

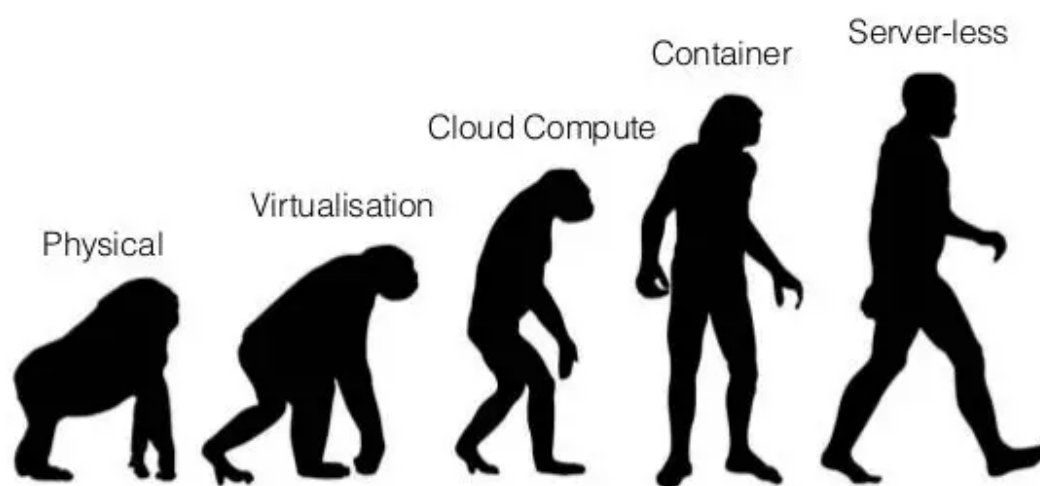
云计算经过这么多年的发展，逐渐进化到用户仅需关注业务和所需的资源。通过 Swarm、K8S 这些编排工具，容器服务让开发者的体验达到很完美的境界。我曾经觉得 Docker 可以替代虚拟机，用户只要关注自己的计算和需要的资源就行，不需要操心到机器这一层。但是因为 Docker 对资源的隔离不够好，各大云厂商的做法还是一个 Docker 对应一台虚拟机，不仅成本高，给用户暴露虚拟机也多余了。

用户为什么需要关注业务运行所需要的 CPU、内存、网络情况？还有没有更好的解决方案？Serverless 架构应运而生，让人们不再操心运行所需的资源，只需关注自己的业务逻辑，并且为实际消耗的资源付费。可以说，随着 Serverless 架构的兴起，真正的云计算时代才算到来了。（其实就是把这部分管理托管出去，就像现在的买阿里服务器一样，不要维护服务器，只管用就行）

容器在开发模式方面并没有提出新的想法，大家还是在用传统的那一套开发模式，需要写一个大而全的后端服务。与之对比，Serverless 架构是事件驱动的，这样让后端的开发体验变得跟前端和移动端很类似了。针对不同客户的需求，先让其购买好相关的资源，然后一个个填坑，给不同的产品添加各种事件处理逻辑就行。这就跟 iOS 开发一样，界面写出来，然后处理一个个事件就好了，大家都很容易理解这种开发模式。

Service Mesh 是对 Kubernetes 中的 service 更上层的抽象，它的下一步是 serverless。

来自：https://blog.csdn.net/weixin_33845477/article/details/87941863



Serverless 的主要优点

1. 开发者更加专注于业务逻辑，开发效率更高。开发一个典型的服务器端项目，需要花很多时间处理依赖、线程、日志、发布和使用服务、部署及维护等相关的工作，基于Serverless架构则不需要操心这些工作。
2. 用户为实际使用的资源付费。用户购买的ECS使用时间一般不到5成，但是为另外5成闲置时间付费了。Lambda按照运行的时间收费，成本会低很多。
3. NO Architecture, NO Ops。架构师的责任是设计一个高可用、高扩展的架构。运维负责整个系统稳定可靠地运行，适当缩减和增加资源。大型云厂商能保证产品的高可用，Serverless架构本身就是高扩展的。Serverless不再需要服务器端的工作人员，给客户节省了大量的资源。架构师和运维的同学应该好好思考一下未来的出路了。架构师可以转型去做销售，整理用户的需求，然后写写CloudFormation的template就好了。
4. 还是成本。IT行业一些领先的公司基础设施非常完善，开发工程师写好代码，然后通过发布平台发布，感觉也是挺方便的。比起Serverless的架构，成本还是要高不少。
 - a. 机器成本。日常、预发、线上，1+1+2=4台服务器少不了。
 - b. 时刻要关注业务数据，盘点资源，看看是否需要扩容和缩减资源。扩容容易，缩减难，造成大量资源闲置。
 - c. 全链路压测是不是很烦？

Serverless的主要缺点

1. 排查问题困难，因为逻辑散落在各处，一个操作可能触发成百上千个Lambda执行。AWS的X-Ray和CloudWatch等产品可以帮助用户排查问题。
2. 准备runtime需要时间，流量瞬间爆发容易导致超时。
3. 带状态的Lambda写起来很困难。
4. Lambda运行有诸多资源限制，比如运行时长、内存、磁盘、打开的文件数量等。
5. 厂商锁定。云计算是赢者通吃的行业，大而全的云厂商优势巨大，Serverless加剧了这种趋势。以前用户还需要自己写很多服务器端的逻辑，迁移的时候，把服务器端代码重新部署一下。采用Serverless架构之后，代码都是各个平台的Lambda代码片段，没法迁移。从客户的角度来看，是不希望自己被某家云厂商所绑架的。所以云计算行业需要做很多标准化的工作，方便用户无缝在各种云之间迁移。

