

容器系列一：容器起源【精选期刊 . 2】

分享人：代豪

有容云首席技术官

作者经历：

1984 年接触编程，开始用 Basic 写程序。独立承担过食糖期货交易系统开发及主要负责过高性能方面的项目，研发项目深入涉足通信、数据库、集群、高性能计算、网格计算、移动终端。

注：本期分享由代豪原创，有容云整理发布。

说到容器，这个概念现在非常火；实际上在 IT 领域内各种技术一直是逐步迭代演进的。这里先从大家用的 windows 说起，我们经常有这样的操作，打开一个 word 进行文档编辑，然后又打开了另外一个程序的安装包 setup。 Ok，这样我们可以简单认为开了两个 windows 进程，一个 word 进程，一个 setup 进程，那么可能存在的问题：

- 1.两个程序可以访问相同的资源，比如 C 盘上的数据，在编辑文档时，setup 进程占用 cpu 过高，影响编辑文档的体验。
- 2.Windows 里有个注册表的东西，这个被搞乱会带来很多麻烦，setup 如果设计兼容性不好有可能把注册表搞乱。
- 3.对于开发者而言，如果写这两个程序的人或团队属于不同机构的，那么有可能产生配置冲突或兼容性的问题。

我们再看一下 linux 系统

我们知道 linux 系统有很多发行版本，比如 centos，ubuntu 等，我们的应用有 app1 for centos，也有 app2 for ubuntu 的，现在我需要在一个 OS 里能跑 app1 和 app2，怎么办？而对于 app1 和 app2 的开发者需要编译出不同发行版的应用程序，for centos，for ubuntu 等。

对于开发者而言，一个程序写好测试过后，在最终的用户环境里可能会产生环境不匹配而导致运行异常的问题。

那么如何解决这些问题呢？

一个自然的想法就是“隔离”+“运行环境打包”，把各个程序或 app 及相应的运行环境打包交付成镜像，这样可以避免环境兼容的问题，又可以互补干扰和破坏，那么在一台机器上同时运行各个打包交付的镜像时就需要把各个镜像隔离加载运行才能保证各个 app 的运行环境独立而不冲突。这里大家自然会想到虚拟机，没错，虚拟机完全可以做这样的事情，每个镜像就是一个虚拟机文件，虚拟机的优势很明显。但我们现在只是希望在一台机器上高效运行多个程序，那么用虚拟机的方式缺点也很明显：

1. 每个 app 镜像里需要包含一个 OS，体积过大

一个机器运行多个不同环境的 app 需要多个 OS 虚拟机，管理和运行的效率成问题。

我们看到这里的主要矛盾是为了运行不同环境的 app 要把 OS 也加进来作为支撑环境，那么为了进一步容易管理和提升运行效率，能不能把 OS 这一层精简呢？答案是可以，这就是容器要做的事情。

精简 OS 这一层，各个 app 执行环境共享 OS 是一个有效的方法，我们先看看 linux 为什么

能做到这一点。实际上现在的 linux 可以理解为两部分，一是内核，一是发行环境。比如我们说的 centos , ubuntu , redhat 等，就是指的发行环境。大家都在相同的 linux 内核的外围发行自己的管理执行程序，各个发行版本就是目前的 centos , ubuntu 等，这时候解决方法就来了：就是提供多个发行环境共享一个 linux 内核的功能。

好，这时候容器的概念就出来了：linux 容器可以理解为多个发行环境共享一个内核，因此各容器可以有不同的发行环境(如 centos , ubuntu , ...)，但只有一个共同的内核，所以你不能启动一个其它内核的运行环境(如 windows)，如果你需要运行一个 centos 环境的容器和一个 windows 环境的容器，这时你需要的就是虚拟机了，而不是容器，或者说需要带有虚拟机内核功能的容器了，再就是你的发行环境里需要地内核版本功能版本比容器地共享内核版本高也会有问题，那么需要升级你的共享内核，也就说简单了容器就是共享 OS 内核的隔离支撑环境，那么 linux 容器的基本核心功能有：

1. 文件系统隔离，每个容器运行在完全独立的根文件系统里。
2. 网络隔离，每个进程容器在自己的网络命名空间里，拥有自己的虚拟接口和网络地址。
3. 资源隔离，为每个进程容器分配不同系统资源，如 cpu , mem , io 等。

再看看 windows，上述的三点同样需要，不同的是大家都知道 windows 里有个注册表，这个对 windows 而言非常重要，因此 windows 的容器对这个问题自然不能回避，那么 windows 容器是如何解决这个问题的呢？Windows 有一个主机注册表，容器里的注册表继承它，但修改时容器自己保存了自己对注册表的修改部分，那么容器里把注册表改乱了也不影响到 host 和其他容器，进一步 Windows 里用 jobobject 来限制容器的资源使用，而 linux 使用 cgroup 来限制容器的资源使用，限于篇幅不再这里详述了。

实际上容器目前功能还在继续丰富迭代，进一步演化出了这种各样的体系和应用，比如强大

的 api，存储，容器集群，基于容器的高性能计算和基于容器的分布式计算等，后续会逐步展开。

温馨提示：

有容云携手 Rancher Labs 推出【[Rancher | 实战微信群](#)】，在线为您分享 Docker 技术干货，更有往期回顾精选期刊等你拿！

本群汇集了 Rancher 中国最强技术精英团队及业内技术派高人，宗旨是为了大家拥有更专业的平台交流 Rancher 实战技术，实时与 Rancher 创始团队面对面！同时欢迎各位分享自己的经验、疑难问题，我们将定期邀请分享嘉宾做各类话题分享及回顾，共同实践研究 Docker 容器生态圈。

对 Rancher 和 Docker 技术感兴趣、或对本文中细节需继续探讨的朋友，欢迎加入本群参与讨论！

加微信群方法：

- 1.关注【有容云】公众号
- 2.留言“我要加群”

QQ 群号：216521218

