



DevOps & Docker浅析

主讲人：方孝君

2021/04/19



CONTENTS

目录

0

DevOps简介

软件工程的哲学

1

DevOps关键技术

虚拟化、容器、微服务

2

Docker概念

镜像、容器、仓库

3

Docker运行原理

build、run、pull



DevOps简介

软件工程的哲学

00 / DevOps的演变——“一骑当千”

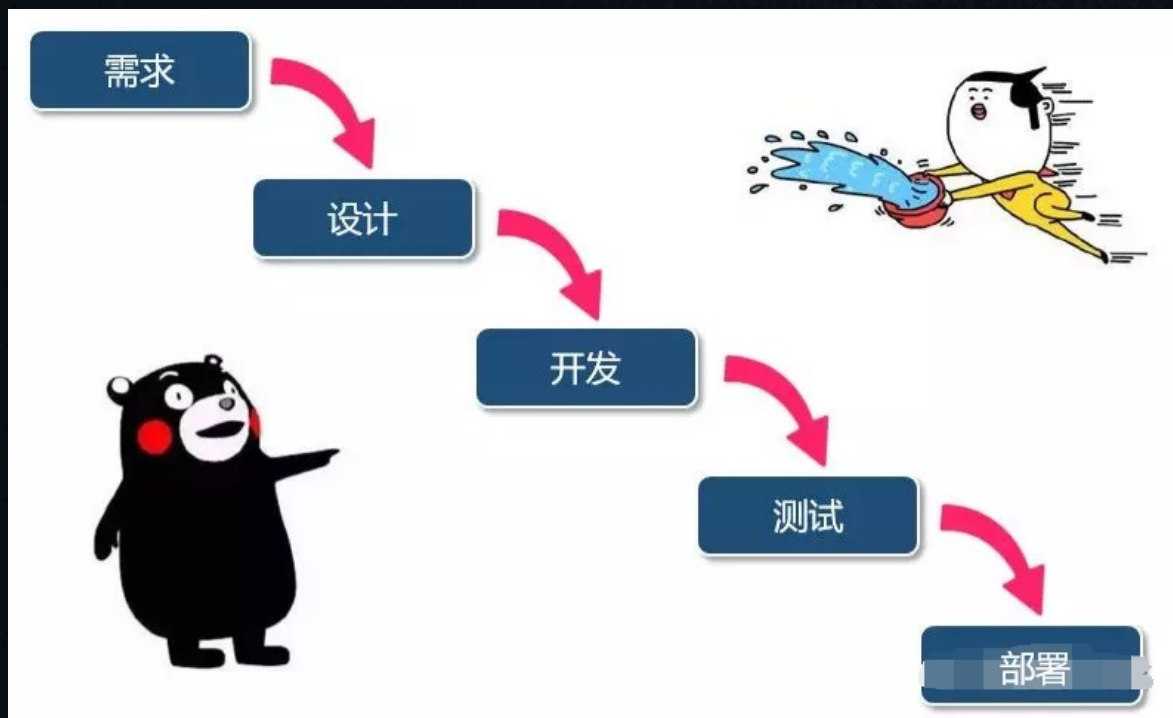


- 最初，程序比较简单，工作量不大，程序员一个人可以完成所有阶段的工作。

00 / DevOps的演变——瀑布模型



- 软件复杂度日益增长，开始出现精细化分工

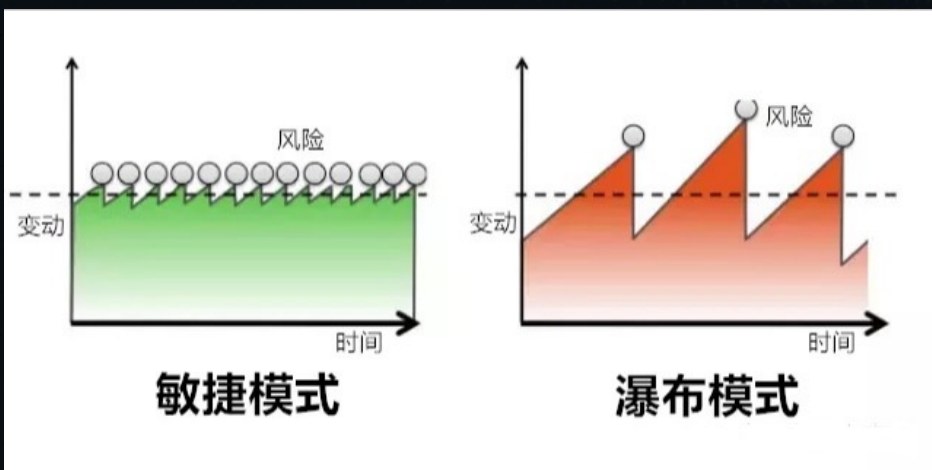


- 早期所采用的软件交付模型，称之为“瀑布（Waterfall）模型”。

00 / DevOps的演变——敏捷开发



- 敏捷开发大幅提高了版本更新速度与开发团队的工作效率



- 敏捷开发带来比瀑布模式更小的风险

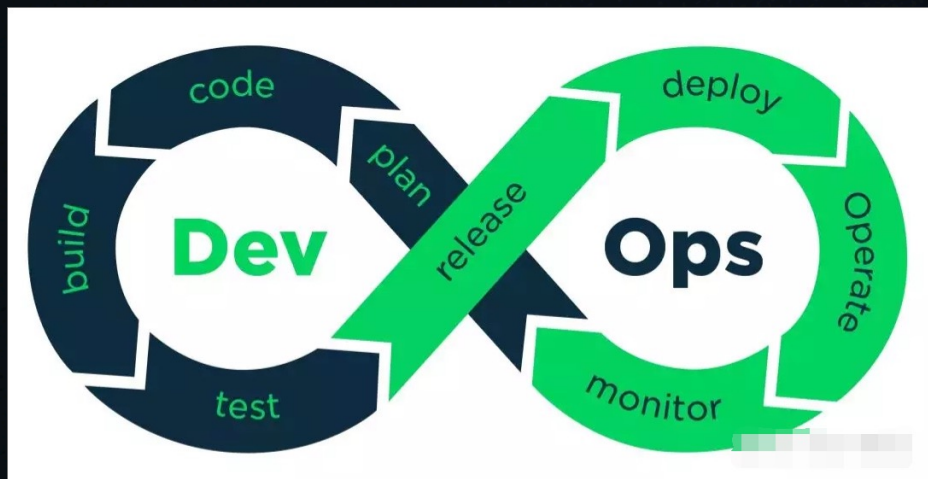
00 / 开发与运维间的矛盾



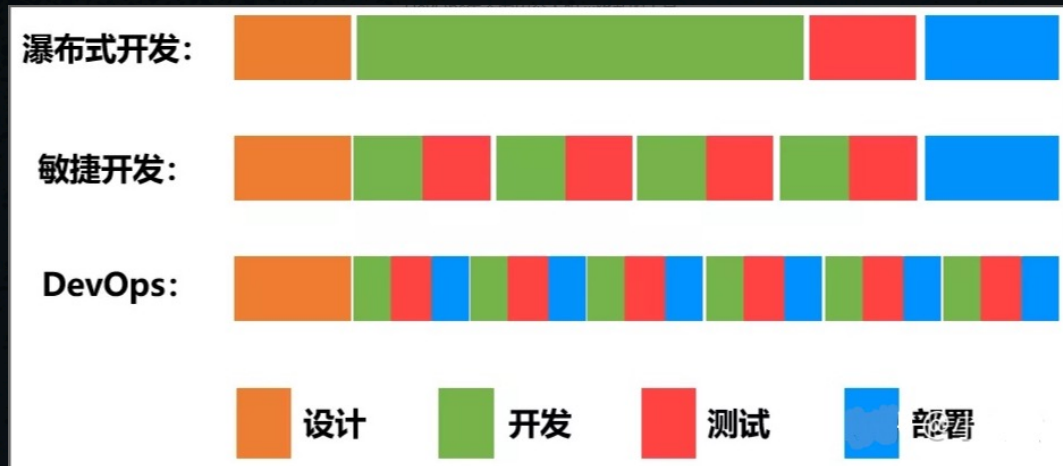
- 开发：快速迭代版本
- 运维：稳定压倒一切

00 / DevOps概念

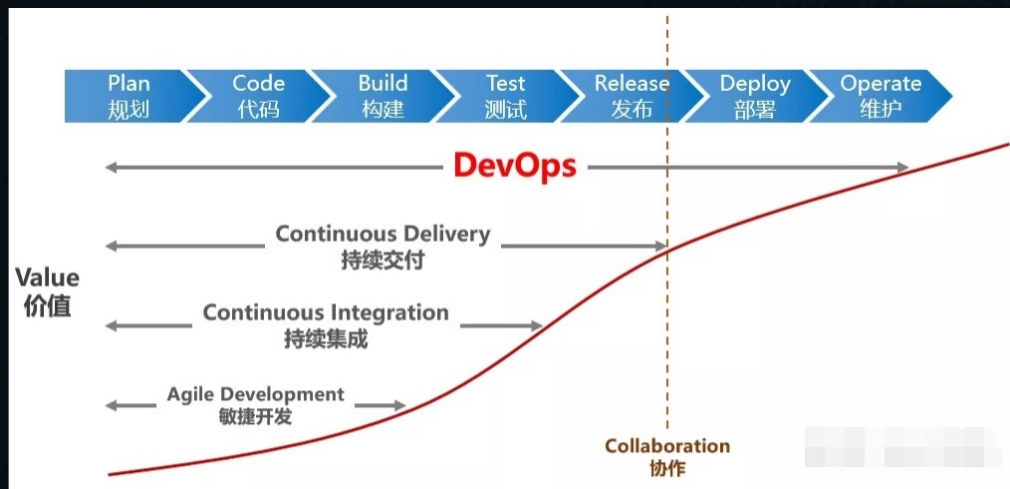
- **Wikipedia** : DevOps is a set of practices that combines software development (Dev) and IT operations (Ops). It aims to shorten the systems development life cycle and provide continuous delivery with high software quality. DevOps is complementary with Agile software development; several DevOps aspects came from the Agile methodology.



00 / DevOps的模式与价值



- DevOps贯穿了软件全生命周期，而不仅限于开发阶段



- DevOps所处的位置，还有它的价值

01

DevOps关键技术

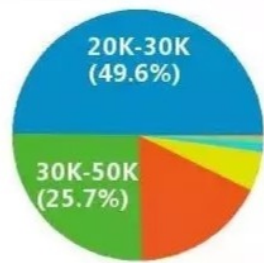
虚拟化、容器、微服务

01 / 需求驱动市场——DevOps工程师

北京devops工程师 • 工资收入水平

北京devops工程师平均工资：¥23710/月，取自 401 份样本

最新招聘 ⁹⁹⁺ 工资收入 就业形势



区间	占比
4.5K-6K	0.7%
6K-8K	0.5%
8K-10K	1%
10K-15K	5%
15K-20K	17.2%
20K-30K	49.6%
30K-50K	25.7%

¥ 23710

近1年 401 份样本 / 可信度：高

你觉得该统计数据准确吗？

偏低 0 靠谱 0 偏高 0

Tencent 腾讯招聘

社会招聘 工作地点 校园招聘 产品和服务 生活在腾讯 登录

搜索工作岗位

查看工作岗位 >

TEG04-腾讯工蜂DevOps开发工程师 (深圳)

分享 收藏

申请岗位

TEG | 深圳 | 技术 | 2021年04月07日

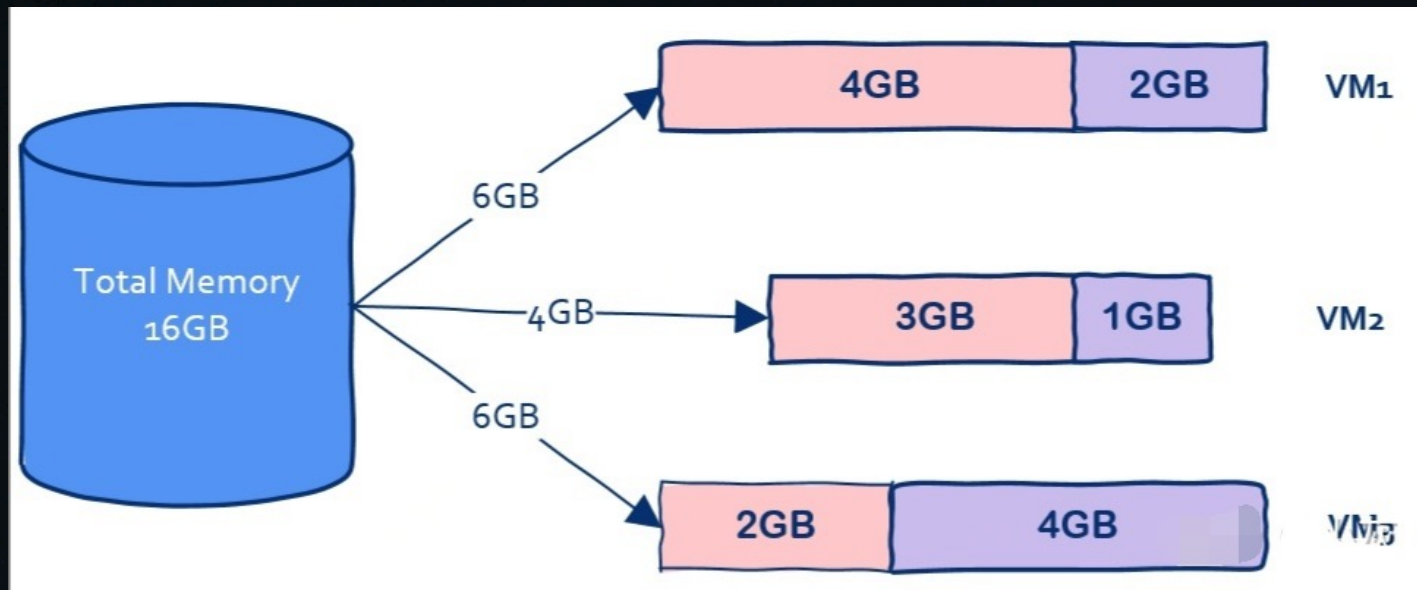
工作职责

负责腾讯工蜂代码管理平台的部署规划，建设运维自动化平台；
负责Git代码管理工具的深度研究，包括协议解析、性能调优等；
参与前沿DevOps技术探索与落地实践，提升团队研发效率；

工作要求

本科及以上学历，2年以上运维和开发经验；
熟悉linux和shell脚本，掌握Java/Python/Golang其中一种编程语言；
擅于开发自动化脚本和工具，有分布式系统的负载均衡、高可用架构、容灾备份等运维经验优先；
掌握Docker/k8s等容器技术，了解微服务架构设计和实现方法；
具备高度的责任心和快速问题定位分析能力，有源码阅读和技术原理钻研精神；
具备以下条件者优先：有Svn/Git代码管理工具的搭建和运维经验；在项目管理、配置管理、CI/CD、自动化测试等DevOps领域有实践和推动经验；

01 / DevOps关键技术——虚拟化

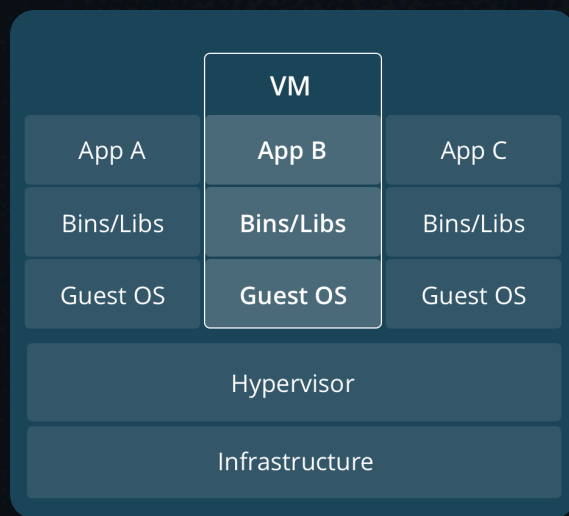
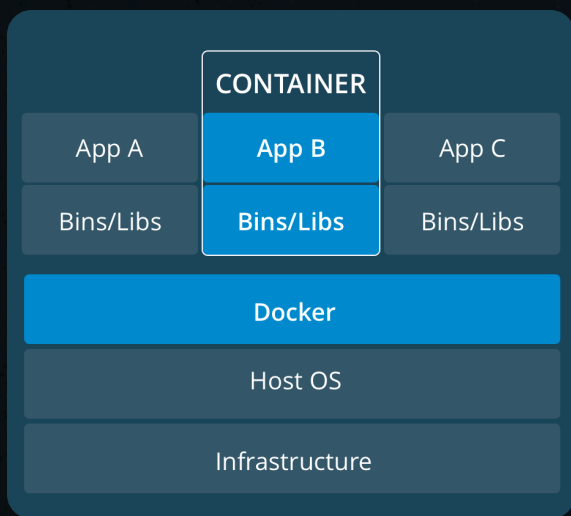


- 和一个单纯的应用程序相比，操作系统是一个很笨重的程序

- 以虚拟机来隔离应用，资源利用率很低，VM本身就要占用很多资源

01 / DevOps关键技术——容器

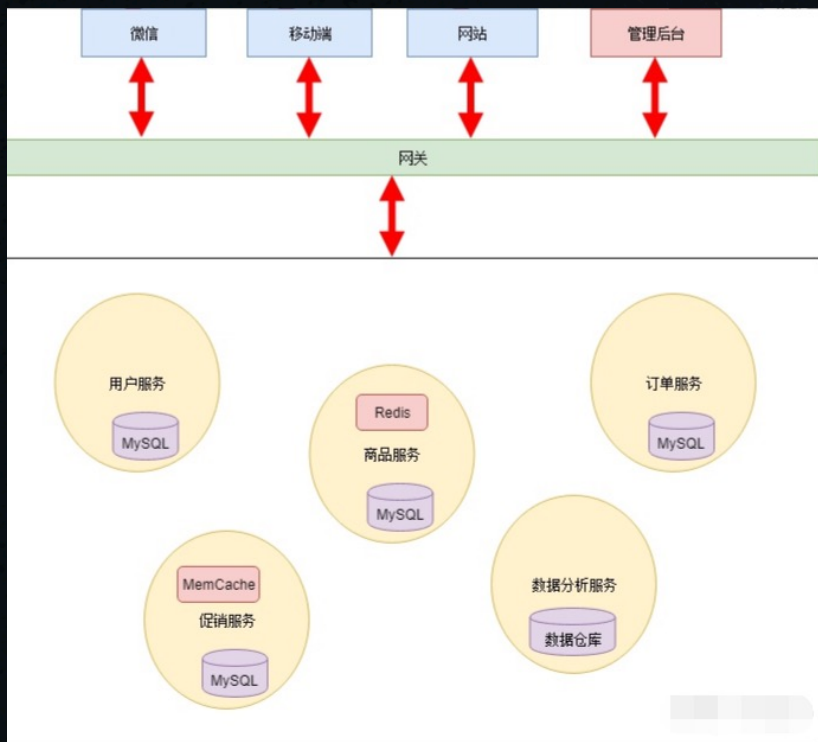
- 容器：Linux容器是与系统其他部分隔离开的一系列进程，具有可移植性和一致性
- 容器与虚拟机



特性	虚拟机	容器
隔离级别	操作系统级	进程级
隔离策略	Hypervisor	CGroups
系统资源	5~15%	0~5%
启动时间	分钟级	秒级
镜像存储	GB-TB	KB-MB
集群规模	上百	上万

01 / DevOps关键技术——微服务

- 微服务的关键：每个服务可以独立开发、测试、部署、迭代，并且可以实现数据库隔离



The screenshot shows a GitLab repository page for 'MetasequolaService'. The page displays a list of submodules with columns for Name, Last commit, and Last update. A red box highlights several submodules.

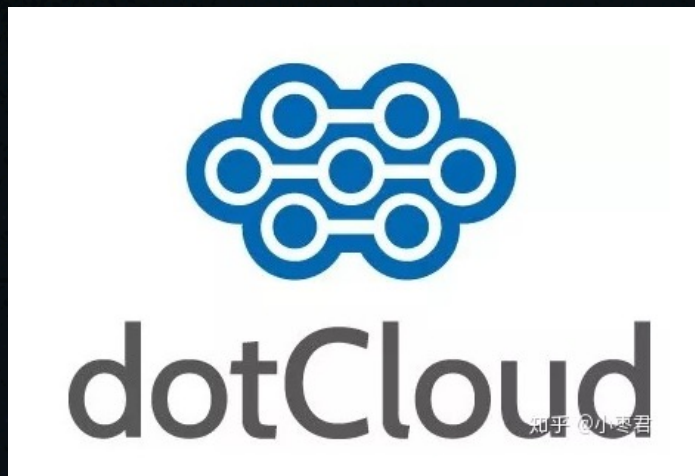
Name	Last commit	Last update
bx-commons	会议回放共享设置	4 days ago
bx-core-server	处理userNumber为null的情况	2 weeks ago
bx-files-server	尝试修复_jpa错误	3 months ago
bx-gateway-zuul	修改测试环境数据库位置, 并修改cicd配置	3 months ago
bx-transaction-server	bx-transaction-server: 测试环境的数据库连接了...	2 months ago
nextlab-tools-server @ ee2d336c	尝试修复_jpa错误	3 months ago
shuishan-portal-server @ 947fb34a	修改会议接口	1 month ago
shuishan-wechat-server @ 78633f62	填坑 视频列表页面完善	1 month ago
super-admin-server @ d0f28b29	修改会议接口	1 month ago
teacher-admin-server @ b7571cda	会议回放共享设置	4 days ago
wechat-official-server @ 13d32265	解决冲突	3 months ago
.gitignore	微信实体类	7 months ago
.gitlab-ci.yml	【基础设施】【我是icy】K8S部署迁移	1 week ago
.gitmodules	添加git submodule	2 months ago
Dockerfile	修复镜像时区	6 months ago
README.md	README.md	8 months ago
cd	messagefix	7 months ago
git	messagefix	7 months ago
pom.xml	题库模块导入导出	8 months ago

02

Docker概念

镜像、容器、仓库

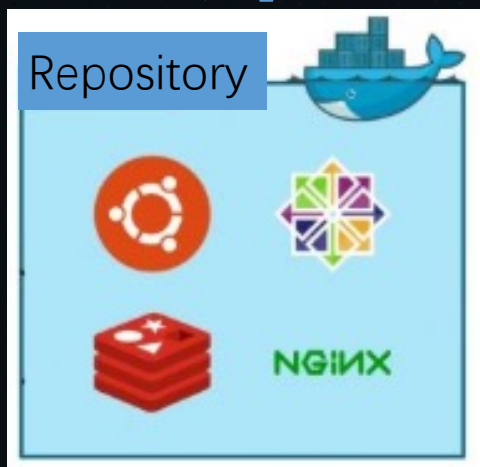
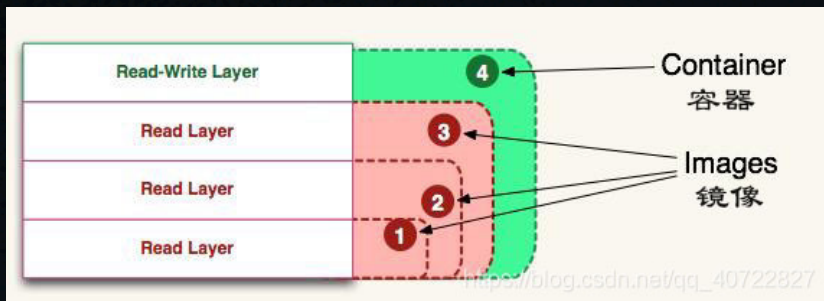
02 / Docker的起源与概念



- dotCloud : Docker的前身，为了企业生计而将其开源

- 我不是容器，我只是容器的搬运工—— Docker本身并不是容器，它是创建容器的工具，是应用容器引擎

02 / Docker核心概念——镜像、容器、仓库



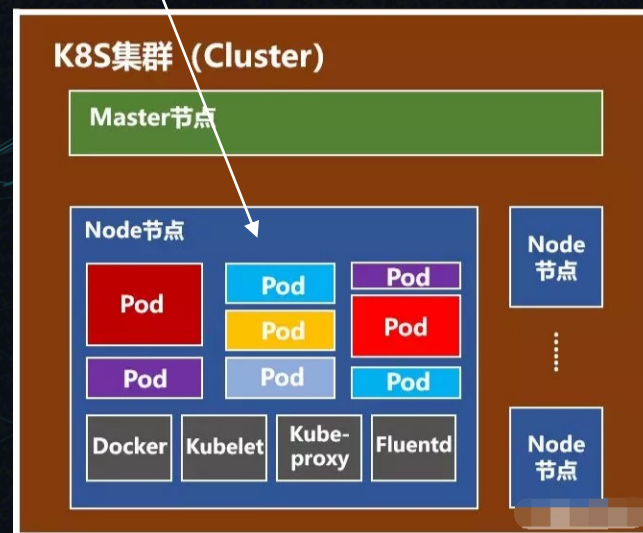
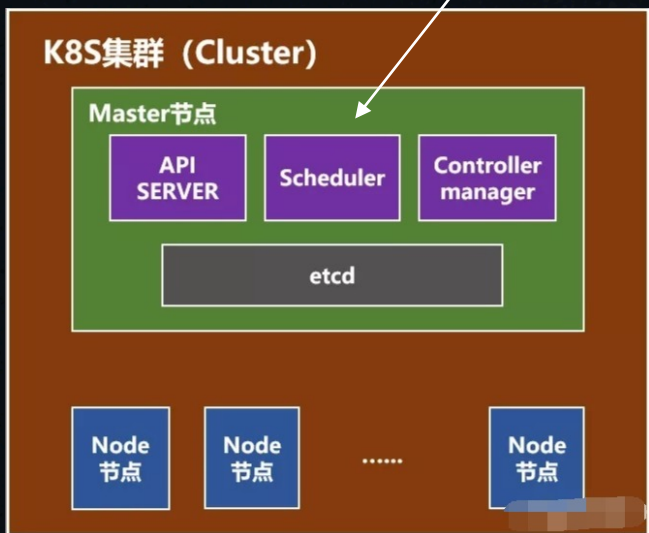
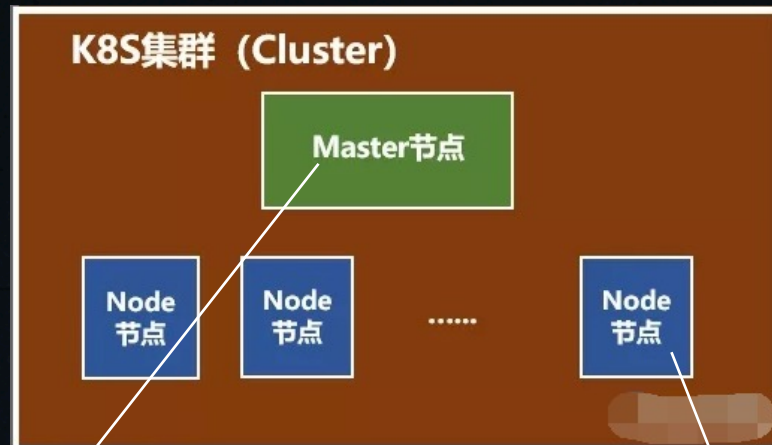
- **镜像(image)**：类似于虚拟机中的镜像，是一个包含有文件系统的面向Docker引擎的只读模板。任何应用程序运行都需要环境，而镜像就是用来提供这种运行环境的。
- **容器(container)**：类似于一个轻量级的沙盒，可以将其看作一个极简的Linux系统环境（包括root权限、进程空间、用户空间和网络空间等），以及运行在其中的应用程序。
- **仓库(repository)**：类似于代码仓库，这里是镜像仓库，是Docker用来集中存放镜像文件的地方。注意与注册服务器（Registry）的区别：注册服务器是存放仓库的地方，一般会有多个仓库；而仓库是存放镜像的地方，一般每个仓库存放一类镜像

02 / 容器编排引擎——K8S



- k8s on github : 76.5k star、27.7k fork

- K8s集群：至少需要三个结点

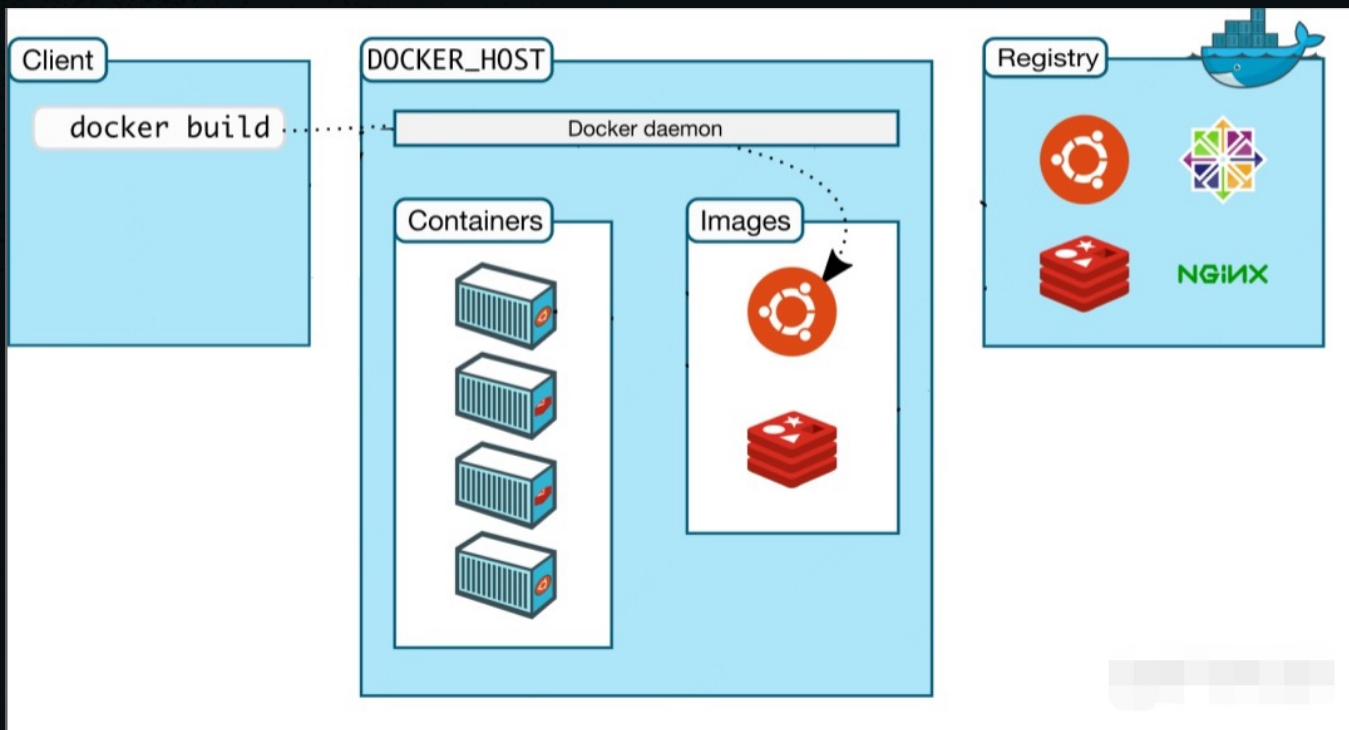


03

Docker运行原理

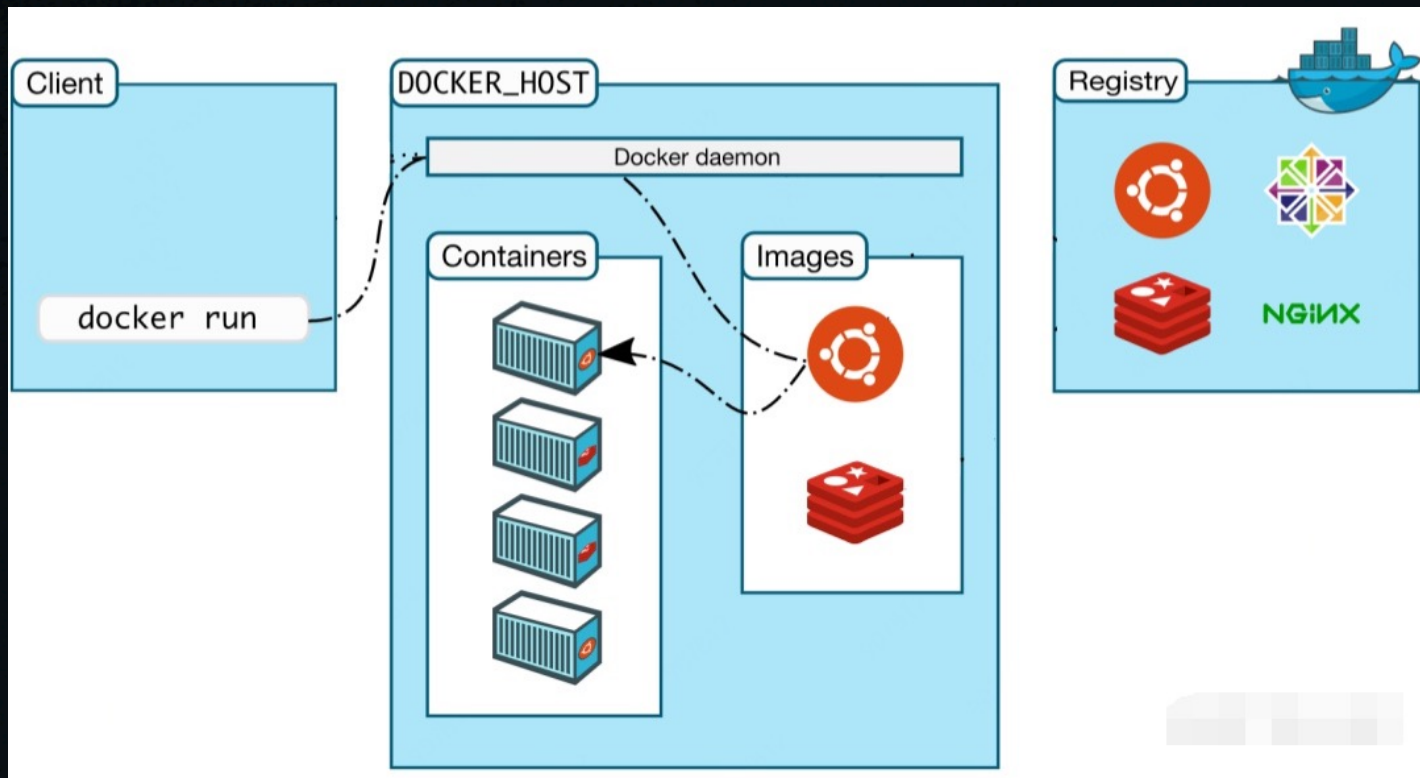
build、run、pull

03 / Docker工作流程——build



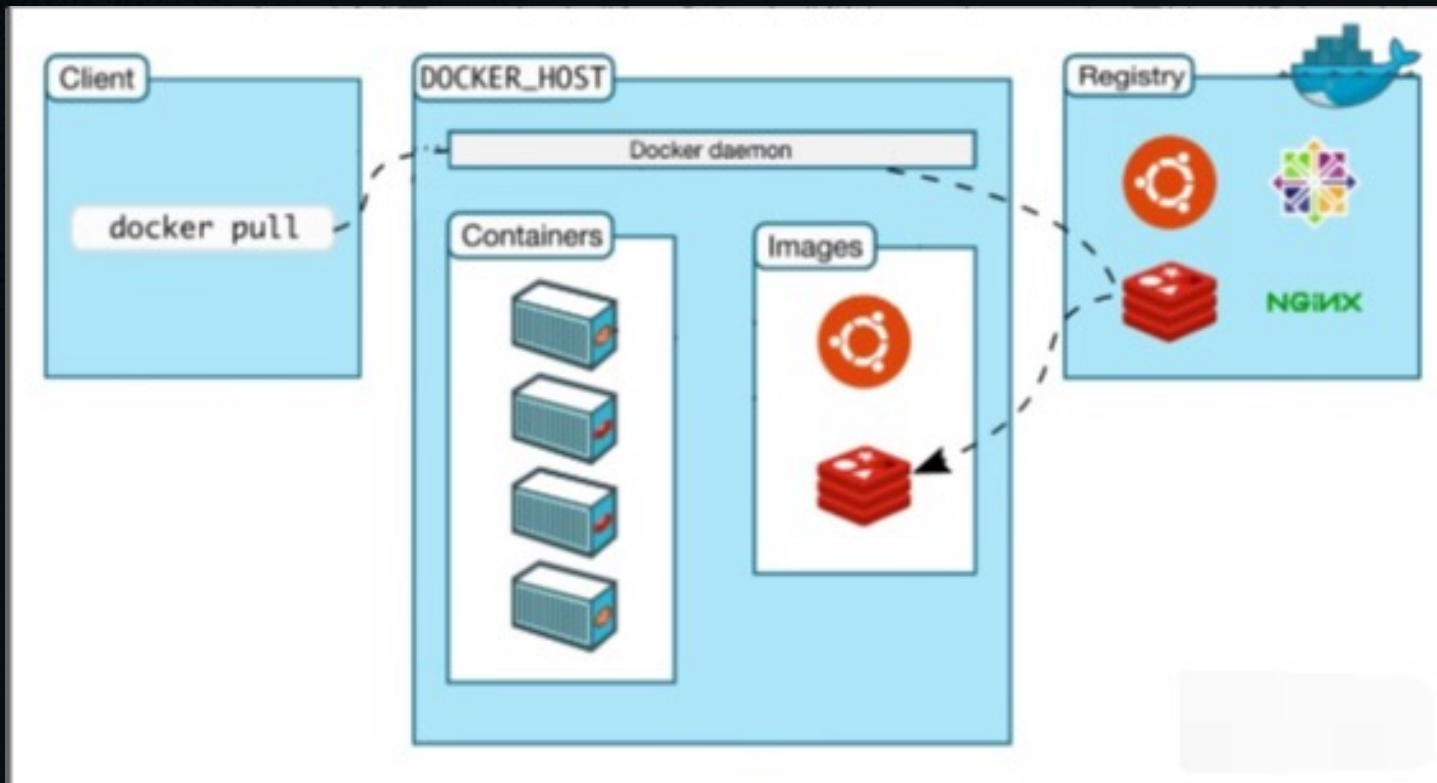
- build : Docker根据dockerfile制作镜像

03 / Docker工作流程——run



- run : Docker找到具体的镜像并执行，形成容器

03 / Docker工作流程——pull&push



- pull&push : 拉取公共镜像&推送私有镜像

03 / Docker底层实现

● NameSpace

Linux中的PID、IPC、网络等资源是全局的，而NameSpace机制是一种资源隔离方案，在该机制下这些资源就不再是全局的了，而是属于某个特定的NameSpace，各个NameSpace下的资源互不干扰，这就使得每个NameSpace看上去就像一个独立的操作系统一样，但是只有NameSpace是不够的

● Control groups

虽然有了NameSpace技术可以实现资源隔离，但进程还是可以不受控的访问系统资源，比如CPU、内存、磁盘、网络等，为了控制容器中进程对资源的访问，Docker采用control groups技术(也就是cgroup)，有了cgroup就可以控制容器中进程对系统资源的消耗了，比如你可以限制某个容器使用内存的上限、可以在哪些CPU上运行等等。



THANK YOU

• 主讲人：方孝君 •

