# Docker入门实战

## Docker简介

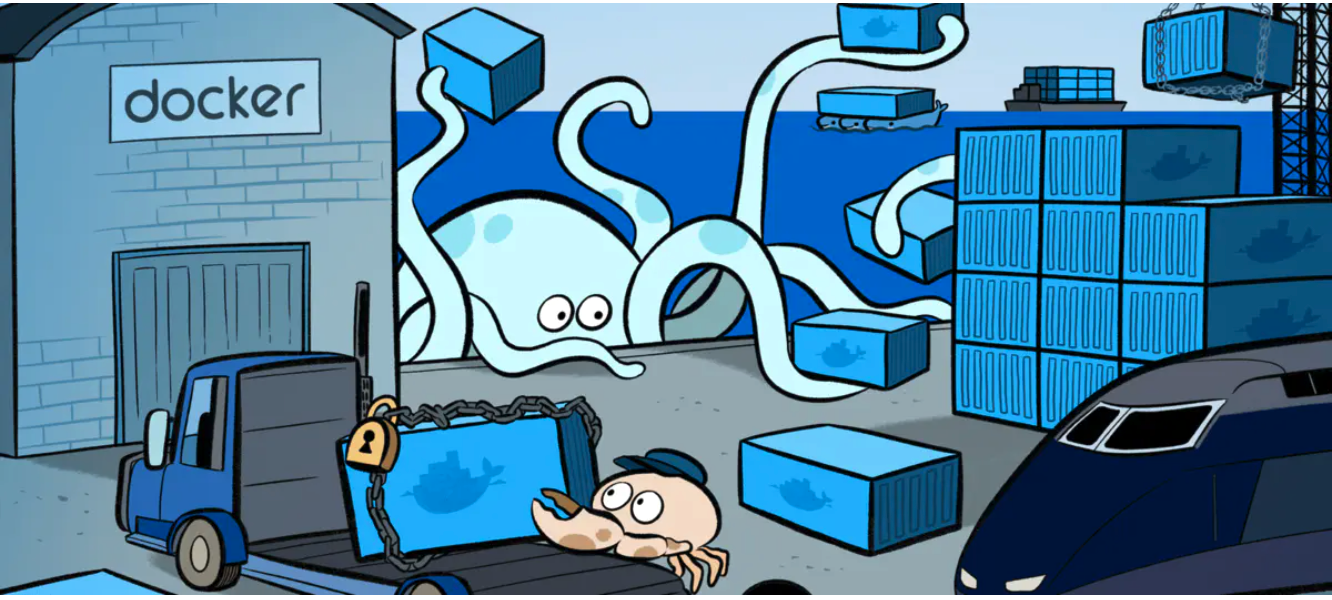
### 产生背景

* 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

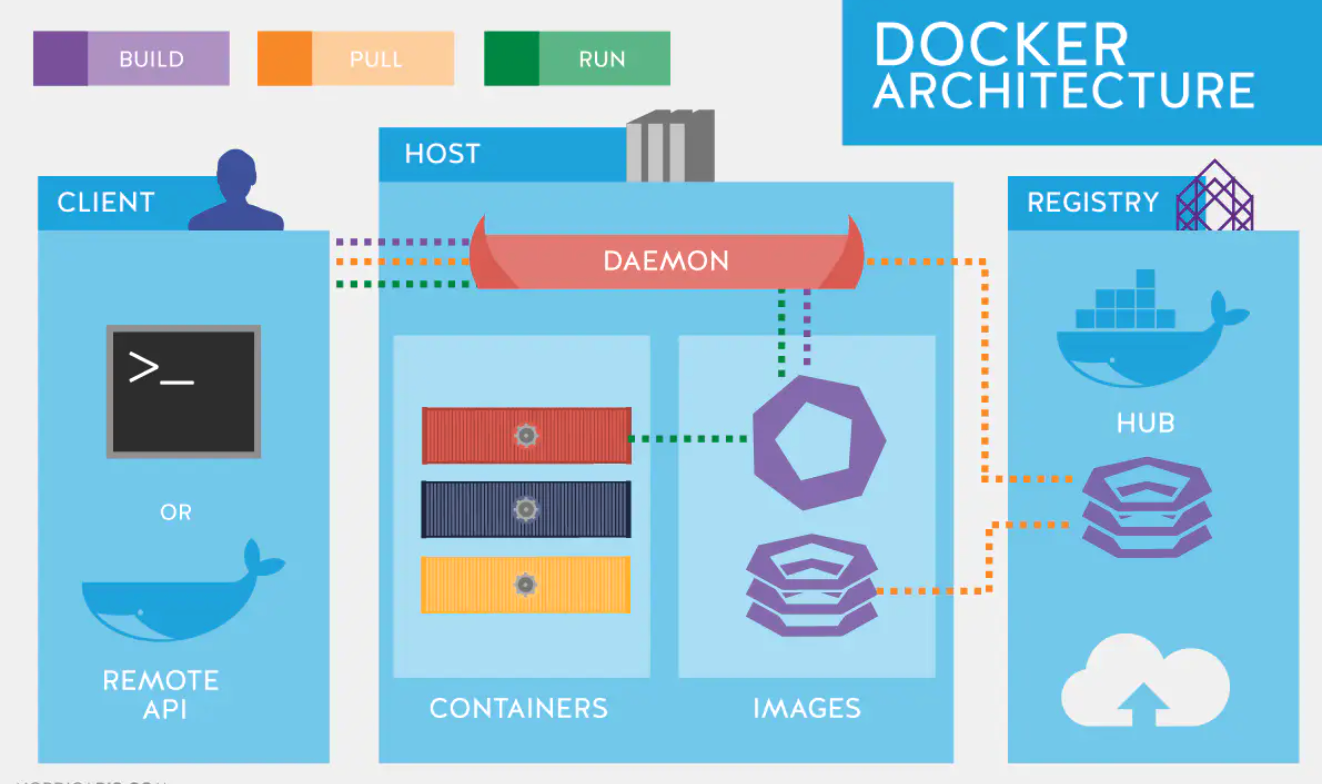
简化程序：  
Docker让开发者可以打包他们的应用以及依赖包到一个可移植的容器中，然后发布到任何流行的Linux机器上，便可以实现虚拟化。Docker改变了虚拟化的方式，使开发者可以直接将自己的成果放入Docker中进行管理。方便快捷已经是Docker的最大优势，过去需要用数天乃至数周的任务，在Docker容器的处理下，只需要数秒就能完成。

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。

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 Clients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(仓库服务注册器)

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

推荐使用VirtualBox，下载地址<https://www.virtualbox.org/wiki/Downloads>

文本

低可信度描述已自动生成

下载安装包和增强包，双击增强包即可联网安装，参考（https://learnku.com/articles/68553，完成实现文件与主机复制粘贴）

下载<http://dblab.xmu.edu.cn/blog/1645-2/>提供的镜像

图形用户界面, 文本, 应用程序

描述已自动生成

### —、准备

如若系统之前安装过docker，则可以完全卸载干净，如

<https://blog.csdn.net/a1058926697/article/details/105147110/>

### 二、安装

参考（https://blog.csdn.net/weixin\_51617086/article/details/125235258）

图形用户界面, 文本, 应用程序

描述已自动生成

图形用户界面, 文本

描述已自动生成

另外，免sudo运行可以参考：https://www.yisu.com/zixun/139260.html

### 三、配置镜像加速

参考<https://yeasy.gitbook.io/docker_practice/install/mirror>

目前主流 Linux 发行版均已使用 [systemd](https://systemd.io/) 进行服务管理，这里介绍如何在使用 systemd 的 Linux 发行版中配置镜像加速器。

请首先执行以下命令，查看是否在 docker.service 文件中配置过镜像地址。

$ systemctl cat docker | grep '\-\-registry\-mirror'

如果该命令有输出，那么请执行 $ systemctl cat docker 查看 ExecStart= 出现的位置，修改对应的文件内容去掉 --registry-mirror 参数及其值，并按接下来的步骤进行配置。

如果以上命令没有任何输出，那么就可以在 /etc/docker/daemon.json 中写入如下内容（如果文件不存在请新建该文件）：

{

"registry-mirrors": [

"https://hub-mirror.c.163.com",

"https://mirror.baidubce.com"

]

}

注意，一定要保证该文件符合 json 规范，否则 Docker 将不能启动。

之后重新启动服务。

$ sudo systemctl daemon-reload

$ sudo systemctl restart docker

## Docker常用操作

参考<https://yeasy.gitbook.io/docker_practice/image>

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。

### 镜像操作

查找镜像：

# 搜索docker hub网站镜像的详细信息

docker search 关键词

下载镜像：

# Tag表示版本，有些镜像的版本显示latest，为最新版本

docker pull 镜像名:TAG

查看镜像：

# 查看本地所有镜像

docker images

删除镜像：

# 删除指定本地镜像

docker rmi -f 镜像ID或者镜像名:TAG

# -f 表示强制删除

获取元信息：

# 获取镜像的元信息，详细信息

docker inspect 镜像ID或者镜像名:TAG

### 容器操作

参考 https://yeasy.gitbook.io/docker\_practice/container

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TD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，通常组合来使用# -p 指定映射端口，将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 查看正在运行的容器

docker ps -a -q

# -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

### 创建镜像

有时候从Docker镜像仓库中下载的镜像不能满足要求，我们可以基于一个基础镜像构建一个自己的镜像。

两种方式：

* 更新镜像：使用docker commit命令
* 构建镜像：使用docker build命令，需要创建Dockerfile文件

### 更新镜像

先使用基础镜像创建一个容器，然后对容器内容进行更改，然后使用docker commit命令提交为一个新的镜像（tomcat为例）。

1. 根据基础镜像，创建容器

docker run --name mytomcat -p 80:8080 -d tomcat

1. 修改容器内容

docker exec -it mytomcat /bin/bash

cd webapps/ROOT

rm -f index.jsp

echo hello world > index. html

exit

1. 提交为新镜像

docker commit -m="描述消息" -a="作者" 容器ID或容器名 镜像:TAG

# 例：# docker commit -m="修改了首页" -a="测试" mytomcat zong/tomcat:v1.0

1. 使用新镜像运行容器

docker run --name tom -p 8080:8080 -d zong/tomcat:v1.0

## 使用Dockerfile构建镜像

Dockerfile是一个包含创建镜像所有命令的文件，使用docker build命令可以根据Dockerfile的内容创建镜像(以tomcat为例)。

1. 创建一个Dockerfile文件 vi Dockerfile

#注意dockerfile指令须大写

FROM tomcat

MAINTAINER zong

RUN rm -f /usr/local/tomcat/webapps/ROOT/index.jsp

RUN echo "<h1>hello world2<h1>" > /usr/local/tomcat/webapps/ROOT/index.html

1. 构建新镜像

docker build -f Dockerfile -t zong/tomcat:v2.0 .

# -f Dockerfile路径，默认是当前目录

# -t 指定新镜像的名字以及TAG