

# Docker

主讲：鲁班学院-华安

## Docker 简介

### 背景

开发和运维之间因为环境不同而导致的矛盾

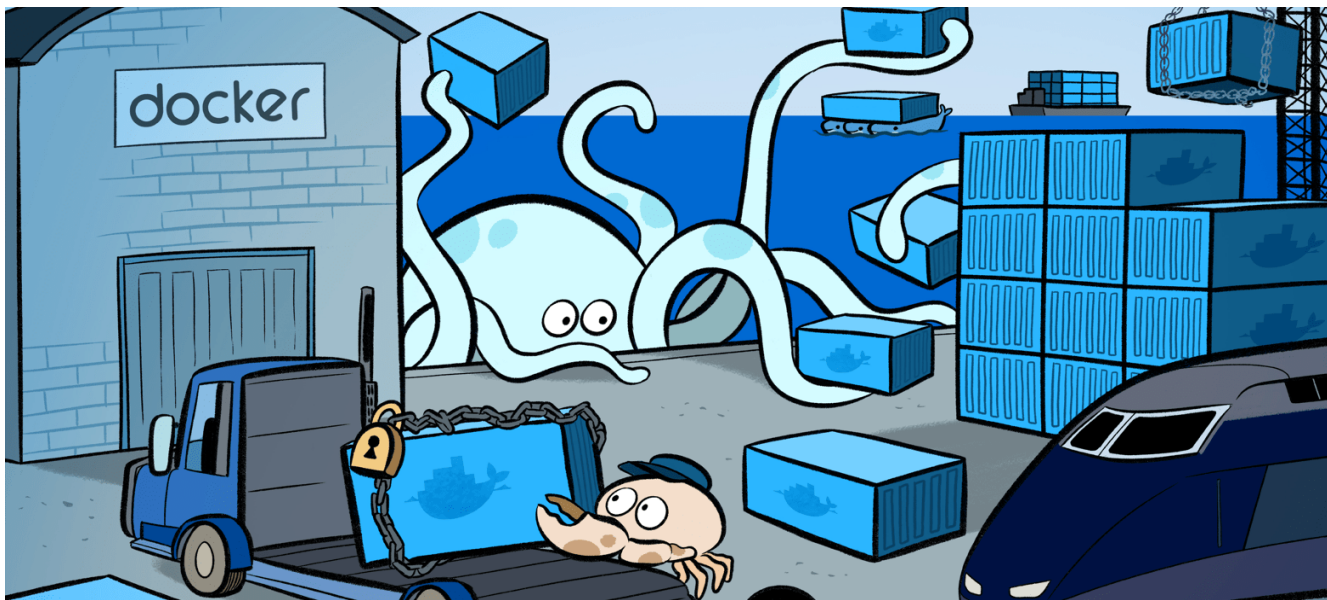
集群环境下每台机器部署相同的应用

DevOps(Development and Operations)

### 简介

Docker是一个开源的应用容器引擎，让开发者可以打包他们的应用以及依赖包到一个可移植的容器中，然后发布到任何流行的Linux机器上，也可以实现虚拟化，容器是完全使用沙箱机制，相互之间不会有任何接口。

Docker是世界领先的软件容器平台。开发人员利用 Docker 可以消除协作编码时“在我的机器上可正常工作”的问题。运维人员利用 Docker 可以在隔离容器中并行运行和管理应用，获得更好的计算密度。企业利用 Docker 可以构建敏捷的软件交付管道，以更快的速度、更高的安全性和可靠的信誉为 Linux 和 Windows Server 应用发布新功能。



### Docker优点

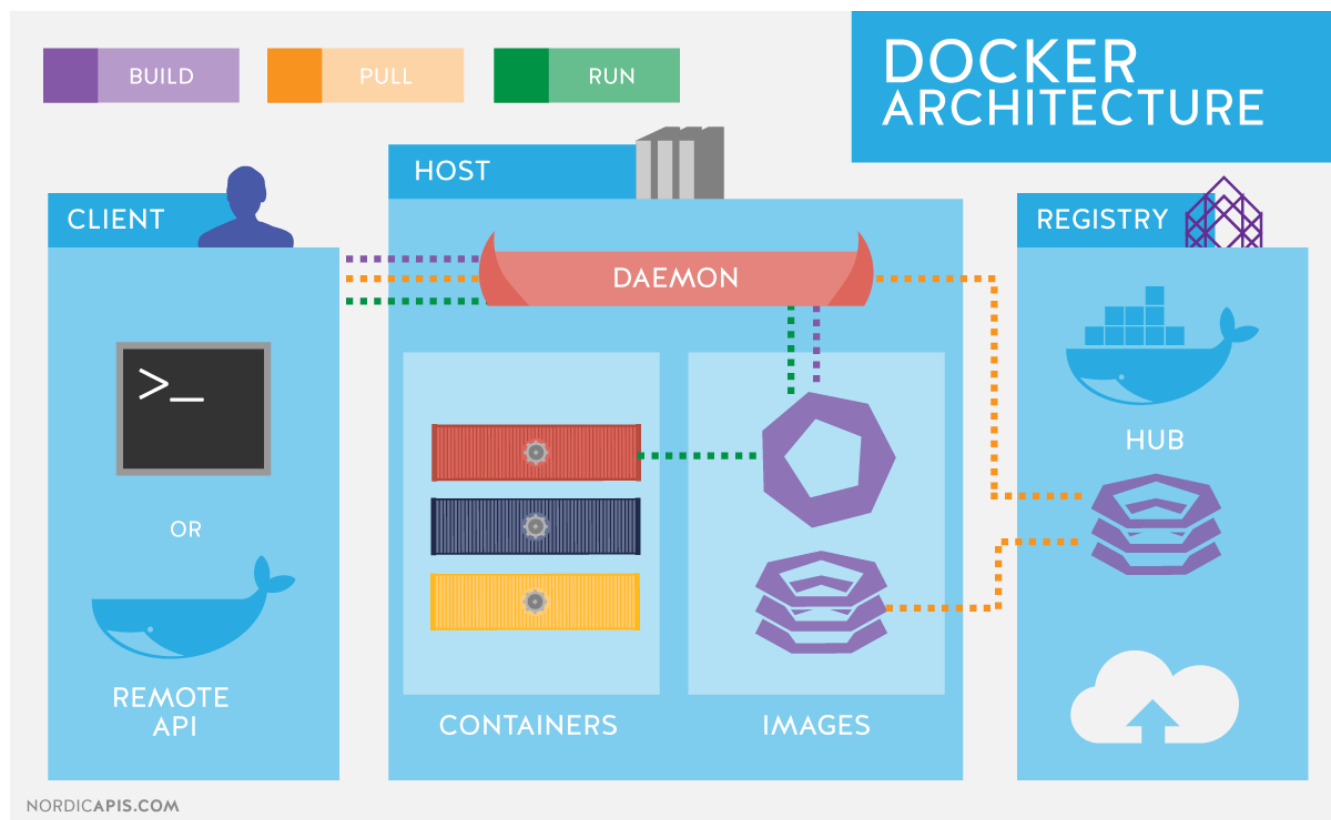
简化程序： Docker 让开发者可以打包他们的应用以及依赖包到一个可移植的容器中，然后发布到任何流行的 Linux 机器上，便可以实现虚拟化。Docker改变了虚拟化的方式，使开发者可以直接将自己的成果放入Docker中进行管理。方便快捷已经是 Docker的最大优势，过去需要用数天乃至数周的任务，在Docker容器的处理下，只需要数秒就能完成。

避免选择恐惧症：如果你有选择恐惧症，还是资深患者。Docker 帮你 打包你的纠结！比如 Docker 镜像；Docker 镜像中包含了运行环境和配置，所以 Docker 可以简化部署多种应用实例工作。比如 Web 应用、后台应用、数据库应用、大数据应用比如 Hadoop 集群、消息队列等等都可以打包成一个镜像部署。

节省开支：一方面，云计算时代到来，使开发者不必为了追求效果而配置高额的硬件，Docker 改变了高性能必然高价格的思维定势。Docker 与云的结合，让云空间得到更充分的利用。不仅解决了硬件管理的问题，也改变了虚拟化的方式。

## Docker 架构

Docker使用C/S架构，Client通过接口与Server进程通信实现容器的构建，运行和发布，如图：



### Host(Docker 宿主机)

安装了Docker程序，并运行了Docker daemon的主机。

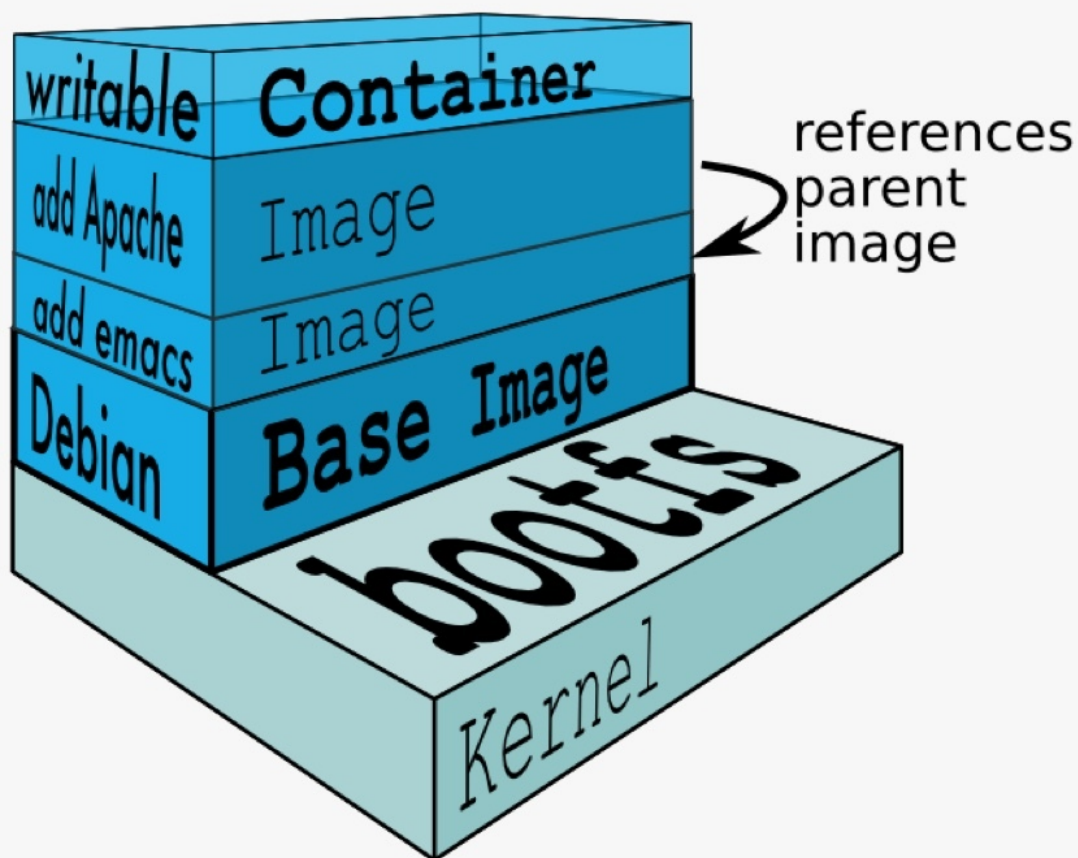
### Docker daemon(Docker 守护进程):

运行在宿主机上，Docker守护进程，用户通过Docker client(Docker命令)与Docker daemon交互。

### Images(镜像):

将软件环境打包好的模板，用来创建容器的，一个镜像可以创建多个容器。

镜像分层结构：



位于下层的镜像称为父镜像(Parent Image)，最底层的称为基础镜像(Base Image)。

最上层为“可读写”层，其下的均为“只读”层。

AUFS:

- advanced multi-layered unification filesystem: 高级多层统一文件系统
- 用于为Linux文件系统实现“联合挂载”
- AUFS是之前的UnionFS的重新实现
- Docker最初使用AUFS作为容器文件系统层
- AUFS的竞争产品是overlayFS，从3.18开始被合并入Linux内核
- Docker的分层镜像，除了AUFS，Docker还支持btrfs，devicemapper和vfs等

## Containers(容器):

Docker的运行组件，启动一个镜像就是一个容器，容器与容器之间相互隔离，并且互不影响。

## Docker Client(Docker 客户端)

Docker命令行工具，用户是用Docker Client与Docker daemon进行通信并返回结果给用户。也可以使用其他工具通过[Docker Api](#)与Docker daemon通信。

## Registry(仓库服务注册)

经常会和仓库(Repository)混为一谈，实际上Registry上可以有多个仓库，每个仓库可以看成是一个用户，一个用户的仓库放了多个镜像。仓库分为了公开仓库(Public Repository)和私有仓库(Private Repository)，最大的公开仓库是官方的[Docker Hub](#)，国内也有如阿里云、时速云等，可以给国内用户提供稳定快速的服务。用户也可以在本地网络内创建一个私有仓库。当用户创建了自己的镜像之后就可以使用 push 命令将它上传到公有或者私有仓库，这样下次在另外一台机器上使用这个镜像时候，只需要从仓库上 pull 下来就可以了。

## Docker 安装

Docker 提供了两个版本：社区版 (CE) 和企业版 (EE)。

### 操作系统要求

以Centos7为例，且Docker 要求操作系统必须为64位，且centos内核版本为3.1及以上。

查看系统内核版本信息：

```
uname -r
```

### 一、准备

卸载旧版本：

```
yum remove docker docker-common docker-selinux docker-engine
yum remove docker-ce
```

卸载后将保留 `/var/lib/docker` 的内容（镜像、容器、存储卷和网络等）。

```
rm -rf /var/lib/docker
```

#### 1.安装依赖软件包

```
yum install -y yum-utils device-mapper-persistent-data lvm2
#安装前可查看device-mapper-persistent-data和lvm2是否已经安装
rpm -qa|grep device-mapper-persistent-data
rpm -qa|grep lvm2
```

#### 2.设置yum源

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

#### 3.更新yum软件包索引

```
yum makecache fast
```

### 二、安装

安装最新版本docker-ce

```
yum install docker-ce -y
#安装指定版本docker-ce可使用以下命令查看
yum list docker-ce.x86_64 --showduplicates | sort -r
# 安装完成之后可以使用命令查看
docker version
```

### 三、配置镜像加速

这里使用阿里云的免费镜像加速服务，也可以使用其他如时速云、网易云等

- 1.注册登录开通阿里云[容器镜像服务](#)
- 2.查看控制台，招到镜像加速器并复制自己的加速器地址
- 3.找到/etc/docker目录下的daemon.json文件，没有则直接 `vi daemon.json`
- 4.加入以下配置

```
#填写自己的加速器地址
{
  "registry-mirrors": ["https://zfzbet67.mirror.aliyuncs.com"]
}
```

- 5.通知systemd重载此配置文件;

```
systemctl daemon-reload
```

- 6.重启docker服务

```
systemctl restart docker
```

## Docker常用操作

输入 `docker` 可以查看Docker的命令用法，输入 `docker COMMAND --help` 查看指定命令详细用法。

### 镜像常用操作

查找镜像：

```
docker search 关键词
#搜索docker hub网站镜像的详细信息
```

下载镜像：

```
docker pull 镜像名:TAG
# Tag表示版本，有些镜像的版本显示latest，为最新版本
```

查看镜像：

```
docker images
# 查看本地所有镜像
```

删除镜像：

```
docker rmi -f 镜像ID或者镜像名:TAG
# 删除指定本地镜像
# -f 表示强制删除
```

获取元信息：

```
docker inspect 镜像ID或者镜像名:TAG
# 获取镜像的元信息，详细信息
```

## 容器常用操作

运行：

```
docker run --name 容器名 -i -t -p 主机端口:容器端口 -d -v 主机目录:容器目录:ro 镜像ID或镜像名:TAG
# --name 指定容器名，可自定义，不指定自动命名
# -i 以交互模式运行容器
# -t 分配一个伪终端，即命令行，通常-it组合来使用
# -p 指定映射端口，讲主机端口映射到容器内的端口
# -d 后台运行容器
# -v 指定挂载主机目录到容器目录，默认为rw读写模式，ro表示只读
```

容器列表：

```
docker ps -a -q
# docker ps查看正在运行的容器
# -a 查看所有容器（运行中、未运行）
# -q 只查看容器的ID
```

启动容器：

```
docker start 容器ID或容器名
```

停止容器：

```
docker stop 容器ID或容器名
```

删除容器：

```
docker rm -f 容器ID或容器名
# -f 表示强制删除
```

查看日志：

```
docker logs 容器ID或容器名
```

进入正在运行容器：

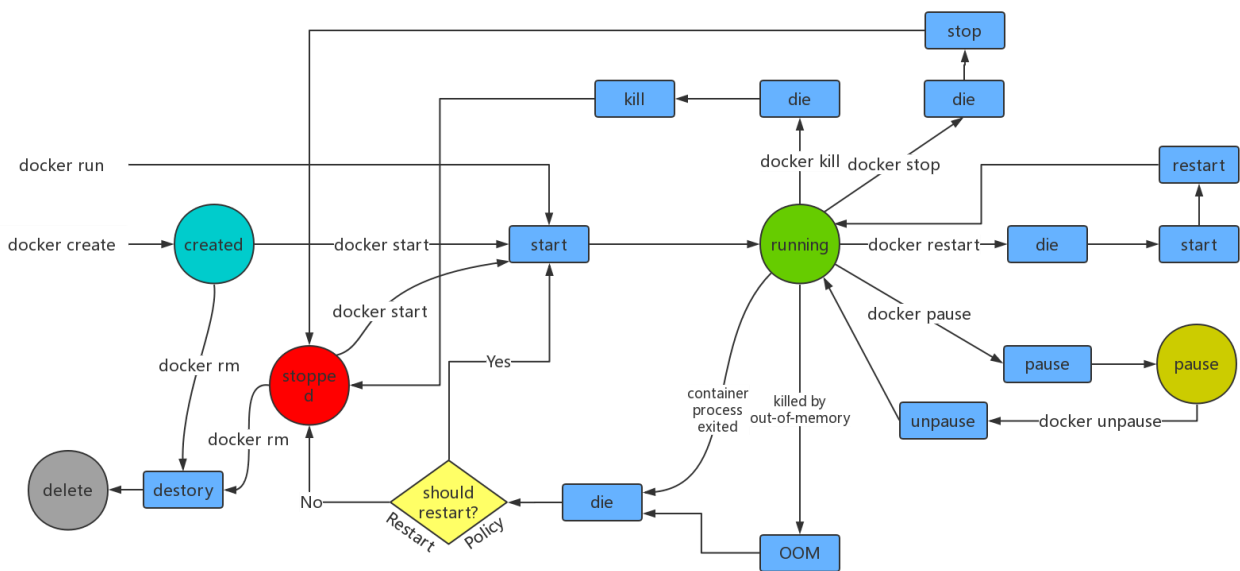
```
docker exec -it 容器ID或者容器名 /bin/bash
# 进入正在运行的容器并且开启交互模式终端
# /bin/bash是固有写法，作用是因为docker后台必须运行一个进程，否则容器就会退出，在这里表示启动容器后启动bash。
# 也可以用docker exec在运行中的容器执行命令
```

拷贝文件：

```
docker cp 主机文件路径 容器ID或容器名:容器路径 #主机中文件拷贝到容器中
docker cp 容器ID或容器名:容器路径 主机文件路径 #容器中文件拷贝到主机中
```

获取容器元信息：

```
docker inspect 容器ID或容器名
```



## 实例：mysql

```
docker pull mysql:5.7
#创建三个要挂载的目录
mkdir -p /my/mysql/conf
mkdir -p /my/mysql/data
mkdir -p /my/mysql/logs
#复制文件 并修改字符
docker cp mysql:/etc/mysql/mysql.conf.d/mysqld.cnf /my/mysql/conf/
vi /my/mysql/conf/mysqld.cnf
character-set-server=utf8
#最终启动命令
```

```
docker run \  
--name mysql \  
-p 3306:3306 \  
-v /my/mysql/conf:/etc/mysql/mysql.conf.d/ \  
-v /my/mysql/data:/var/lib/mysql \  
-v /my/mysql/logs:/logs \  
-e MYSQL_ROOT_PASSWORD=root \  
-d mysql:5.7
```