

Docker不香吗？为什么还要用k8s？

macrozheng 1月8日

以下文章来源于爱笑的架构师，作者雷架



爱笑的架构师

华中科技大学硕士，大厂打工中.....

Docker



k8s



随着k8s 作为容器编排解决方案变得越来越流行，有些人开始拿 Docker 和 k8s进行对比，不禁问道：Docker 不香吗？

k8s 是kubernets的缩写，'8'代表中间的八个字符。

其实 Docker 和 k8s 并非直接的竞争对手，它俩相互依存。Docker 是一个容器化平台，而 k8s 是 Docker 等容器平台的协调器。

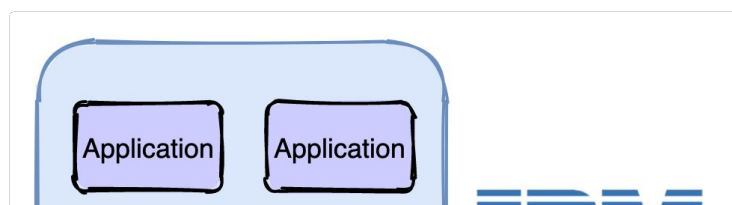
容器化时代来了

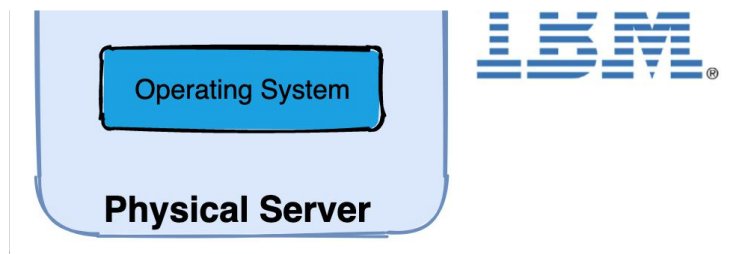
虚拟化技术已经走过了三个时代，没有容器化技术的演进就不会有 Docker 技术的诞生。



虚拟化技术演进

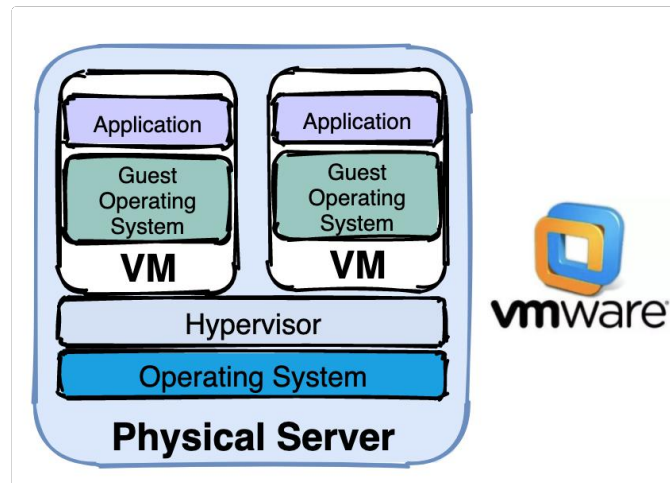
(1) 物理机时代：多个应用程序可能会跑在一台机器上。





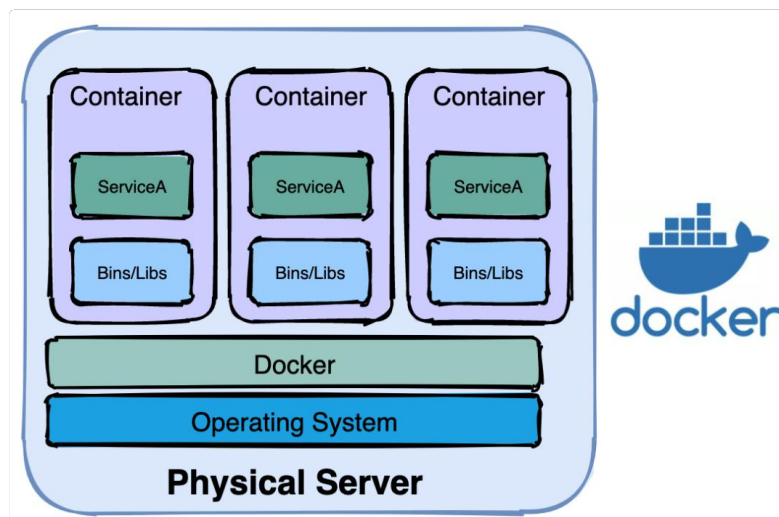
物理机时代

(2) 虚拟机时代：一台物理机器安装多个虚拟机（VM），一个虚拟机跑多个程序。



虚拟机时代

(3) 容器化时代：一台物理机安装多个容器实例（container），一个容器跑多个程序。



容器化时代

容器化解决了软件开发过程中一个令人非常头疼的问题，用一段对话描述：

测试人员：你这个功能有问题。

开发人员：我本地是好的啊。

开发人员编写代码，在自己本地环境测试完成后，将代码部署到测试或生产环境中，经常会遇到各种各样的问题。明明本地完美运行的代码为什么部署后出现很多 **bug**，原因有很多：不同的操作系统、不同的依赖库等，总结一句话就是因为本地环境和远程环境不一致。

容器化技术正好解决了这一关键问题，它将软件程序和运行的基础环境分开。开发人员编码完成后将程序打包到一个容器镜像中，镜像中详细列出了所依赖的环境，在不同的容器中运行标准化的镜像，从根本上解决了环境不一致的问题。

容器化技术的尖刀武器

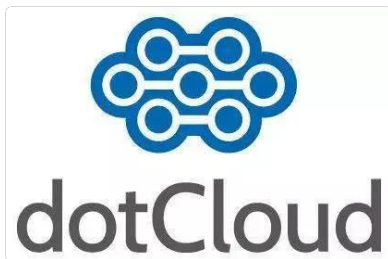


容器化技术的特点

- 可移植性：不依赖具体的操作系统或云平台，比如在阿里云或腾讯云直接随意迁移。
- 占地小：容器只需要其应用程序以及它需要运行的所有容器和库的依赖清单，不需要将所有的依赖库都打包在一起。
- 共享 **bin** 和 **lib**：不同的容器可以共享 **bin** 和 **lib**，进一步节省了空间。

Docker 横空出世

2010年一位年轻小伙子在美国旧金山成立了一家名叫【dotCloud】的公司，开发了 Docker 的核心技术，从此开启了容器技术的时代。



Docker原公司名

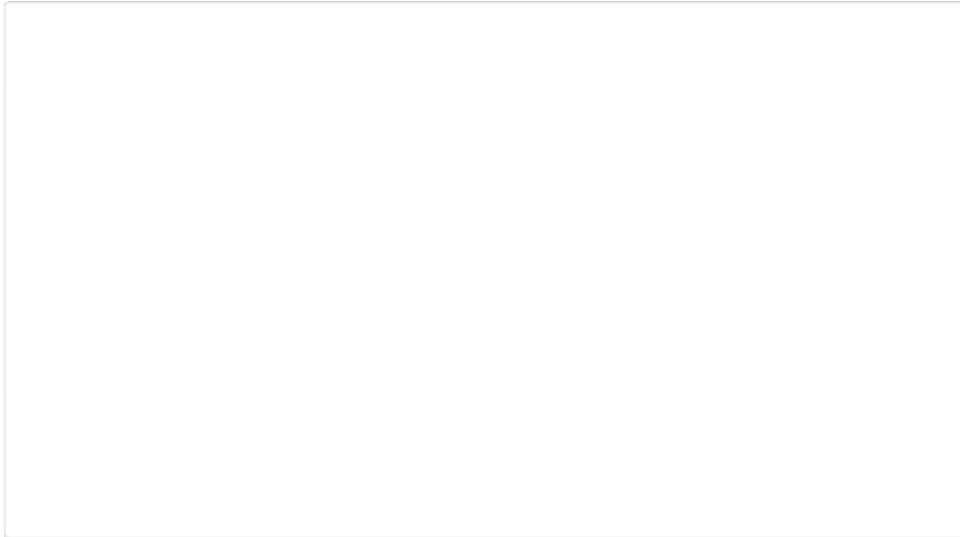
后面 dotCloud 公司将自己的容器技术进行了简化和标准化，取名为 Docker，就是大家熟悉的鲸鱼 logo。





2013年dotCloud 公司宣布将 Docker 开源，随着越来越多的工程师发现了它的优点，Docker 的人气迅速攀升，成为当时最火爆的开源技术之一。

当前有30%以上的企业在其AWS环境中使用Docker，并且这个数字还在继续增长。



Docker使用率越来越高

Docker怎么用？

其实大多数人谈论 Docker 时说的是 Docker Engine，这只是一个构建和运行的容器。

在运行容器前需要编写Docker File，通过 dockerFile 生成镜像，然后才能运行 Docker 容器。

Docker File 定义了运行镜像（image）所需的所有内容，包括操作系统和软件安装位置。一般情况下都不需要从头开始编写 Docker File，在 Docker Hub 中有来自世界各地的工程师编写好的镜像，你可以基于此修改。

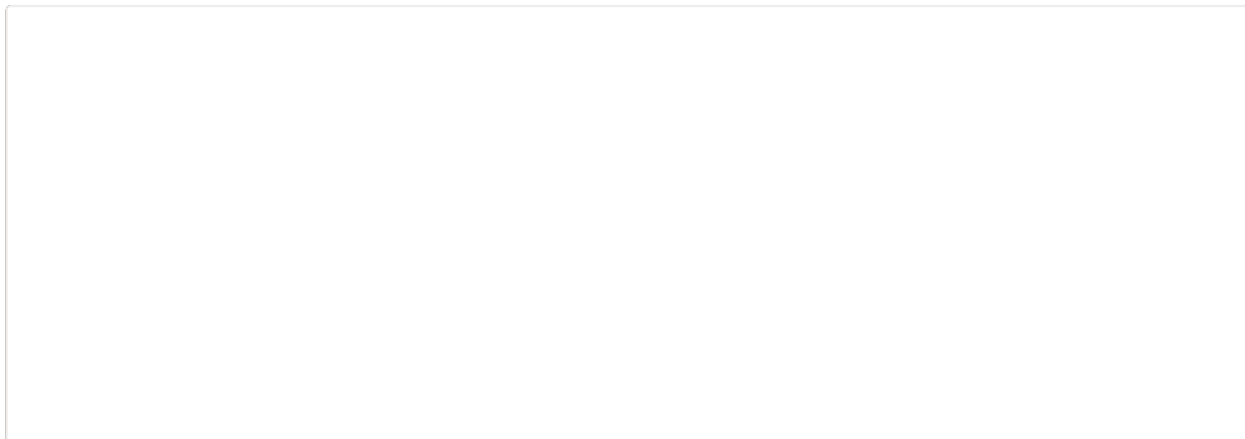
编排系统的需求催生 k8s

尽管Docker为容器化的应用程序提供了开放标准，但随着容器越来越多出现了一系列新问题：

- 如何协调和调度这些容器？
- 如何在升级应用程序时不会中断服务？
- 如何监视应用程序的运行状况？
- 如何批量重新启动容器里的程序？

解决这些问题需要容器编排技术，可以将众多机器抽象，对外呈现出一台超大机器。现在业界比较流行的有：k8s、Mesos、Docker Swarm。

在业务发展初期只有几个微服务，这时用 **Docker** 就足够了，但随着业务规模逐渐扩大，容器越来越多，运维人员的工作越来越复杂，这个时候就需要编排系统解救opers。

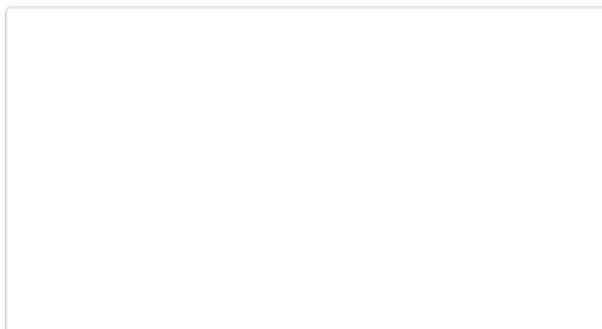


应用程序的声明周期

一个成熟的容器编排系统需要具备以下能力：

- 处理大量的容器和用户
- 负载均衡
- 鉴权 and 安全性
- 管理服务通信
- 多平台部署

| k8s与Docker Swarm江湖恩怨



k8s VS Docker Swarm

如果你非要拿 **Docker** 和 **k8s** 进行比较，其实你更应该拿 **Docker Swarm** 和 **k8s** 比较。

Docker Swarm 是 **Docker** 自家针对集群化部署管理的解决方案，优点很明显，可以更紧密集成到 **Docker** 生态系统中。

虽说 Swarm 是 Docker 亲儿子，但依旧没有 k8s 流行，不流行很大程度是因为商业、生态的原因，不多解释。

| k8s是做什么用的？

K8s是Google研发的容器协调器，已捐赠给CNCF，现已开源。

Google 利用在容器管理多年的经验和专业知识推出了 k8s，主要用于自动化部署应用程序容器，可以支持众多容器化工具包括现在非常流行的Docker。

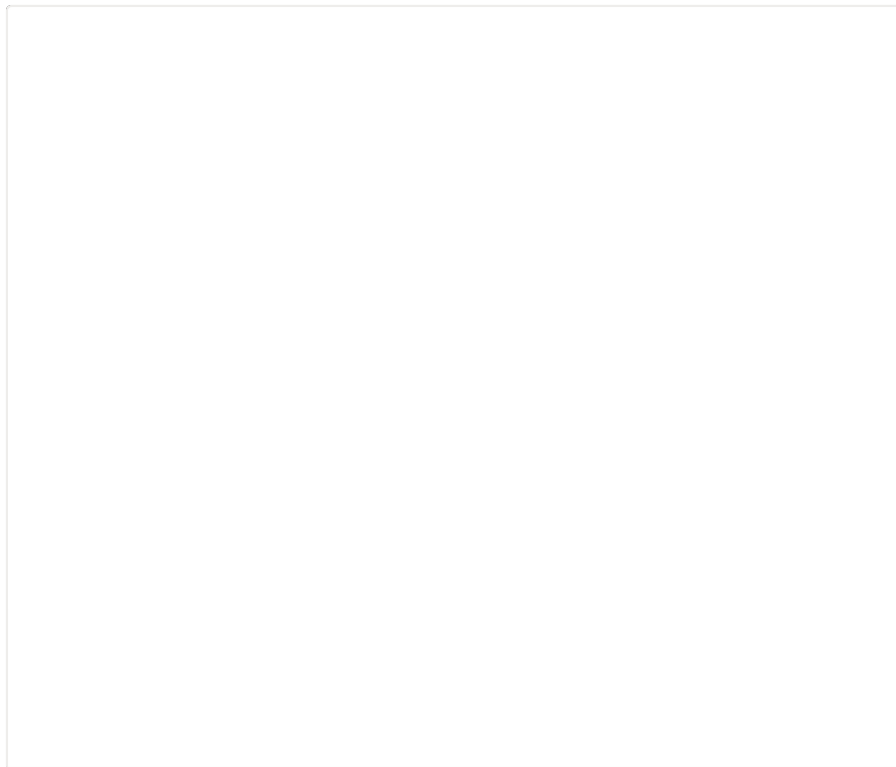
目前k8s 是容器编排市场的领导者，开源并公布了一系列标准化方法，主流的公有云平台都宣布支持。

一流的厂商都在抢占标准的制高点，一堆小厂商跟着一起玩，这就叫生态了。国内的大厂商都在干嘛呢？抢社区团购市场，玩资本游戏，哎？！

| K8s 架构和组件

k8s 由众多组件组成，组件间通过 API 互相通信，归纳起来主要分为三个部分：

- controller manager
- nodes
- pods



k8s集群架构图

- **Controller Manager**，即控制平面，用于调度程序以及节点状态检测。

- **Nodes**，构成了Kubernetes集群的集体计算能力，实际部署容器运行的地方。
- **Pods**，Kubernetes集群中资源的最小单位。

| Docker与k8s 难舍难分

Docker 和 k8s 在业界非常流行，都已经是事实上的标准。

Docker 是用于构建、分发、运行容器的平台和工具。

而 k8s 实际上是一个使用 Docker 容器进行编排的系统，主要围绕 pods 进行工作。Pods 是 k8s 生态中最小的调度单位，可以包含一个或多个容器。

Docker 和 k8s 是根本上不同的技术，两者可以很好的协同工作。

| 开发实践，灵魂追问

（1）没有 **k8s** 可以使用 **docker** 吗？

可以。实际上一些小型公司，在业务不太复杂的情况下都是直接使用 Docker。尽管 k8s 有很多好处，但是众所周知它非常复杂，业务比较简单可以放弃使用 k8s。

（2）没有 **Docker** 可以使用 **k8s** 吗？

k8s 只是一个容器编排器，没有容器拿什么编排？！

k8s 经常与 Docker 进行搭配使用，但是也可以使用其他容器，如RunC、Containerted 等。

（3）**Docker Swarm** 和 **k8s** 怎么选？

选 k8s。2019年底 Docker Enterprise 已经出售给 Mirantis，Mirantis 声明要逐步淘汰 Docker Swarm，后续会将 k8s 作为默认编排工具。

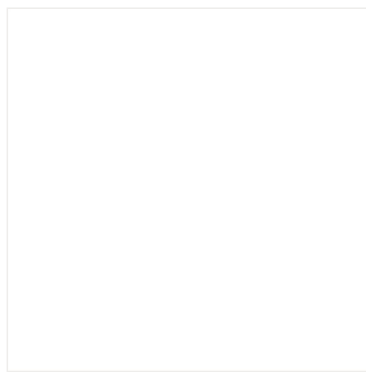
| 最后一个问题

Docker 不香吗？为什么还要用 k8s

Docker很香，但 k8s 在业务达到一定规模后也得启用。学会了吗？

推荐阅读

- [干掉PowerDesigner！这款极简易用的数据库设计工具真香！](#)
- [数据库迁移搞炸了！没用这款开源神器的锅？](#)
- [新来的小哥代码里有坏味道，严重怀疑是钱给少了！](#)
- [小宇宙爆发！Spring Boot 新特性：节省95%内存占用！](#)
- [干掉mapper.xml！MyBatis新特性动态SQL真香！](#)
- [阿里彻底拆中台了！](#)
- [不了解这些超甜的语法糖，别说你会Java！](#)
- [Lombok有啥牛皮的？SpringBoot和IDEA官方都要支持它！](#)
- [40K+Star！Mall电商实战项目开源回忆录！](#)
- [mall-swarm 微服务电商项目发布重大更新，打造Spring Cloud最佳实践！](#)



欢迎关注，点个在看

阅读 3267

分享

收藏

赞 19

在看 7