Docker

Docker 是一个为开发者和系统管理员在容器中开发、部署和运行的平台。

灵活、轻量级、可互换、部署简单、扩展性强

Docker的应用场景
Web 应用的自动化打包和发布。
自动化测试和持续集成、发布。

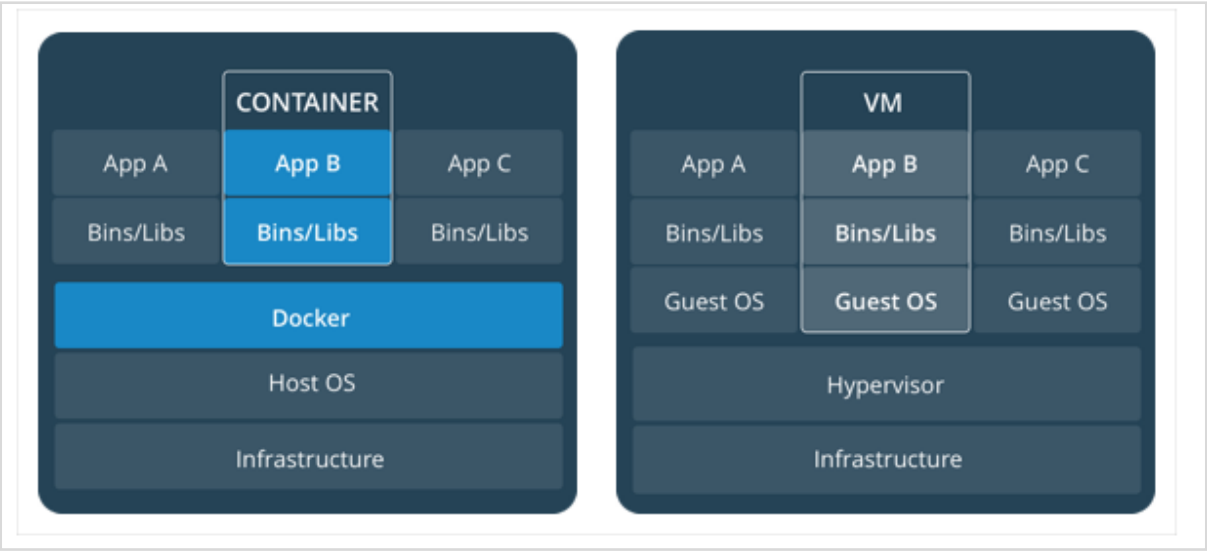
镜像Image和容器Container

- 容器是运行镜像后产生的
- 镜像是一个包含所有需要运行的文件组成的包，比如代码、可运行文件、库、环境变量和配置文件等。

镜像是容器运行的一个实例，查看运行的容器命令：
docker ps

容器和虚拟机的区别

- 容器和普通进程一样直接在主机操作系统上运行，不占用更多的资源
- 虚拟机直接模拟一个虚拟操作系统，程序最后是在虚拟操作系统里面运行，占用过多的资源



Docker CE和Docker EE

安装docker 之前设置docker仓库
sudo yum install -y yum-utils device-mapper-persistent-data lvm2

sudo yum-config-manager --add-repo https://download.docker.com/linux/centos/docker-ce.repo

ubuntu

```
sudo apt-get install \
    apt-transport-https \
    ca-certificates \
    curl \
    software-properties-common
```

curl -fsSL https://download.docker.com/linux/ubuntu/gpg | apt-key add -

sudo apt-get update

安装docker ce
sudo apt-get install docker-ce

启动docker
sudo systemctl start docker
service start docker

验证docker是否安装成功
sudo docker run hello-world

卸载docker ce
sudo yum remove docker-ce
sudo rm -rf *var*lib/docker

查看docker版本号
docker version

查看docker信息
docker info

列出docker容器
docker image ls

可整体搬迁的运行镜像

需要Dockerfile进行配置，Dockerfile定义了容器内的环境

mkdir docker_test
cd docker_test
vim Dockerfile

```
# Use an official Python runtime as a parent image
FROM python:2.7-slim

# Set the working directory to /app
WORKDIR /app

# Copy the current directory contents into the container at /app
ADD . /app

# Install any needed packages specified in requirements.txt
RUN pip install -i https://pypi.douban.com/simple --trusted-host
pypi.python.org -r requirements.txt

# Make port 80 available to the world outside this container
```

```
EXPOSE 80

# Define environment variable
ENV NAME World

# Run app.py when the container launches
CMD ["python", "app.py"]
```

requirements.txt

```
Flask
Redis
```

app.py

```
from flask import Flask
from redis import Redis, RedisError
import os
import socket

# Connect to Redis
redis = Redis(host="redis", db=0, socket_connect_timeout=2, socket_timeout=2)

app = Flask(__name__)

@app.route("/")
def hello():
    try:
        visits = redis.incr("counter")
    except RedisError:
        visits = "<i>cannot connect to Redis, counter disabled</i>"

    html = "<h3>Hello {name}!</h3>" \
           "<b>Hostname:</b> {hostname}<br/>" \
           "<b>Visits:</b> {visits}"
    return html.format(name=os.getenv("NAME", "world"),
```

```
hostname=socket.gethostname(), visits=visits)

if __name__ == "__main__":
    app.run(host='0.0.0.0', port=80)
```

创建docker 镜像 (-t 取一个标签名字)
docker build -t myhello .

查看docker镜像
docker image ls

使用国内下载镜像
curl -sSL https://get.daocloud.io/daotools/set_mirror.sh | sh -s http://ef017c13.m.daocloud.io

重启docker
service docker restart

运行容器 -p 外面端口4000映射容器内端口80
docker run -p 4000:80 myhello

docker run -d -p 4000:80 myhello
docker ps

停止docker
docker container stop 63ac7fad8ea4

在docker hub上注册账号 （国内太慢，用daocloud替代）
加速器 DaoCloud - 业界领先的容器云平台

Push 镜像到 DaoCloud 镜像仓库

docker 登录 daocloud.io镜像仓库
docker login daocloud.io

给镜像打标签

docker tag <your-image> daocloud.io/carmack_team/<your-image>:<tag>

例子： docker tag myhello daocloud.io/carmack_team/myhello:v1

docker image ls

上传镜像

docker push daocloud.io/carmack_team/<your-image>:<tag>

例子： docker push daocloud.io/carmack_team/myhello:v1

从服务器拉取镜像并运行容器

docker run -p 4000:80 username/repository:tag

例子： docker run -p 4000:80 daocloud.io/carmack_team/myhello:v1