# Docker与vm、lxc区别

docker 是什么？docker 能干什么？docker 为什么这么火？相信不少人都有这个疑问。我也有。觉得 docker 涉及到的技术太多了，从 linux 内核到各种云计算的解决方案，对于我来说实在是有点艰难的。然而学了一个月的 docker 相关的东东，还是想简单谈谈，说一下我的理解，希望对 docker 的初学者有所帮助。

**Docker 的历史**

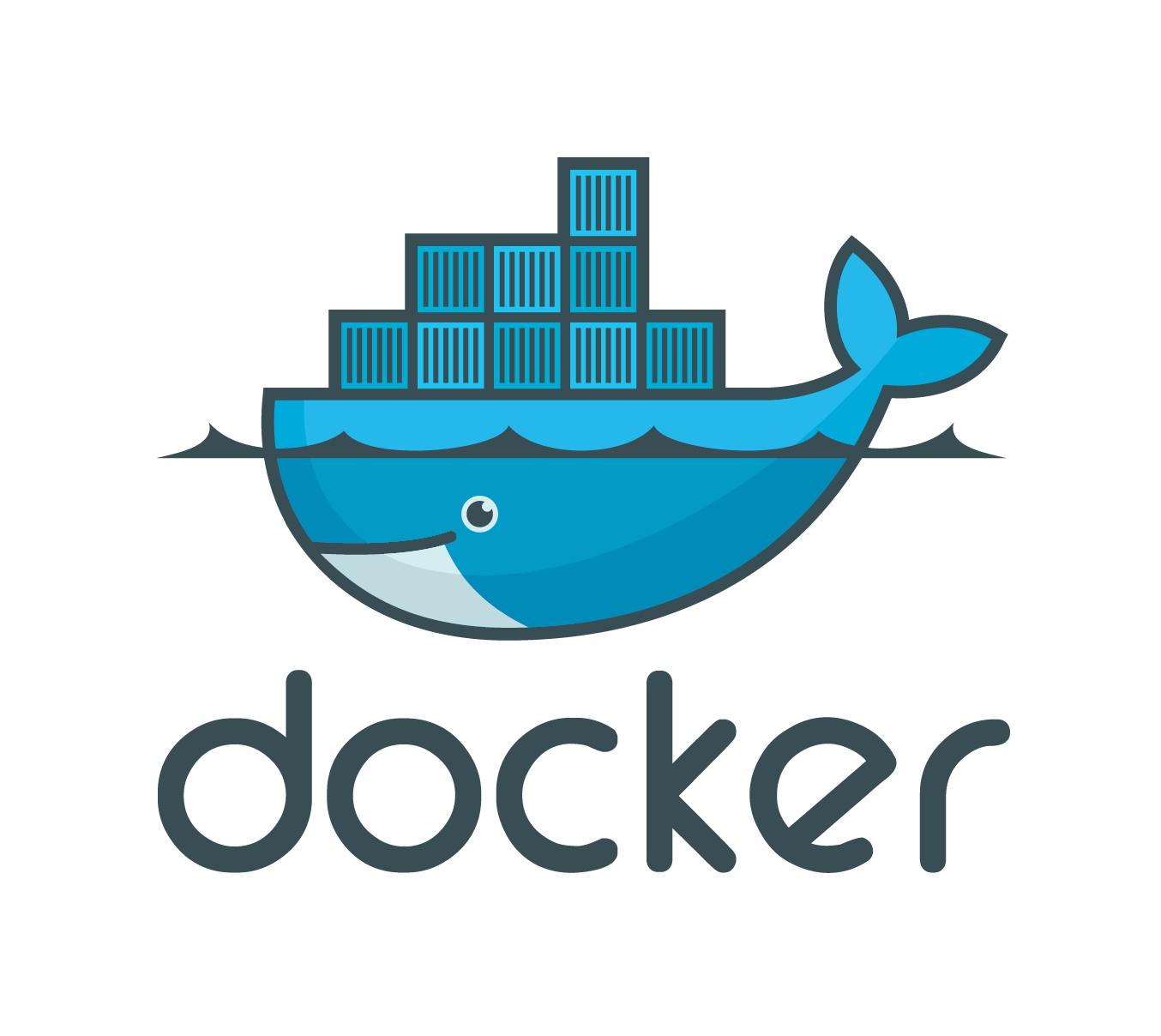
当我们在谈论一项技术的时候，经常会追溯它的历史，这可以帮助我们更好地理解它是怎么发展起来的，是为了解决什么问题。

docker 起源于国外一家很酷的公司 dotclound， 2013 年在 Github 上开源并推进，项目吸引了世界范围内大量的目光，社区非常活跃，许多巨头也参与进来，而 dotclound 直接就把自己改名为 docker 了….

关于 docker 的开源发展可以详细看这篇: Docker开源之路

**Docker 是什么**

那么，docker 究竟是什么？docker 是一个基于LXC 的高级容器引擎。听起来是不是不知道在说什么？简单地说，docker 是一个轻量级的虚拟解决方案，或者说 —— 一个超轻量级的虚拟机。你一定理解虚拟机是什么，那么，你现在可以认为 docker 是一个秒级启动的虚拟机，可以轻易创建和删除，就这一点，是不是酷毙了。

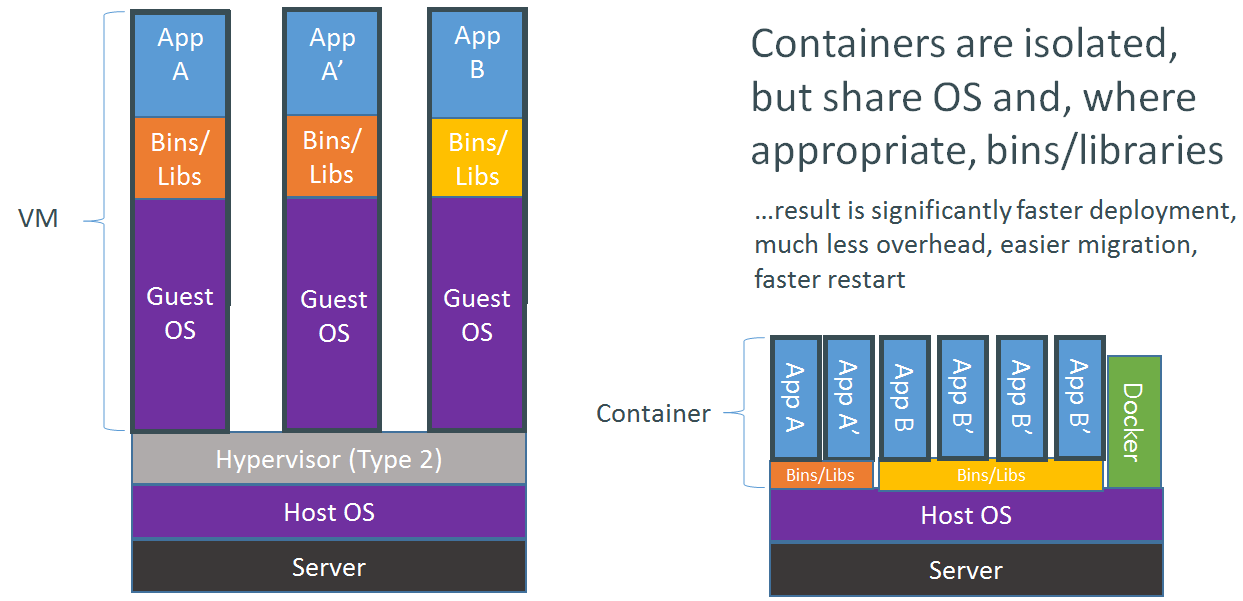


我们会经常看到这个 docker 的图，它告诉我们，docker 是一种集装箱式的工作方式。正如我们会将各种不同的货物统一打包成一个个集装箱，进行标准的管理和运输，在 docker 的世界里，我们把应用和应用所依赖的运行环境打包成一个个 image，然后分发到任意支持 docker 的平台，就可以在这些平台运行我们的应用，提供服务。

那支持 docker 的平台有哪些？因为 docker 是基于 linux 的，在任意的 linux 发行版我们都可以原生地支持 docker，只要对应的内核版本大于 3.10 并且是 64bit，，而在 Windows 和 Mac 中，我们可以通过 boot2docker 来运行 docker，这几乎意味着，所有的主流平台都支持 docker 的运行 —— 从此你不再需要为跨平台而苦恼。

**Docker vs VM**

说起虚拟化，与 VM 的比较是离不开的，我们可以看看下面这张图：



vm 的理念是在宿主的系统之上，自己虚拟了一个硬件平台，然后运行一个不同的 OS。这意味着它要求很多的资源，在一台机器上，你最多就跑几个虚拟机吧。

而 docker 是依托于宿主机提供的内核，仅仅把一个不同的 linux 发行版本所需的特性打包成一个 image, 这样子当你运行一个 ubuntu 镜像，你会感觉就是在一个 ubuntu 的操作环境里，但实际调用的系统接口都是来源于宿主机。所以当你运行一些内核相关的命令时，你就会发现一些端倪，比如 uname -a, 这时给出的信息肯定是宿主机的。

因为共用宿主机的内核，所以 docker 所需的资源也很少，性能开销很小，通常可以在秒内启动，有些已经可以做到毫秒内启动了。在一台机器上，你完全可能做到同时运行上千个 docker 的容器。

docker 的运行方式也让我们对 VM 的理念产生质疑，假如我们需要虚拟一个不同的运行环境，是否需要一个完全新的系统？其实我们要的只是运行一个应用所需的依赖环境，不是吗？

总的来说，docker 抛弃传统 VM 试图模拟完整机器的思路，本着“面向应用”的核心理念，以应用为单元进行”集装封箱”。

**Docker vs LXC**

LXC 是什么？LXC 就是 Linux Container，官网。 LXC 也是一种轻量的虚拟技术，Linux 原生支持的容器。可以说 docker 就是基于 LXC 发展起来的，提供 LXC 的高级封装，发展标准的配置方法。

LXC 的定位是替代传统的虚拟机，侧重于提供一个个操作系统，如 Ubuntu、Debian等。Docker 是面向应用的，官方提倡一个容器即是一个应用，以应用为中心。所以，docker 还提供了统一的打包部署方案，即 Dockerfile, 还有版本控制，image 复用，远程仓库以供镜像共享等。

可以说，LXC 只是 Docker 的底层技术之一，而 Docker 已经在此之上发展出了一个生态系统，如果有另一种容器虚拟技术的话，也许 LXC 也只是配置文件里的 option 选项，libContainer 项目就是在做这样的事情。

**Docker 在解决什么**

看到这里，你应该对 docker 在解决的问题有一些模糊的概念，我们不妨总结一下：

也许以后 docker 成为了新的软件管理模式，我们不会再看到开发人员为每个不同发行版本编写配置文件，处理复杂的系统依赖。无论我们需要什么服务，我们只需要安装 docker，然后 pull 一个镜像到本地就可以了。事实上，我们已经开始看到这种趋势了，毕竟 docker 作为开发和测试来说都太方便了。