

# kubernetes是什么?

---

是一个自动化容器操作的开源平台，这个平台是开源的，可扩展的。  
虽然kubernetes是一个很厉害的工具，但它不是一个虚拟化服务器，也不是操作系统。

## kubernetes可以做什么?

---

主要是对容器进行操作。

对容器提供了很多编排解决方案，可以使你非常健壮的运行分布式系统。

具体点就是：

- 1.服务发现（暴露容器的访问方式，就跟访问个网站一样）和负载均衡（懂得都懂）
- 2.存储编排（可以自动挂载指定额存储系统）nfs 云存储都可以
- 3.自动发布和回滚（灰度发布，自动更新一部分，慢慢全部替代可以全做）
- 4.自愈（能重启机器，替换或杀死不健康容器，容器就绪前避免调度者发现该容器。）
- 5.密钥及配置管理（存密码和ssh密钥等等，也可以管理敏感信息，比如更新密钥，更新配置等）

## kubernetes有什么组件？，都是干啥的?

---

### master组件:

---

#### 1.apiserver

--这是平台的前端，管理工具如kuboard kubectl等就是通过这个组件实现对K8S集群的管理 API就是应用程序接口的意思，API接口实现计算机软件之间的相互通信。

#### 2.etcd

--键值存储系统，共享配置，服务发现组件，支持一致性（选举产生leader）和高可用，基于这个特性，使得集群具有一定的容错能力，Kubernetes集群的所有配置信息都存储在 etcd 中。

#### 3.kube-scheduler

--起到调度作用，监控所有新建尚未分配到节点上的pod，并且自动选择为POD选择一个合适的节点去运行。（节点的解释：一个节点可以是一个物理机或者虚拟机。每个节点包含运行Pod的必要服务进程。这些服务包含容器运行时，kubelet和kube-proxy）

#### 4.kube-controller-manager

--运行所有控制器 控制器都被合并并运行中一个进程里面。控制器包含节点控制器，副本控制器，端点控制器，service Account &Token控制器。

（节点控制器：负责监听节点停机的事件并作出对应响应）

（副本控制器：负责为集群中每一个 副本控制器对象（Replication Controller Object）维护期望的Pod 副本数）

（端点（Endpoints）控制器：负责为端点对象（Endpoints Object，连接 Service 和 Pod）赋值）

（Service Account & Token控制器：负责为新的名称空间创建 default Service Account 以及 API Access Token）

## 5.cloud-controller-manager

--只运行特定于云基础设施供应商的控制器。默认不安装，想看去官网。

## node组件:

---

### 1.kubelet

--每一个集群节点上都有这个代理程序，确保pod中的容器处于运行和健康状态

### 2.kube-proxy

--是一个网络代理程序，也运行中集群中的每一个节点上，负责维护一些网络规则，这些规则使得集群内外正确的与POD进行网络通信。

### 3.Calico

这也是一个网络组件是生态系统中另一种流行的网络选择，不仅提供主机和pod之间的网络连接，还涉及网络安全和管理。Calico CNI插件在 CNI 框架内封装了Calico的功能。

## 4.容器引擎:

docker容器引擎负责运行容器。Kubernetes支持多种容器引擎：[Docker \(opens new window\)](#)、[containerd \(opens new window\)](#)、[cri-o \(opens new window\)](#)、[rktlet \(opens new window\)](#)以及任何实现了 [Kubernetes容器引擎接口 \(opens new window\)](#)的容器引擎

## Addons(插件)

---

必须的插件:

DNS Addon  
的DNS记录。

--所有的K8S集群都应该有Cluster DNS 这是一个DNS服务器，存放了K8S

### kuboard 以及 Dashboard

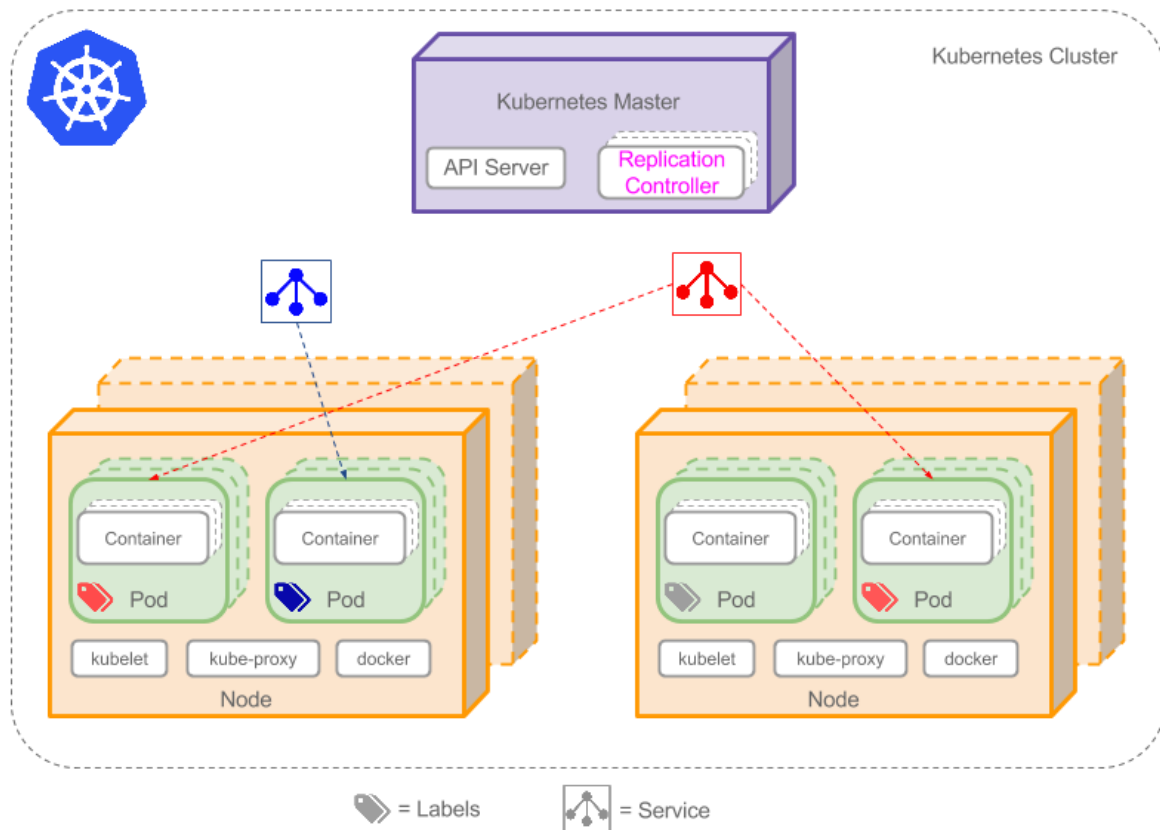
是一个微服务管理界面 (web)

这俩侧重点不一样，具体可以看官网。

## 集群

---

集群是一组节点，这些节点可以是物理服务器或者虚拟机，之上安装了Kubernetes平台。下图展示这样的集群。注意该图为了强调核心概念有所简化。这里可以看到一个典型的Kubernetes架构图。



上图可以看到如下组件，使用特别的图标表示Service和Label：

- PodContainer（容器）
- Label (🏷️)（标签）
- Replication Controller（复制控制器）
- Service (🌳)（服务）
- Node（节点）
- Kubernetes Master（Kubernetes主节点）

## Pod

[Pod \(opens new window\)](#)（上图绿色方框）安排在节点上，包含一组容器和卷。同一个Pod里的容器共享同一个网络命名空间，可以使用localhost互相通信。Pod是短暂的，不是持续性实体。你可能会遇到这些问题：

- 如果Pod是短暂的，那么我怎样才能持久化容器数据使其能够跨重启而存在呢？是的，Kubernetes支持 **卷** 的概念，因此可以使用持久化的卷类型。
- 是否手动创建Pod，如果想要创建同一个容器的多份拷贝，需要一个个分别创建出来么？可以手动创建单个Pod，但是也可以使用Replication Controller使用Pod模板创建出多份拷贝，下文会详细介绍。
- 如果Pod是短暂的，那么重启时IP地址可能会改变，那么怎样才能从前端容器正确可靠地指向后台容器呢？这时可以使用Service，下文会详细介绍。

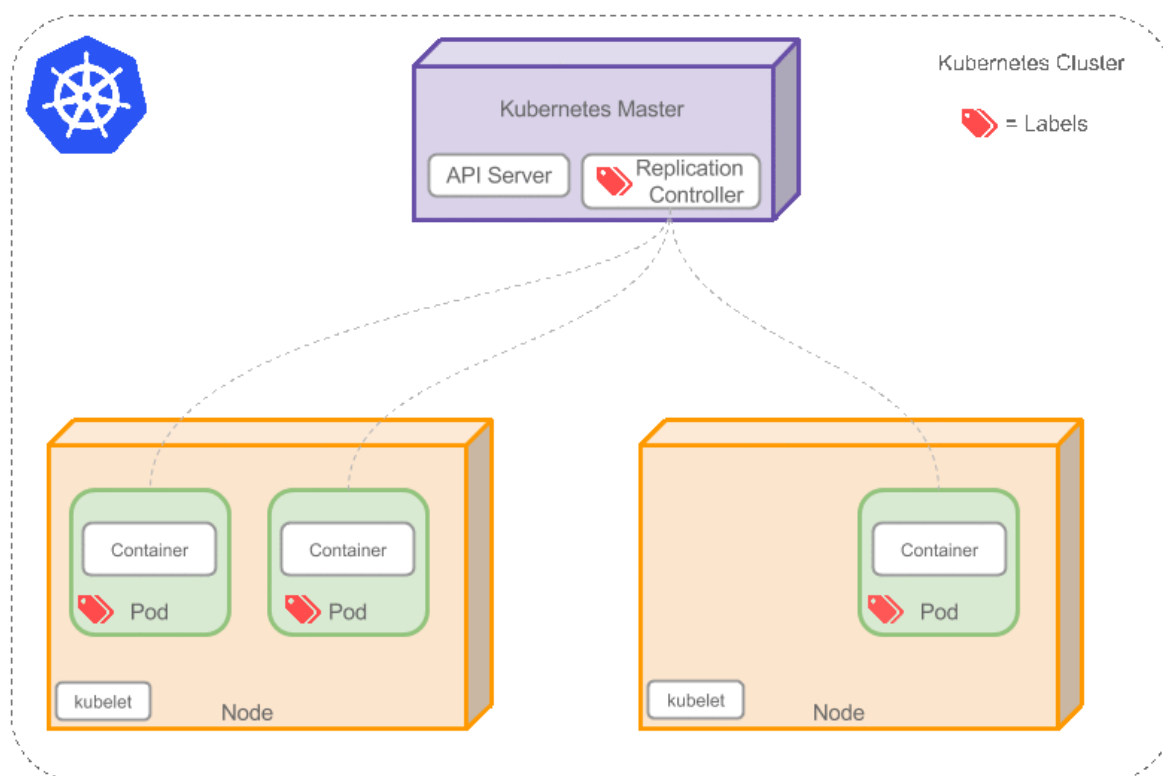
## Label

正如图所示，一些Pod有Label (🏷️)。一个Label是attach到Pod的一对键/值对，用来传递用户定义的属性。比如，你可能创建了一个"tier"和"app"标签，通过Label (**tier=frontend, app=myapp**) 来标记前端Pod容器，使用Label (**tier=backend, app=myapp**) 标记后台Pod。然后可以使用 [Selectors \(opens new window\)](#)选择带有特定Label的Pod，并且将Service或者Replication Controller应用到上面。

## Replication Controller

是否手动创建Pod，如果想要创建同一个容器的多份拷贝，需要一个个分别创建出来么，能否将Pods划到逻辑组里？

Replication Controller确保任意时间都有指定数量的Pod“副本”在运行。如果为某个Pod创建了Replication Controller并且指定3个副本，它会创建3个Pod，并且持续监控它们。如果某个Pod不响应，那么Replication Controller会替换它，保持总数为3。如下面的动画所示：



如果之前不响应的Pod恢复了，现在就有4个Pod了，那么Replication Controller会将其中一个终止保持总数为3。如果在运行中将副本总数改为5，Replication Controller会立刻启动2个新Pod，保证总数为5。还可以按照这样的方式缩小Pod，这个特性在执行滚动 [升级 \(opens new window\)](#)时很有用。

当创建Replication Controller时，需要指定两个东西：

1. Pod模板：用来创建Pod副本的模板
2. Label：Replication Controller需要监控的Pod的标签。现在已经创建了Pod的一些副本，那么在那些副本上如何均衡负载呢？我们需要的是Service。

TIP

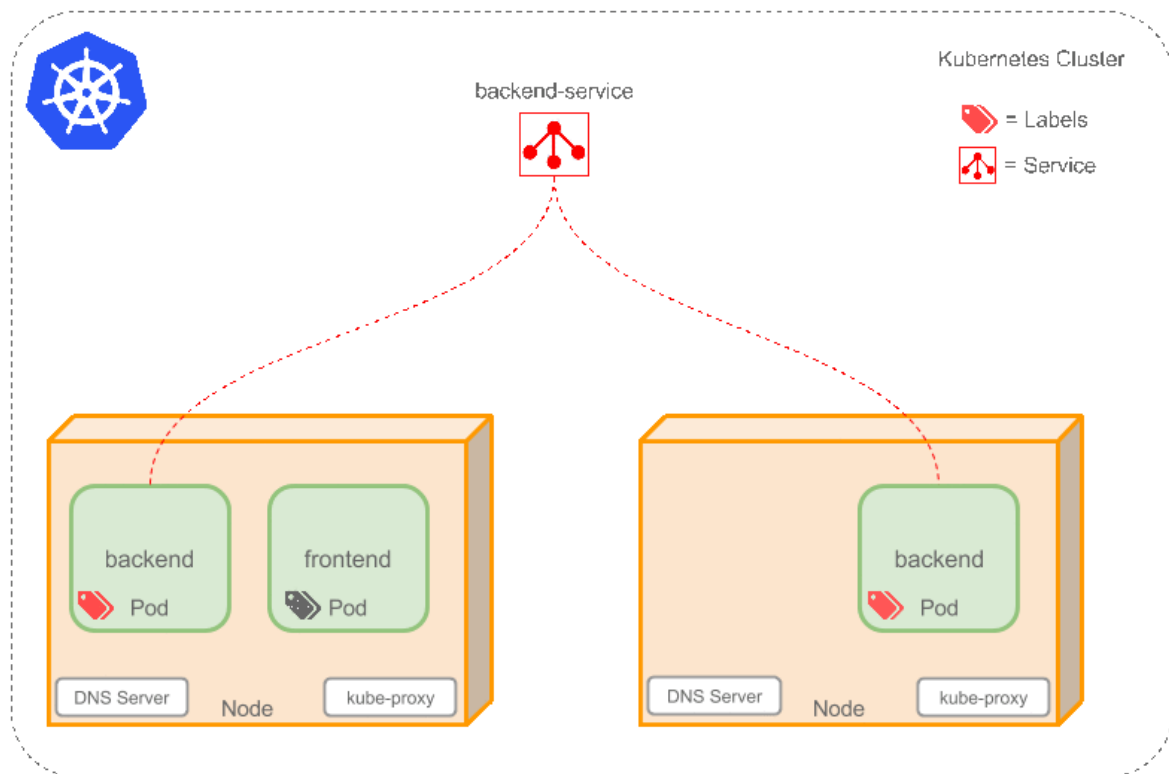
最新 Kubernetes 版本里，推荐使用 Deployment

## Service

如果Pods是短暂的，那么重启时IP地址可能会改变，怎么才能从前端容器正确可靠地指向后台容器呢？[Service \(opens new window\)](#)**抽象** 现在，假定有2个后台Pod，并且定义后台Service的名称为'backend-service'，label选择器为(tier=backend, app=myapp) 的Service会完成如下两件重要的事情：

- 会为Service创建一个本地集群的DNS入口，因此前端Pod只需要DNS查找主机名为 'backend-service'，就能够解析出前端应用程序可用的IP地址。
- 现在前端已经得到了后台服务的IP地址，但是它应该访问2个后台Pod的哪一个呢？Service在这2个后台Pod之间提供透明的负载均衡，会将请求分发给其中的任意一个（如下面的动画所示）。通过每个Node上运行的代理（kube-proxy）完成。

下述动画展示了Service的功能。注意该图作了很多简化。如果不进入网络配置，那么达到透明的负载均衡目标所涉及的底层网络和路由相对先进。如果有兴趣，有更深入的介绍。



每个节点都运行如下Kubernetes关键组件：

- Kubelet：是主节点代理。
- Kube-proxy：Service使用其将链接路由到Pod，如上文所述。
- Docker或Rocket：Kubernetes使用的容器技术来创建容器。

## Kubernetes Master

集群拥有一个Kubernetes Master（紫色方框）。Kubernetes Master提供集群的独特视角，并且拥有一系列组件，比如Kubernetes API Server。API Server提供可以用来和集群交互的REST端点。master节点包括用来创建和复制Pod的Replication Controller。