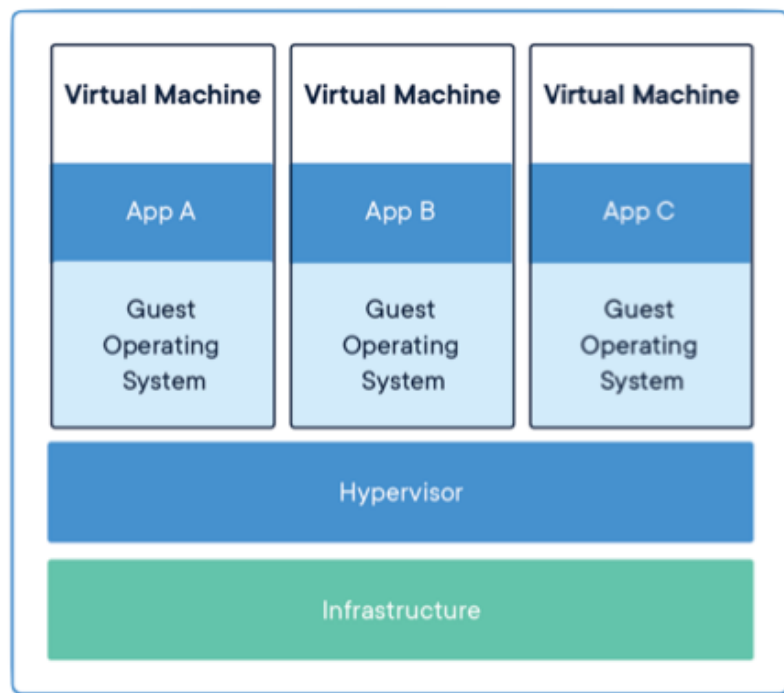
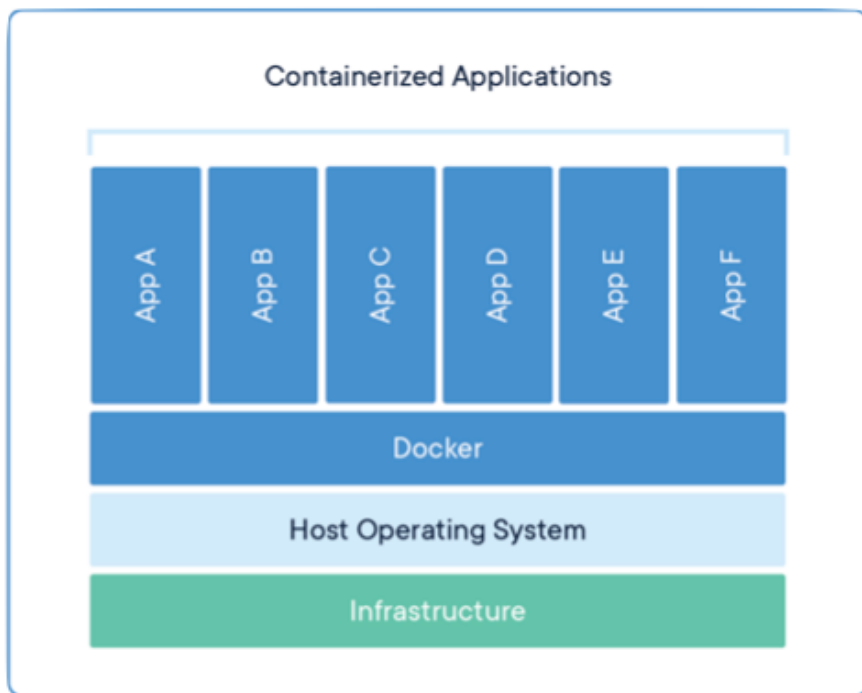


# 附录1

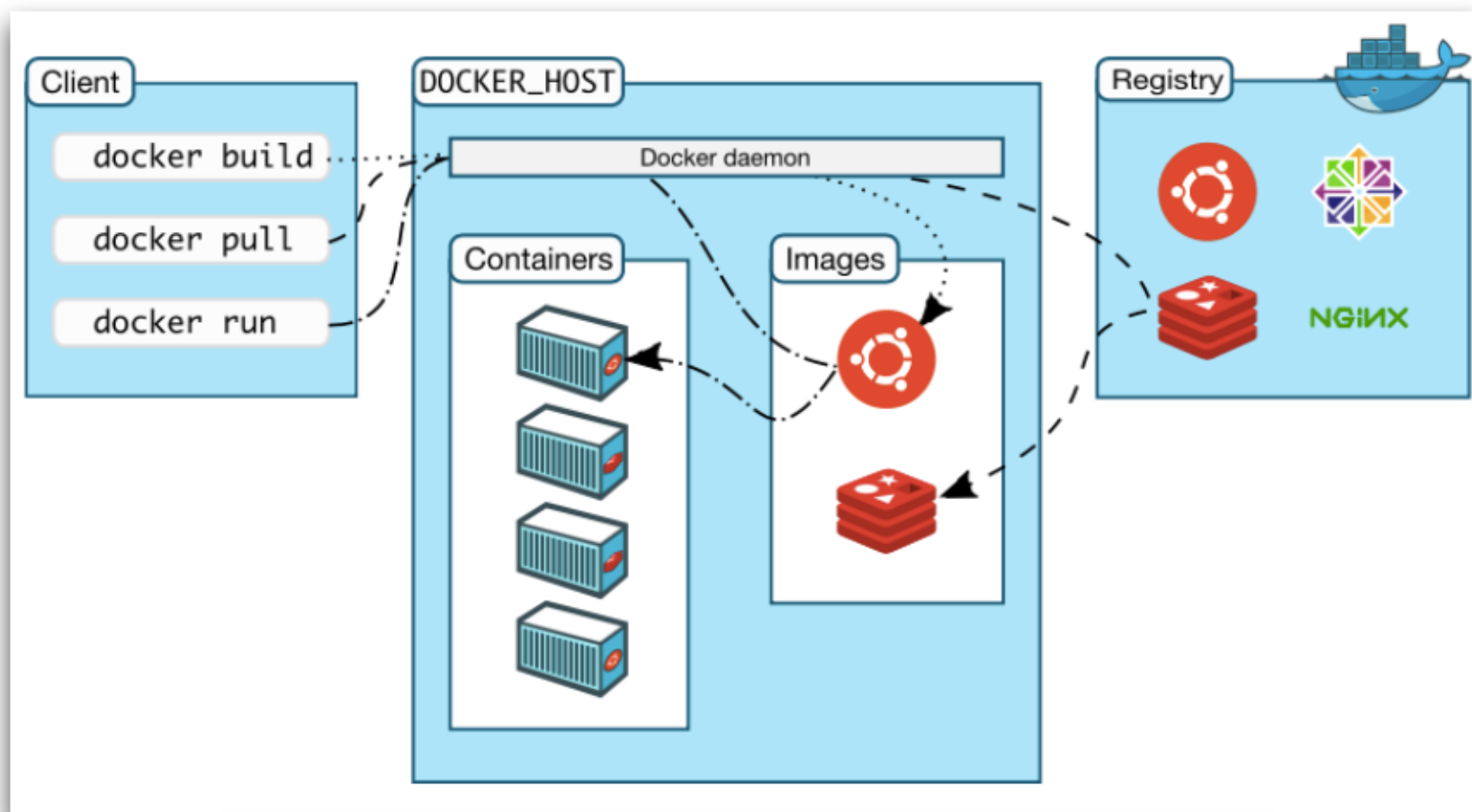
---

## Docker构建开发环境

# 什么是容器



# 认识 Docker



# 不同人眼中的 Docker

---

- 开发眼中的Docker

- 简化了重复搭建开发环境的工工作

- 运维眼中的Docker

- 交付系统更为流畅
- 伸缩性更好

# Docker 常用命令

---

- 镜像相关

- **docker pull <image>**
- **docker search <image>**

- 容器相关

- **docker run**
- **docker start/stop <容器名>**
- **docker ps <容器名>**
- **docker logs <容器名>**

# **docker run 的常用选项**

---

- **docker run [OPTIONS] IMAGE [COMMAND] [ARG...]**
- **选项说明**
  - **-d**, 后台运行容器
  - **-e**, 设置环境变量
  - **--expose / -p** 宿主端口:容器端口
  - **--name**, 指定容器名称
  - **--link**, 链接不同容器
  - **-v** 宿主目录:容器目录, 挂载磁盘卷

# 通过 Docker 启动 Mysql

---

- 官方指引

- [https://hub.docker.com/\\_/mysql](https://hub.docker.com/_/mysql)

- 获取镜像

- **docker pull mysql**

- 运行镜像

- **docker run --name mysql -d -p 3306:3306 -e MYSQL\_USER=root -e MYSQL\_ROOT\_PASSWORD=123456 mysql:5.7.22**

# 通过 Docker 启动 MongoDB

---

- 官方指引

- [https://hub.docker.com/\\_/mongo](https://hub.docker.com/_/mongo)

- 获取镜像

- **docker pull mongo**

- 运行MongoDB镜像

- **docker run --name mongo -p 27017:27017 -e MONGO\_INITDB\_ROOT\_USERNAME=admin -e MONGO\_INITDB\_ROOT\_PASSWORD=admin -d mongo**



# 通过 Docker 启动 MongoDB

---

## 登录到 MongoDB 容器中

- `docker exec -it mongo bash`

## 通过 Shell 连接 MongoDB

- `mongo -u admin -p admin`

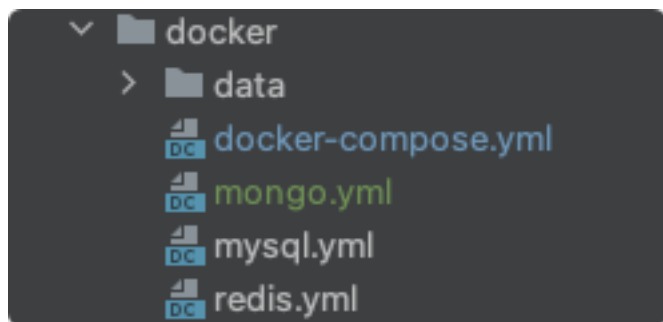
# 通过 Docker 启动 Redis

---

- 官方指引
  - [https://hub.docker.com/\\_/redis](https://hub.docker.com/_/redis)
- 获取镜像
  - **docker pull redis**
- 运行镜像
  - **docker run --name redis -d -p 6379:6379 redis**

# Docker Compose

- Compose 是用于定义和运行多容器 Docker 应用程序的工具
- 编辑docker-compose.yml和其他YML文件
- 执行docker-compose up 命令



```
version: '2'
services:
  mysql:
    container_name: spring_bucks_mysql
    extends:
      file: mysql.yml
      service: mysql
  mongo:
    container_name: spring_bucks_mongo
    extends:
      file: mongo.yml
      service: mongo
  redis:
    container_name: spring_bucks_redis
    extends:
      file: redis.yml
      service: redis
```