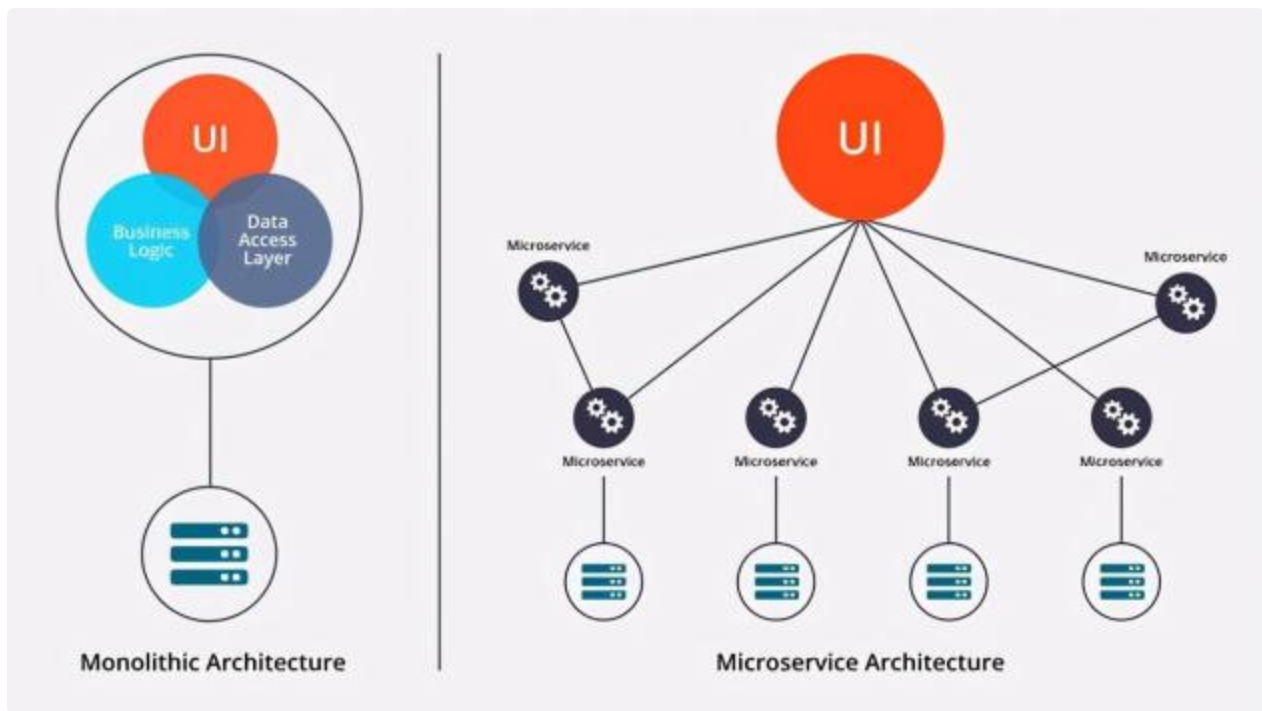


1.前言

过去大多数软件应用属于大型的**单体应用**，以单个进程或者分布在少数服务器上的几个进程的方式运行。如今，这些传统应用仍然广泛存在，但是发布周期较长、迭代也不频繁。在每个发布周期结束之前，开发人员会将整个应用打包后转交给运维团队进行部署和监控。一旦发生硬件故障，运维团队会将应用手动迁移到其他正常的服务器。

如今，这种大型的单体应用正逐渐地被分解成更小的、可独立运行的组件，也就是日渐流行起来的**微服务**。因为微服务彼此之间是解耦的，所以可以被独立开发、部署、升级和伸缩。这使得我们可以对组件进行快速地迭代更新，并且能够及时地跟进快速变化的业务需求。



但是随着可部署的组件的增加以及数据中心的增长，配置、管理以及保持整个系统流畅的运行变得越来越困难。为了获取更高的资源利用率并降低硬件开销，将各个组件部署到什么地方变得越来越难以决策。手动地处理这些事情显然会很痛苦，我们需要采取一些自动化的机制，包括自动的将这些组件调度到服务器、自动化的配置、监管、和故障处理。这正是**Kubernetes**能帮助到我们的地方。

Kubernetes(以下简称k8s)使开发人员不再依赖运维人员的协助就能够部署应用，并且可以按照自己想要的频率进行部署。k8s也能帮助运维团队监控整个系统，并且在遇到硬件故障时对应用进行重新调度。系统管理员的工作重心从**监管单个应用程序**转移到主要**监管k8s和基础设施的其他部分**，而k8s本身会负责管理应用程序。

k8s对硬件基础设施进行**抽象**，将整个数据中心对外暴露为一个巨大的计算资源。它使我们在部署和运行组件时，不用关注底层的服务器。部署多组件应用时，k8s会为每一个组件分配一个合适的服务器然后进行部署，使其能够轻易地发现应用中的其他组件并与之通信。k8s不仅为大部分**本地部署**（on-premises）的数据中心带来了巨大的好处，而且当被应用在由云厂商构建和管理的那种超大型数据中心时，这种增益更加突出。通过k8s，云厂商们能够给开发人员提供一个简单的**云平台**，可部署和运行任何类型的应用程序，云厂商的系统管理员甚至都不用知道这些在他们平台上运行的成百上千的应用到底是什么。随着越来越多的公司把k8s作为运行应用程序的最佳平台，它正成为运行分布式应用的标准方式，不管应用是云端部署还是本地部署。