

5.什么是Kubernetes

1.Kubernetes的起源

2.鸟瞰Kubernetes

3.Kubernetes的核心功能都做了什么

让开发人员聚集到核心应用功能上

让运维团队取得更高的资源利用率

4.对Kubernetes集群架构的理解

控制面板 (Control Plane)

工作节点 (Worker Nodes)

随着系统中可部署组件越来越多，要想管理好它们变得越来越难。因此需要一个更好的方式去部署和管理这些组件，并且支持基础设施的全球性伸缩。谷歌可能是第一个意识到这一点的公司，也是全球为数不多的、运行着成千上万个服务器并且不得不处理如此大规模下的部署管理问题的公司。这迫使他们去找出好的解决方案使成千上万的组件的管理变得有效且成本低廉。

1.Kubernetes的起源

谷歌之前开发了有一款叫做Borg的内部系统（后来还有一个新系统叫做Omega）。这个系统帮助应用开发人员和系统管理员管理那些成千上万的应用和服务。除了简化开发和管理，它也为他们带来了更高的基础设施利用率，当组织很庞大的时候这一点是非常重要的。当你运行了成千上万的机器时，哪怕一丁点的利用率方面的提升也意味着节省了数百万美元。因此开发这样一个系统的动机是显而易见的。

在保守Borg和Omega系统的秘密数十年之后，在2014年，Google开放了Kubernetes，一个基于Borg、Omega和谷歌其他内部系统实践而来的开源系统。

2.鸟瞰Kubernetes

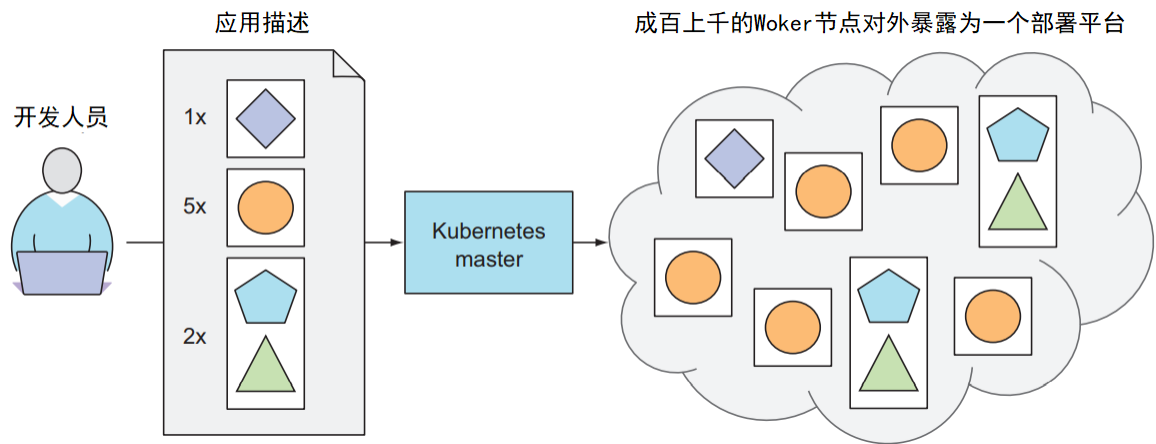
Kubernetes（以下简称K8s）是一个软件系统，它使得我们能够在其上很容易地部署和管理容器化的应用。它依赖于Linux容器的特性来运行异构的应用，而不需要知道这些应用的任何内部细节，也无需在每个主机上手动部署这些应用。由于这些应用是运行在容器中，所以它们不会影响运行在同一台服务器上的其他应用。当你需要在同一个硬件上为不同组织运行应用时，这一点就很关键了。这对于云提供商来说至关重要，因为他们努力在追求硬件的最佳利用率的同时也不得不保障所承载应用的完全隔离。

K8s使我们能够在成千上万的电脑节点上运行应用，就好像所有这些节点是一个单个的、巨大的计算机。K8s对底层的基础设施进行了抽象，因此简化了开发和运维团队的开发、部署和管理工作。

无论我们的集群包含几个节点还是成千上万个节点，通过K8s部署应用的方式都是一样的。集群的规模完全不会带来什么差异性。额外的集群节点只是简单地代表一些对部署应用可用的额外的资源。

3.Kubernetes的核心功能都做了什么

下面展示了一幅最简单的kubernetes系统图：



K8s系统由一个主（master）节点和若干个工作（woker）节点构成。当开发人员向master提交一个应用清单时，K8s会将这些应用部署到worker节点集群中。一个组件具体会部署到哪个节点对于开发人员和系统管理员来说是无关紧要的。

开发人员可以指定某些应用必须一起运行，K8s会将它们部署到同一个工作节点上。其他应用将被分散部署到集群中，但是不管部署到哪儿，它们都能以相同的方式互相通信。

让开发人员聚集到核心应用功能上

K8s可以被当作是集群的操作系统。有了K8s，开发人员不再需要在应用中实现某些与基础设施相关的服务，因为可以依赖K8s来提供这些服务，包括服务发现、扩容、负载均衡、自我修复，甚至领导选举。开发人员因此可以聚焦到实现应用的实际功能上，而不是浪费时间在思考如何将应用与基础设施进行集成上面。

让运维团队取得更高的资源利用率

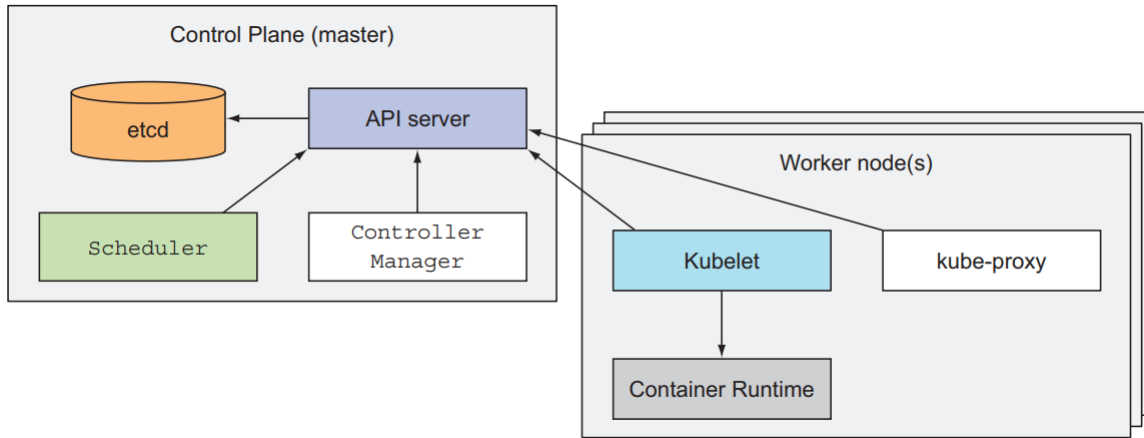
K8s将容器化应用运行在集群中的某个地方，给集群中的应用组件提供相关信息来发现彼此，并保证它们的运行。因为应用不需要关心当前运行在什么节点上，K8s因此可以随时迁移应用程序到其他地方，并且通过混合和匹配应用可以获得比手动调度更高的资源利用率。

4.对Kubernetes集群架构的理解

现在我们来更深入地了解一下K8s集群是由什么组成的。在硬件级别，一个K8s集群由很多节点组成，这些节点可以被分成两类：

- Master节点（主节点）：控制和管理整个K8s集群系统的**控制面板（Control Plane）**位于该节点上
- Worker节点（工作节点）：运行用户实际部署的应用

下图展示了运行在这两种节点上的组件：



控制面板（Control Plane）

控制面板用于控制集群并使其正常运转。它由很多组件构成。这些组件可以运行在单个主节点上，也可以以副本的形式分别部署到多个主节点上，从而确保高可用性。这些组件包括：

- **API Server**：作为用户以及控制面板中的其他组件通信的入口
- **Scheduler**：调度应用程序（为应用的每一个可部署的组件分配一个工作节点）
- **Controller Manager**：它执行集群级别的功能，比如复制组件、跟踪工作节点、处理节点故障等等
- **etcd**：一个可靠的分布式数据存储中心。它能持久化存储集群配置。

控制面板中的组件保持以及控制集群的状态，但是它们不会运行应用程序，这个任务是由工作节点完成的。

工作节点（Worker Nodes）

工作节点就是运行容器化应用的机器。运行、监控以及为你的应用提供服务的任务是由如下几个组件完成的：

- Container Runtime：Docker、rkt或者其他**容器运行时**，负责运行你的容器
- Kubelet：与API Server通信并管理所在节点上的容器
- kube-proxy：负责应用组件之间网络流量的负载均衡

我们将在后面的学习中深入讲解这些组件。