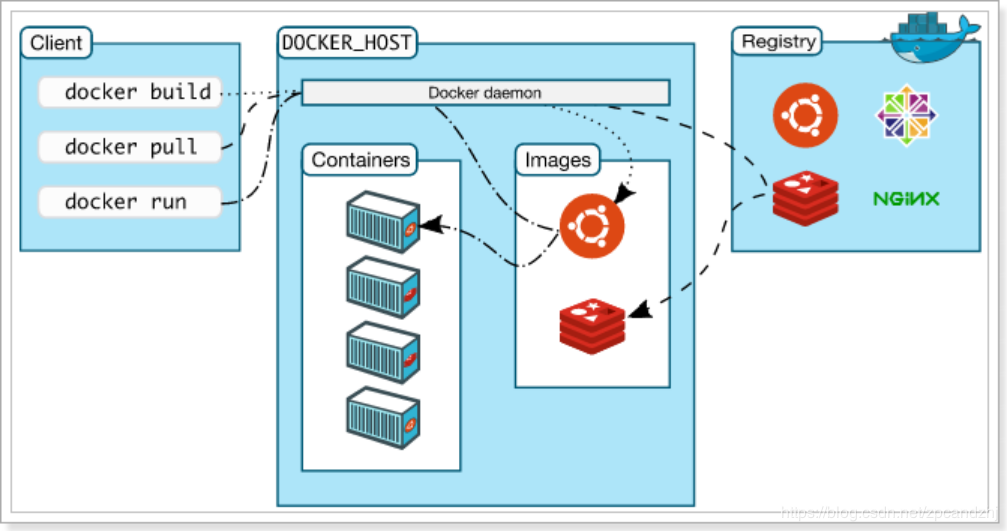
# Docker介绍

## 介绍

1. 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**镜像**中，然后发布到任何流行的 Linux或Windows 机器上，也可以实现虚拟化。
2. 容器是完全使用沙箱机制，相互之间不会有任何接口，容器开销极低。

## Docker架构



1. Client
   1. 用户接口，可以调用Docker的各种命令与Docker daemon(Docker进程)交互
   2. Build：打包程序与依赖环境，产生一个镜像(image)
   3. Push/Pull：从公开/私有仓库(Registry)里，进行镜像推送/拉去操作
   4. Run：获取镜像，创建Docker容器(container)
2. Docker daemon
   1. Docker进程负责支撑Docker 容器的运行以及本地镜像的管理
3. Registry
   1. 仓库，存储镜像文件

## 镜像

1. Docker运行容器需要本地存在相应镜像，本地不存在会通过设置的镜像仓库进行下载
2. 镜像命名方式：[USERNAME]NAME[:TAG]，TAG指版本

## 容器

1. 容器是镜像的一个运行实例
2. 是一个独立运行的应用，包含相应运行环境。

# 常用命令

先启动docker服务，才能运行docker命令

1. systemctl start docker
2. systemctl restart docker
3. systemctl stop docker
4. systemctl daemon-reload 守护进程重启

## Docker基本命令

1. docker pull [OPTIONS] NAME[:TAG|@DIGEST]
   1. 从远程仓库下载镜像
   2. -a 下载所有版本的镜像
2. docker push [OPTIONS] NAME[:TAG]
   1. 推送本地镜像至远程仓库
3. docker run [OPTIONS] IMAGE [COMMAND] [ARG...]
   1. IMAGE 指启动的容器对应的镜像，每个容器拥有一个唯一标识ID
   2. COMMAND 容器启动只是有了运行环境，一般还需要指定具体的执行命令去启动程序
   3. 交互式操作 -i -t
      1. docker run -it IMAGE /bin/bash(启动容器内的命令行)
      2. 此时终端处于容器内，终端命令前缀变为：root@CONTAINER ID:/#
      3. exit退出容器内终端
   4. 后台运行 -d
      1. docker run --name=自定义容器名 -d IMAGE COMMAND
      2. 容器后台运行，会返回容器标识ID
   5. -p 指定主机真实端口与容器虚拟端口之间的映射
   6. -v 将主机目录挂载至容器虚拟目录，这样容器就可以访问主机目录，将数据存储至该挂载目录，实现容器和数据的分离
4. docker ps [OPTIONS]
   1. 查看运行中容器情况
   2. -a 查看所有容器运行情况，包括已结束的
5. docker logs [OPTIONS] CONTAINER
   1. 查看容器内运行日志
6. docker top CONTAINER [ps OPTIONS]
   1. 查看容器内运行进程
7. docker exec [OPTIONS] CONTAINER COMMAND [ARG...]
   1. 向指定容器内添加进程，即在容器内执行其他命令，常用作进入容器的手段
   2. docker exec -it CONTAINER /bin/bash

## image相关命令

1. docker image ls [OPTIONS] [REPOSITORY[:TAG]]
   1. 查看本地镜像文件
2. docker rmi [OPTIONS] IMAGE [IMAGE...]
   1. 删除本地镜像
3. docker commit [OPTIONS] CONTAINER [REPOSITORY[:TAG]]
   1. 从一个容器中创建副本，产生镜像文件
   2. 容器状态有没有要求？？？
4. docker build [OPTIONS] PATH | URL | -
   1. 从零开始创建一个镜像
   2. 需要先创建一个Dockerfile，指明镜像创建内容
   3. -t 可以设置创建镜像的名称
   4. PATH是Dockerfile所在目录路径
5. docker tag SOURCE\_IMAGE[:TAG] TARGET\_IMAGE[:TAG]
   1. 为镜像增加一个标签

## container相关命令

1. docker container ls/ps [OPTIONS]
   1. 查看运行中容器情况
   2. -a 查看所有容器运行情况，包括已结束的
2. docker create [OPTIONS] IMAGE [COMMAND] [ARG...]
   1. 创建一个容器，并没有运行
3. docker start/restart [OPTIONS] CONTAINER [CONTAINER...]
   1. 通过容器ID，重启已停止的多个容器
4. docker stop/kill [OPTIONS] CONTAINER [CONTAINER...]
   1. 通过容器ID，停止运行中的个容器
5. docker rm [OPTIONS] CONTAINER [CONTAINER...]
   1. 删除容器