

lect 23: Kubernetes(K8S)

容器的全生命周期管理

K8S是拿Go语言去写的。刚才我们起docker跑的时候起了一个nodejs。我们现在发现一个nodejs扛不住，希望起多个。如果我们有一个集群有三台机器，每台机器都可以跑nodejs，但是刚才的操作每台机器都得手动启动，如果容器崩了那么我们得手动发现手动重启。

有没有工具能够自动帮我实现这一套东西。我们希望工具来判断把新的容器实例可以放到哪里去，包括容器的重启和回滚。针对虚拟机KVM或者Xen的，我们就可以使用OpenStack就可以使用这个来管理。

KVM / Xen / OpenStack: 针对虚拟机的eureka

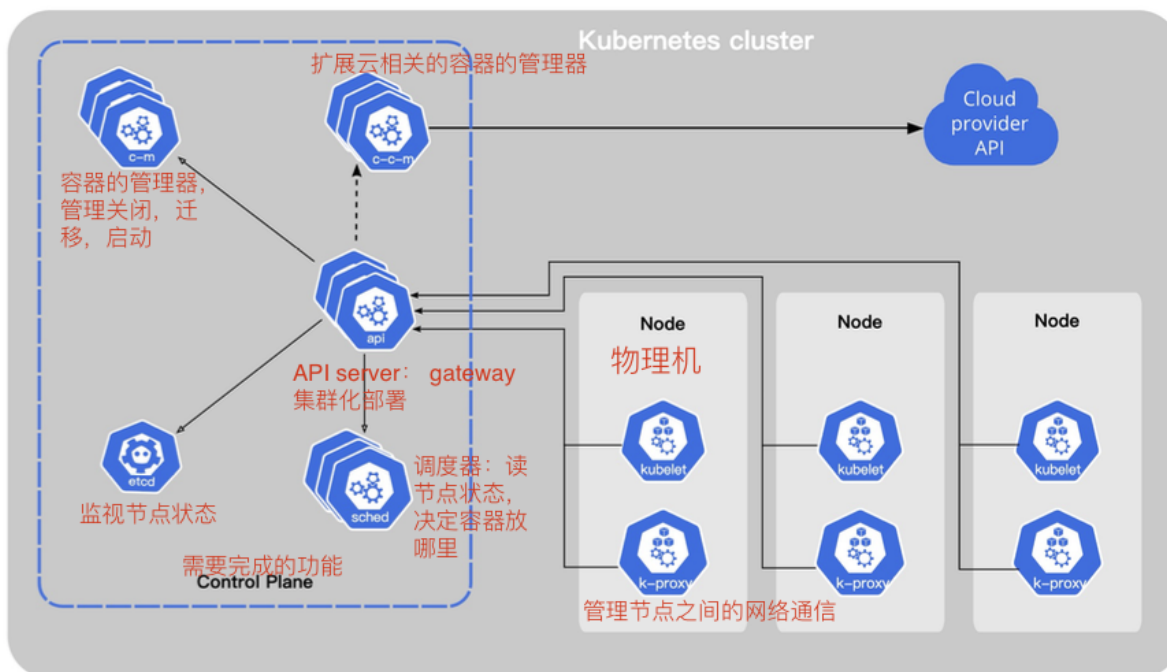
所以K8S就是用来管理容器包裹起来部署的workload，workload可能是一个程序、云盘或者其他东西。它真正跑起来的典型的workload就是应用。**K8S就是负责容器的全生命周期的管理，创建、管理、销毁。**有点像eureka + gateway

- **K8S可以实现如下的功能：**

1. 服务发现和负载均衡，一样的道理，所有的容器都要在K8S这里注册一下，用的就是注册中心和gateway的方式。
2. 存储编排，把哪些卷怎么样挂接到container里去。除了卷，K8S里还允许卷来自于第三方，比如百度云。
3. 因为程序逻辑的问题，导致程序状态不对了。K8S可以把程序回滚，在另一个地方启动一个一模一样的容器把状态回滚到过去的某一个点上。
4. 背包问题，容器就是物品，物理机就是背包，怎么让每个包的负载都一致，没有一个特别满的包。
5. 自愈问题，软件导致容器不能正常运行了。管理的状态发生了变化。比如程序陷入了死循环或者抛出了异常，此时就会释放掉资源再起一个新的实例出来。
6. 安全和配置管理的问题，比如挂载的卷的信息的加密、权限管理等。

在一个集群里面，kubernetes 应该对用户屏蔽容器在哪里，类似gateway.

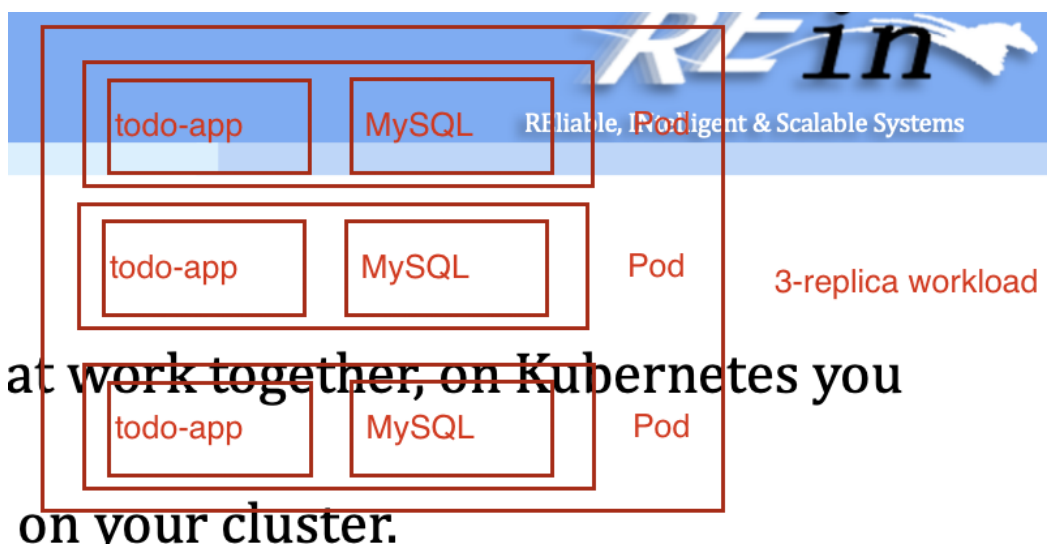
- **SDN：software defined network(软件定义网络)**



它的结构里就包含了K8S里所有的构建。这些Node就表示集群中可以跑容器的物理机。这里面虚线框出来的就是K8S的控制面，就是它要完成的所有的功能。

所以总结一下，K8S实际上就像ZooKeeper一样。它眼里**监控的就是物理机的节点和容器**，而ZooKeeper也是在监控一堆机器。

在K8S眼里，要部署的就是一个workload，可以理解为就是一堆需要一起工作的东西，以pod的形式出现。Pod（最小可部署单元）就是来描述负载里是由多少种容器构成的。



A **workload** is an application running on Kubernetes.

pod 由多个容器构成

Whether your workload is a single component or several that work together, on Kubernetes you run it inside a set of **pods**.

如上的3个构成了一个replica set。所以pod就和image一样，只是现在一个image包含了多个应用。一个pod就是scheduler调度的最小资源。

在做容器的迁移的时候，用户是感知不到的。这些复杂性全部由K8S来管理。

minikube



- minikube quickly sets up a local Kubernetes cluster on macOS, Linux, and Windows. We proudly focus on helping application developers and new Kubernetes users.
- <https://minikube.sigs.k8s.io/docs/>

```
~ ➤ time minikube start
🐾 minikube v1.13.0 on Darwin 10.15.6
🌟 Using the docker driver based on user configuration
👉 Starting control plane node minikube in cluster minikube
🔥 Creating docker container (CPUs=2, Memory=3892MB) ...
🔧 Preparing Kubernetes v1.19.0 on Docker 19.03.8 ...
🔍 Verifying Kubernetes components...
🌟 Enabled addons: default-storageclass, storage-provisioner
💡 kubectl not found. If you need it, try: 'minikube kubectl -- get pods -A'
👉 Done! kubectl is now configured to use "minikube" by default

Executed in 23.96 secs  fish           external
   usr time    1.66 secs  237.00 micros    1.66 secs
   sys time    0.78 secs   943.00 micros    0.78 secs
```

补充：**Docker和K8S的区别**

Dockers是容器化技术，K8S是一套自动化部署工具，可全生命周期管理Dockers容器。

K8S是谷歌开发的容器集群管理系统。在Dockers技术的基础上，为容器化的应用提供部署运行、资源调度、服务发现和动态伸缩等一些列完整功能，提高了大规模容器集群管理的便捷性。

K8S是一个完备的分布式系统支撑平台，具备完善的集群管理能力，多扩多层次的安全防护和准入机制、多租户应用支撑能力、透明的服务注册和发现机制、内建智能负载均衡器、强大的故障发现和自我修复能力、服务滚动升级和在线扩容能力、可扩展的资源自动调度机制以及多粒度的资源配额管理能力。

Docker是一个开源的应用容器引擎，让开发者可以打包它们的应用以及依赖包到一个可移植的镜像中，然后发布到任何流行的Linux或Windows机器上，也可以实现虚拟化。

Docker使用客户端-服务器架构模式，使用远程API来管理和创建Docker容器。Docker容器通过Docker镜像来创建，容器与镜像的关系类似于面向对象编程中的对象与类。<https://www.csdn.net/txpp520>