

简单的说：一个虚拟机上，有很多用户在使用。也就是说，一个机器能满足好几个用户。感觉还不错哈。但这是有问题的，

1. 其物理资源还是共享的，一旦某个用户的操作占用大量资源，那其他用户就会受到很大影响
2. 这里的用户也可以是应用(像k8s)，每个应用在一个虚拟机上执行，总有先后，如何保证重要应用优先执行
3. 资源共享，也存在安全问题

在HBase1.1.0发布之前，HBase同一集群上的用户、表都是平等的，没有优劣之分。这种‘大同’社会看起来完美，实际上有很多问题。最棘手的主要有这么两个，其一是某些业务较其他业务重要，需要在资源有限的情况下优先保证核心重要业务的正常运行，其二是有些业务在某些场景下会时常‘抽风’，QPS常常居高不下，严重消耗系统资源，导致其他业务无法正常运转。

这实际上是典型的多租户问题，

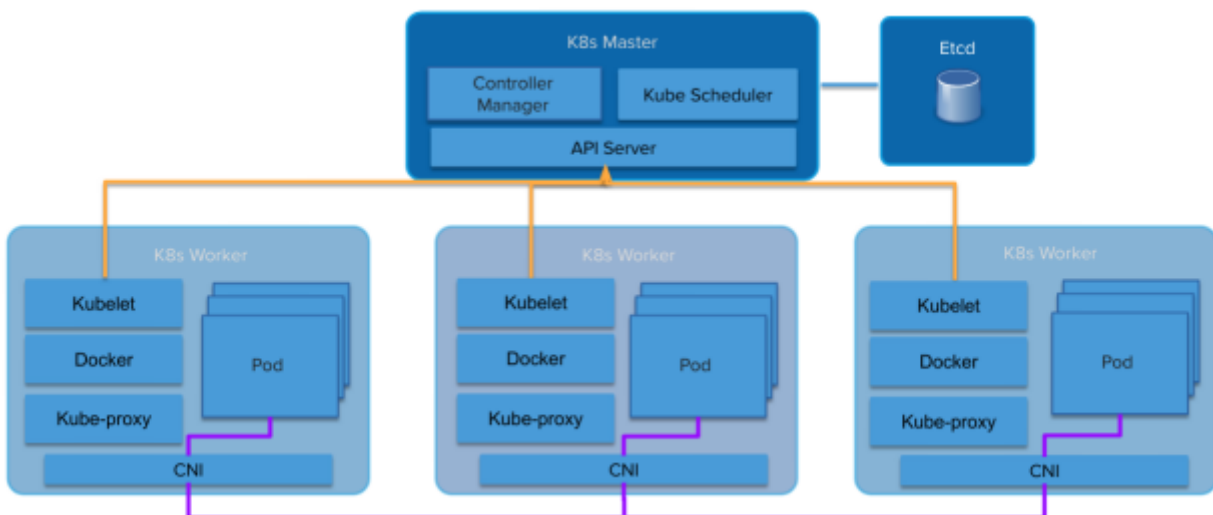
来自: [https://blog.csdn.net/lw\\_ghy/article/details/60779482](https://blog.csdn.net/lw_ghy/article/details/60779482)

k8s

我认为这也可能成为容器崩溃的核心原因——多租户机制。

Linux容器在设计之初并没有考虑到安全的隔离沙箱（例如Solaris Zones或者FreeBSD Jails）。相反，容器采用的是共享内核模式，其利用内核功能提供基础性的进程隔离功能。正如Jessie Frazelle在文章中指出，“容器并不真实存在。”

更麻烦的是，大多数Kubernetes组件无法感知到租户。虽然我们可以使用命名空间以及Pod安全策略，但API本身确实不具备租户感知能力。此外，kubenet或者kube-proxy等内部组件也存在同样的问题。这意味着Kubernetes能够提供的只是一种“软租户”模式。



抽象泄漏。建立在容器之上的平台会继承容器技术的诸多软租户因素。正如建立在硬多租户虚拟机基础之上的平台（包括VMware、Amazon Web Srevices以及OpenStack等），也都继承了这种硬租户机制一样。

Kubernetes集群本身成了“硬租户”面临的第一道坎，也因此导致大部分用户只能使用“多集群”这一新兴模式，而非“单一共享”集群。相信很多朋友都发现了，谷歌GKE Service的用户往往需要面向多个团队部署数十个Kubernetes集群，有时候每一位开发者都拥有自己的集群。这类作法最终导致了严重的Kube泛滥问题。

“这类作法最终导致了严重的Kube泛滥问题。”

原文：<https://blog.csdn.net/M210ZgSsVc7r69eFdTj/article/details/85333601>