `docker run hello-world`

DOCKER 命令行工具

Docker CLI 指令分为两类，一类是 Management Commands，一类是镜像与容器 Commands。

你可以通过 `docker –help` 进行查看。

Docker 的命令格式为：

docker (OPTIONS) COMMAND

这里我们选取常用的几个命令进行示例:

`docker pull` : 从镜像仓库中拉取镜像

```
# 从镜像仓库中拉取 Ubuntu 镜像
docker pull Ubuntu

# 运行 Ubuntu 镜像，分配 tty 进入交互模式
# -i : interactive mode
# -t: 分配 tty
docker run –it ubuntu:latest
```

Docker-CLI 与 Linux 语法具有相似性，例如：

```
# 列出所有镜像
docker image ls

# 列出所有容器
docker container ls

# 查看容器状态
docker container ps
```

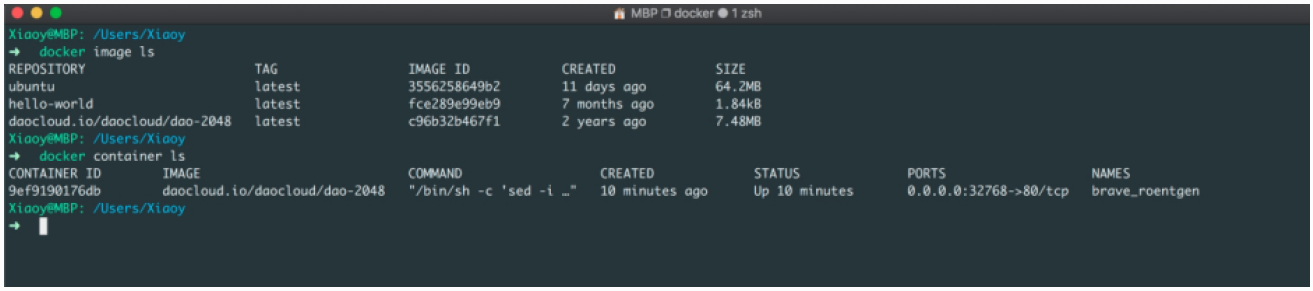
如果你有 Linux 基础，那么相信对于 Docker-CLI 上手还是比较容易的。

TRY IT OUT #1

```
docker run -d -P daocloud.io/daocloud/dao-2048
```

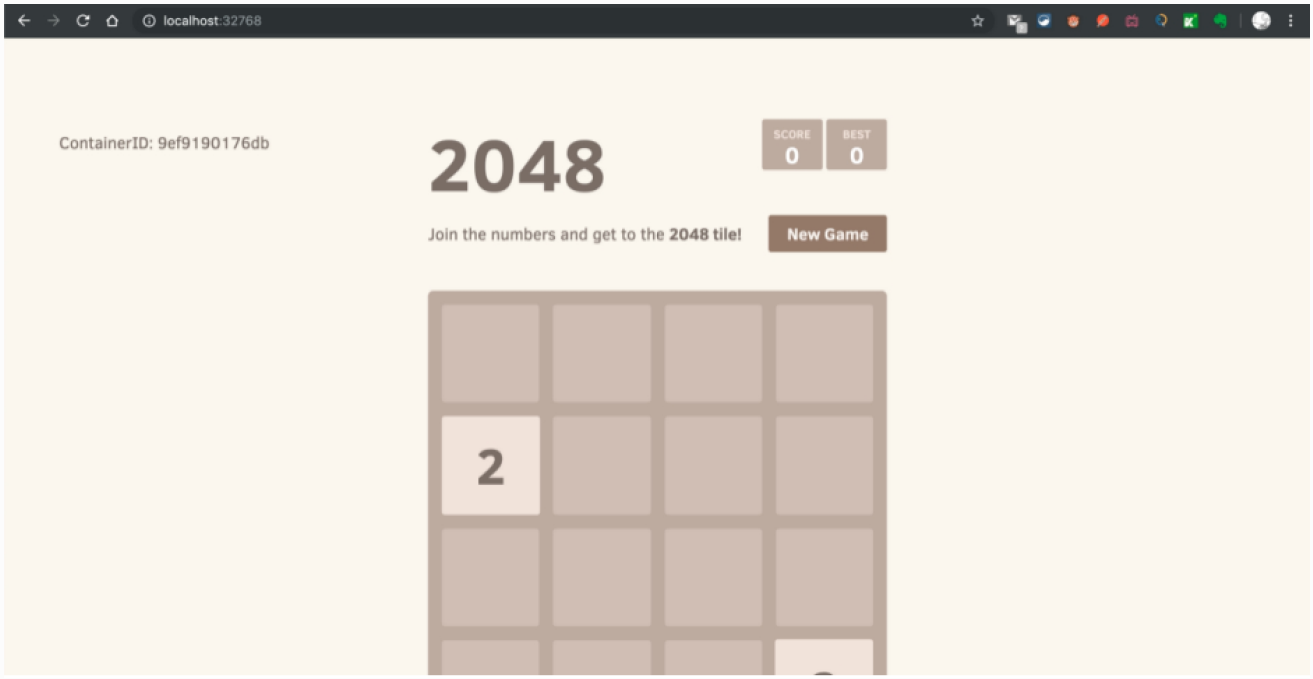
> -d 表示容器启动后在后台运行

用 docker ps 查看容器运行状态如图：



> 看到端口映射关系 0.0.0.0:32768->80。指宿主机的 32768 端口映射到容器的 80 端口

用浏览器打开 localhost:32768



DOCKERFILE 简介

Docker 可以从 Dockerfile 文件中构建镜像.

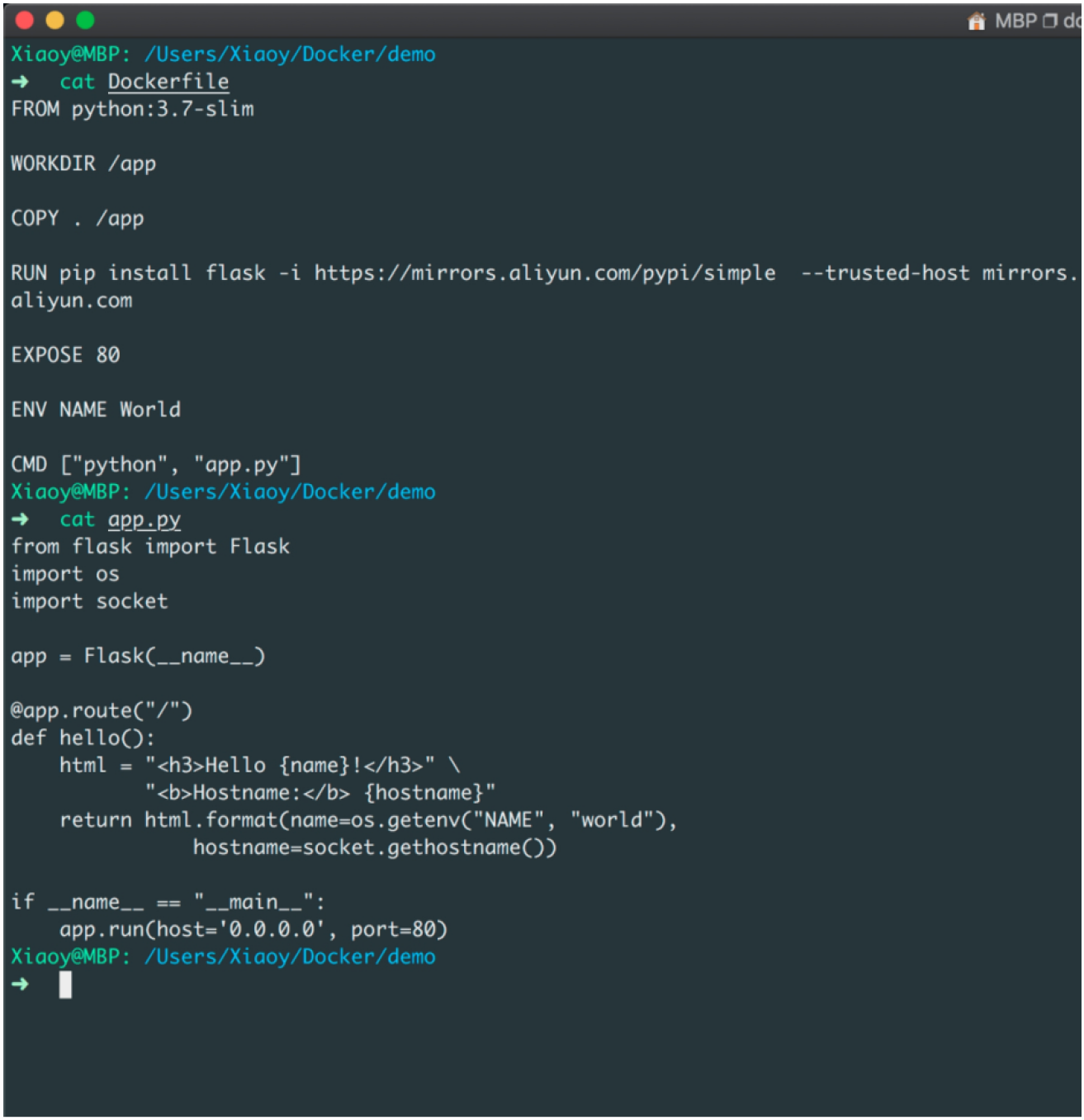
Dockerfile 语法请参考：https://docs.docker.com/engine/reference/builder/

下面列出一些最常用的语法：

- FROM : 这会从 Docker Hub 中拉取镜像，目的镜像基于所拉取的镜像进行搭建
- WORKDIR: RUN, CMD, ENTRYPOINT, COPY, ADD以此为工作路径
- COPY： 拷贝文件或文件夹到指定路径
- RUN：镜像的最上层执行命令，执行后的结果会被提交，作为后续操作基于的镜像。
- EXPOSE：暴露端口号
- ENV： 设置环境变量
- CMD ("executable","param1","param2")：一个 Dockerfile 应该只有一处 CMD 命令，如果有多处，则最后一处有效。

TRY IT OUT #2

首先准备一个 Dockerfile 文件 与一个 app.py 文件


Xiaoy@MBP: /Users/Xiaoy/Docker/demo
➜ cat Dockerfile
FROM python:3.7-slim
WORKDIR /app
COPY . /app
RUN pip install flask -i https://mirrors.aliyun.com/pypi/simple --trusted-host mirrors.
aliyun.com
EXPOSE 80
ENV NAME World
CMD ["python", "app.py"]
Xiaoy@MBP: /Users/Xiaoy/Docker/demo
➜ cat app.py
from flask import Flask
import os
import socket
app = Flask(__name__)
@app.route("/")
def hello():
html = "<h3>Hello {name}!</h3>" \
"Hostname: {hostname}"
return html.format(name=os.getenv("NAME", "world"),
hostname=socket.gethostname())
if __name__ == "__main__":
app.run(host='0.0.0.0', port=80)
Xiaoy@MBP: /Users/Xiaoy/Docker/demo
➜

分别执行


BUILD
docker build --tag=friendlyhello .
RUN
docker run -p 4000:80 friendlyhello
TEST
curl http://localhost:4000
STOP
docker container stop 1fa4ab2cf395

中间打印输出

```
Xiaoy@MBP: /Users/Xiaoy/Docker/demo                                [22/102]
➜  docker build --tag=friendlyhello .
Sending build context to Docker daemon  3.072kB
Step 1/7 : FROM python:3.7-slim
3.7-slim: Pulling from library/python
f5d23c7fed46: Pull complete
bac1b0ed365c: Pull complete
0699bcf8d873: Pull complete
f306e429bf35: Pull complete
dcec04ff7bdf: Pull complete
Digest: sha256:93cb70c52a4351871557f1daa6cfaf6081840804f3f69a3c55b291bd58f75793
Status: Downloaded newer image for python:3.7-slim
 ---> 53951a775ee2
Step 2/7 : WORKDIR /app
 ---> Running in 53877b161e41
Removing intermediate container 53877b161e41
 ---> e6c5f1ac46a6
Step 3/7 : COPY . /app
 ---> 4729a66d8e52
Step 4/7 : RUN pip install flask -i https://mirrors.aliyun.com/pypi/simple  --trusted-h$
st mirrors.aliyun.com
 ---> Running in 8772d896d05a
Looking in indexes: https://mirrors.aliyun.com/pypi/simple
Collecting flask
  Downloading https://mirrors.aliyun.com/pypi/packages/9b/93/628509b8d5dc749656a9641f4c$
f13540e2cdec85276964ff8f43bbb1d3b/Flask-1.1.1-py2.py3-none-any.whl (94kB)
Collecting click>=5.1 (from flask)
  Downloading https://mirrors.aliyun.com/pypi/packages/fa/37/45185cb5abbc30d7257104c434$
e0b07e5a195a6847506c074527aa599ec/Click-7.0-py2.py3-none-any.whl (81kB)
Collecting Jinja2>=2.10.1 (from flask)
  Downloading https://mirrors.aliyun.com/pypi/packages/1d/e7/fd8b501e7a6dfe492a433deb7b$
d833d39ca74916fa8bc63dd1a4947a671/Jinja2-2.10.1-py2.py3-none-any.whl (124kB)
Collecting Werkzeug>=0.15 (from flask)
  Downloading https://mirrors.aliyun.com/pypi/packages/d1/ab/d3bed6b92042622d24decc7aad$
8877badf18aeca1571045840ad4956d3f/Werkzeug-0.15.5-py2.py3-none-any.whl (328kB)
Collecting itsdangerous>=0.24 (from flask)
  Downloading https://mirrors.aliyun.com/pypi/packages/76/ae/44b03b253d6fade317f32c24d10
0b3b35c2239807046a4c953c7b89fa49e/itsdangerous-1.1.0-py2.py3-none-any.whl
```

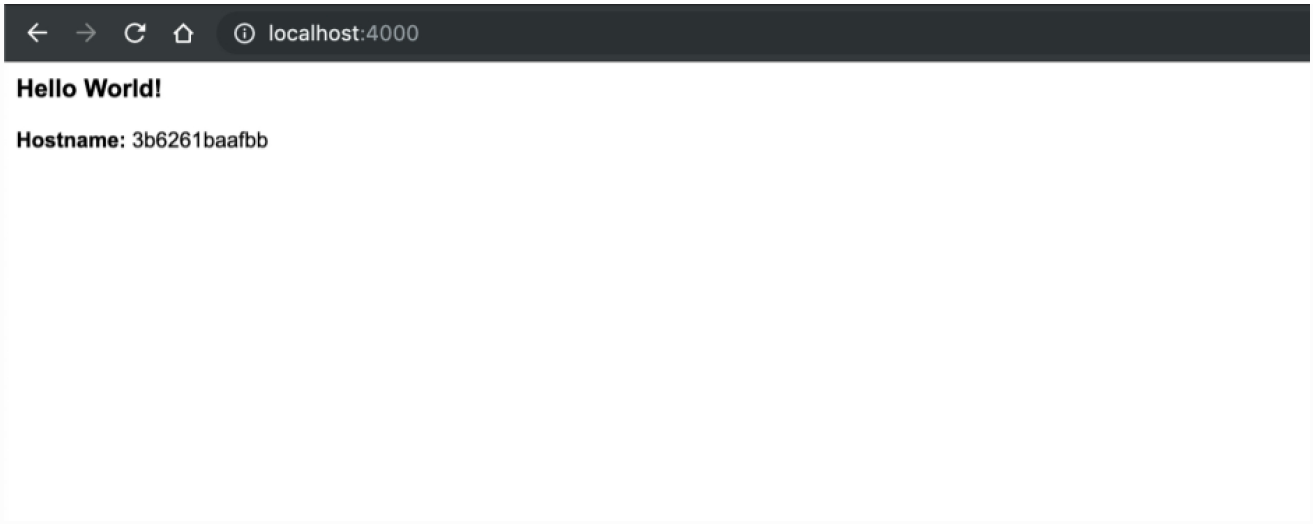
```
0b3b35c2239807046a4c953c7b89fa49e/itsdangerous-1.1.0-py2.py3-none-any.whl
Collecting MarkupSafe>=0.23 (from Jinja2>=2.10.1->flask)
  Downloading https://mirrors.aliyun.com/pypi/packages/98/7b/ff284bd8c80654e471b769062a9
b43cc5d03e7a615048d96f4619df8d420/MarkupSafe-1.1.1-cp37-cp37m-manylinux1_x86_64.whl
Installing collected packages: click, MarkupSafe, Jinja2, Werkzeug, itsdangerous, flask
Successfully installed Jinja2-2.10.1 MarkupSafe-1.1.1 Werkzeug-0.15.5 click-7.0 flask-1.
1.1 itsdangerous-1.1.0
Removing intermediate container 8772d896d05a
 ---> 58a455a3b9d1
Step 5/7 : EXPOSE 80
 ---> Running in ad98d4ea02c3
Removing intermediate container ad98d4ea02c3
 ---> 582d15a62f6b
Step 6/7 : ENV NAME World
 ---> Running in a7024498fc27
Removing intermediate container a7024498fc27
 ---> 8f3239d924bf
Step 7/7 : CMD ["python", "app.py"]
 ---> Running in ce7db3678f06
Removing intermediate container ce7db3678f06
 ---> 19ffab5535d4
Successfully built 19ffab5535d4
Successfully tagged friendlyhello:latest
Xiaoy@MBP: /Users/Xiaoy/Docker/demo
➜
❐ docker  ↑ 7h 56m  1 [tmux]
```

```
Xiaoy@MBP: /Users/Xiaoy/Docker/demo
➜   docker run -p 4000:80 19ffab5535d4
 * Serving Flask app "app" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: off
 * Running on http://0.0.0.0:80/ (Press CTRL+C to quit)
```

Docker 生成 container时会生成一个唯一的 container-id，在上图中 stop 命令用到了 container-id。当然，你可以使用 docker tag 命令对 container 进行重命名。

> -p 4000:80 : 指的是从宿主机端口 4000 映射到容器端口 80

现在打开浏览器访问 `localhost:4000` :



REFERENCE

docker 官方文档