

软件复用技术与云

—— Docker容器与云计算

1352958 金敏

过去几年开源界以OpenStack为代表的云计算持续火了好久，这两年突然又冒出一个叫Docker的容器技术，其发展之迅猛远超预料。网上介绍Docker容器的文章已经很多了，用一句话概括：Docker是一种轻量虚拟化的容器技术，提供类似虚拟机的隔离功能，并使用了一种分层的联合文件系统技术管理镜像，能极大简化环境运维过程。Docker容器云则是使用Docker技术打造的一站式容器云服务平台，即CaaS（Containers as a Service）——容器即服务；可以将它简单看作为PaaS的升级版，使用Docker容器技术的CaaS平台功能更强大，使用灵活，部署更方便。Docker容器作为一个全新的概念虽然广受关注，但是对于如何使用Docker，将容器技术真正落地用起来，目前估计很多人还没有非常清晰的思路。我们在这里讨论主要目的就是根据Docker容器的特点，描述Docker容器到底能用来干什么。

可以认为容器云是处于IaaS和PaaS之间的位置，容器编排集成度越高越像PaaS，反之越像IaaS，据此这里简单地将Docker容器的使用分两大类：主机和服务。总而言之，容器云可以认为是虚拟机级别的软件复用技术。

1. 作为云主机使用

相比虚拟机来说，容器使用的是一系列非常轻量级的虚拟化技术，使得其启动、部署、升级跟管理进程一样迅速，用起来灵活又感觉跟虚拟机一样没什么区别，所以有些人直接使用Docker的Ubuntu等镜像创建容器，当作轻量的虚拟机来使用。

特别是现在随着系统、软件越来越多，开发测试环境越来越复杂，仅仅靠多用户共享的这种方式节省资源带来的后果就是环境完全不可控。Docker容器的出现让每个人仅仅通过一个几KB的Dockerfile文件就能构建一个自定义的系统镜像，进而启动一个完整系统容器，让人人都能成为DevOps。

容器云主机也完全能像普通主机一样随意启动、稳定运行、关机、重启，所以在上面随意搭建博客、小网站等完全不在话下。除了常用的托管服务业务，你完全可以自定义任何用法，包括在上面使用任何云服务提供商的云硬盘、云数据库，部署各种你需要的服务。

目前Docker容器管理服务器在Windows下运行需要借助Toolbox工具，虽然微软在2014年底就计划提供Windows Server容器镜像但目前还没有发布，所以想在Docker里跑Windows系统的容器的人还需要等待，希望到时候微软能裁剪出一种轻巧的Windows基础镜像，毕竟容器本身就是一种更轻量级的系统。

2. 作为服务使用

如果你仅仅把Docker容器当作一个轻量的固定虚拟机用，那其实只能算是另类用法，Docker容器最重要价值在于提供一整套平台无关的标准化技术，简化服务的部署、升级、维护，只要把需要运维

的各种服务打包成标准的集装箱，就可以在任何能运行Docker的环境下跑起来，达到开箱即用的效果，这个特点才是Docker容器风靡全球的根本原因。

下面重点列举几个使用Docker容器部署服务的场景。

1. Web应用服务

Web应用服务是使用最广泛的一类服务，典型的架构是前端一个Tomcat + Java服务，后端MySQL数据库。

前端的Java Web服务器是最适合使用Docker容器的，先将Java运行环境、web服务器直接打包成一个通用的基础Docker镜像，之后再将自定义应用代码或编译程序包加入到该基础镜像中就能产生一个新的应用镜像，最后通过Docker服务立马就能以容器的形式启动Web应用服务。

因为web应用程序一般是无状态的，随着业务访问量增减，用同样的镜像新建、销毁容器即可轻松实现伸缩（前面还需配上DNS域名或者负载均衡的服务），例如下图所示。

2. 持续集成和持续部署

互联网行业倡导敏捷开发，持续集成部署CI/CD便是最典型的开发模式。

使用Docker容器云平台，就能实现从代码编写完成推送到git/svn后，自动触发后端CaaS平台将代码下载、编译并构建成测试Docker镜像，再替换测试环境容器服务，自动在Jenkins或者Hudson中运行单元/集成测试，最后测试通过后，马上就能自动将新版本镜像更新到线上，完成服务升级。整个过程全自动化，一气呵成，最大程度地简化了运维成本，而且保证线上、线下环境完全一致，线上服务版本与git/svn发布分支也实现统一。

3. 微服务架构使用

如果说上面两种应用场景还不足以体现出与传统的PaaS平台相比的巨大优势的话，那么对微服务的架构这种复杂又灵活的使用场景的无缝支持绝对具有革命意义。

微服务架构将传统分布式服务继续拆分解耦，形成一些更小服务模块，服务模块之间独立部署升级，这些特性与容器的轻量、高效部署不谋而合。

如下图所示，每个容器里可以使用完全不同环境的镜像服务，容器启动即产生了一个独立的微服务主机节点（独立的网络IP），上层服务与下层服务之间服务发现通过环境变量注入、配置文件挂载等多种方式灵活解决，而且还可以直接将云平台提供的各种云服务与自定义的微服务整合组成一个强大的服务集群。

更重要的是，拥有如此多服务的集群环境迁移、复制也非常轻松，只需选择好各服务对应的Docker服务镜像、配置好相互之间访问地址就能很快搭建出一份完全一样的新集群。

4. 其他应用场景

跑一次性/定时任务

有些情况下用户只是需要执行一次性的任务，例如计算出某个结果即可，如果采用传统的服务模式，服务器需要一直运行，造成的极大的资源浪费。而容器的快速创建、销毁能很灵活满足这种完全按需付费的场景。只要制作好一次性程序运行的Docker镜像，当需要运行的时候实时通过镜像创建出容器来执行任务，程序执行完成容器自动退出释放资源。

当然，容器本身只包含静态的镜像和上层的临时可读写层，运行的数据是不会持久化的，所以任务相关数据可以从云存储、云硬盘或者云端数据库里读写。

科学计算服务

在信息化的今天，除了计算机领域，自然科学等领域也越来越需要借助一些复杂的程序软件来做计算分析，如果还要求他们学习繁杂的服务运维部署就太麻烦了，国外就有科研人员用Docker自行部署IPython Notebook服务器开展数据科学研究的例子。

游戏和网联网等

腾讯曾宣称他们已经将Docker容器用在游戏服务端使用Docker，因为对游戏不了解，具体怎么用的没太关注，有需要的人可以参考下他们的使用思路。

网易蜂巢是网易基于自研IaaS为企业及开发者打造的基于Docker的容器云，提供安全灵活的容器即服务，兼具应用引擎类PaaS服务的便利性和IaaS服务的灵活性，是虚拟主机的进化版本。为满足不同用户需求，如上文所述，目前同时提供容器（主机）和服务两种使用方式，容器可看成是一个独立的主机节点，提供SSH客户端登录实现自由操作；服务则是一组无状态的可随时伸缩的服务组，对外通过网易的负载均衡暴露统一服务入口提供服务。

为何Docker如此特别？

Docker在虚拟的容器环境之上增加一个应用部署引擎。它是一个轻量级但十分强大的关于虚拟化技术的开源容器，在容器中还整合了构建并容器化应用的工作流程。

好处：

Docker可以使你迅速地根据组件组合出应用，并消除代码分发过程中遇到的障碍。例如你可以在同一个主机上运行两个应用相同但版本不同的Docker容器。

Docker使你的代码尽可能快的通过测试并部署到生成环境。

Docker难以置信的简单易用。无论是在简易版Linux、OSX，还是Windows主机上都可以开始学习Docker。（Windows需要直接运行兼容Linux内核或者运行Docker镜像的虚拟机）。

你可以在几分钟之内容器化你的应用。大多数Docker容器在一秒内即可启动。

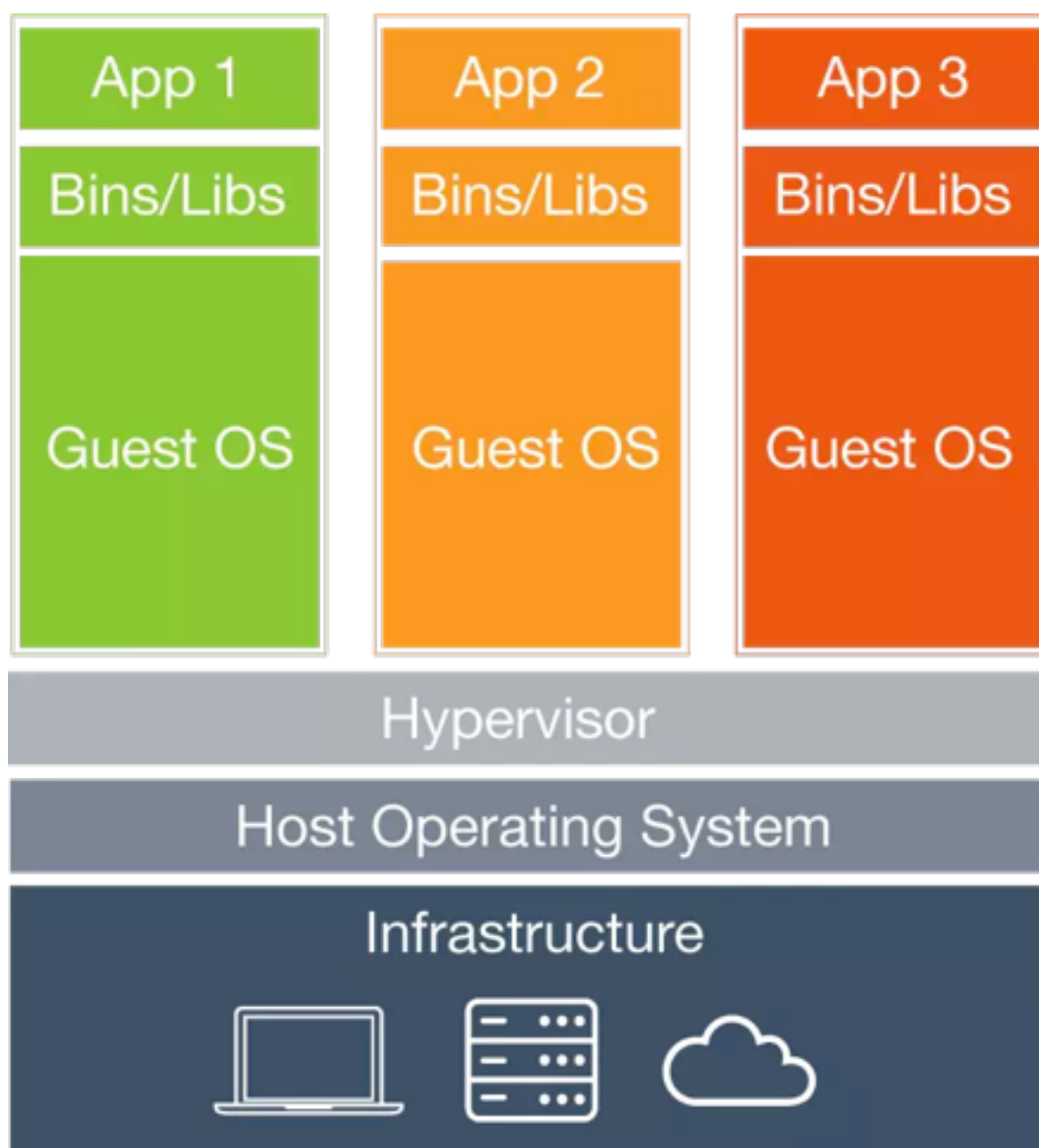
Docker容器几乎可以在任何地方运行。你可以在台式机、物理服务器、虚拟机上部署容器到数据中心，上传到公有云和私有云均可。这样，你就可以在任何地方运行完全一样的容器。

在我们继续讨论Docker惊艳的容器化特征之前，我们先看一下虚拟机和容器的区别。

容器与虚拟机享有相似的资源隔离和分配，但是容器有一套不同的架构方式使得它本身更加便携和高效。我们看下面两个演示二者差别的两个表。

虚拟机

每一个虚拟机包括应用、必要的二进制文件和库、完整的客户操作系统，大小总计数十GB。重要的是，虚拟机需要耗时几分钟来启动，而且对资源敏感，然后容器只需要几秒甚至更少时间启动，在资源较少负载过大时也能应付的了。所以，你可以在本地计算机运行很少的几个虚拟机用于开发目的，但是你可以运行很多很多的容器。这可以使你在开发环境中高效的运行应用环境的完整备份。而且，在生成环境中，这些资源节约很容易转化为成本节约。



容器组件

Docker客户端跟服务器（守护程序）通信，服务器来做所有的工作。Docker通过包含命令行客户端（供Linux、OS X和Windows使用）和完整RESTful接口的二进制文件来分发。你可以在一台Linux主机上运行Docker守护进程和客户端，或者将你的本地Docker客户端连接到另一台Linux主机的远程Docker守护进程。

Docker镜像

Docker镜像是Docker的构建基石。这些镜像是Docker生命周期中的构建部分。他们是一种分层的形式，根据一系列说明一步一步构建产生的统一文件系统。Dockerfile是一个文本文档，它包含用户可以在命令行调用的所有命令从而组成一个镜像。使用一个基本的Docker镜像，可以添加、拷贝文件、执行命令、开放端口。

你可以认为Docker文件是一种源码，而镜像是为产生容器而被编译的代码，而容器则是运行时代码。Docker文件具有高度的便携性，可以随时共享、存储和更新。

注册服务

一旦你构建出一个镜像，你就可以在注册服务中存储它。注册服务有：公共的和私有的。像Docker Hub、Quay.io、Container Registry by Google Cloud的注册服务都有公共和私有的选项。你当然也可以在自己机器上运行一个自己的私有注册服务。

Docker 使你能把你的应用和应用的依赖打包到一个良好的自包含镜像中。然后你可以用该那个镜像来在容器中运行你的应用。问题是，你通常也打包了一些你可能不需要比你需要的更多的东西，最终座椅最重你得到了一个巨大的镜像和巨大的容器。大多数开始使用Docker的人会使用Docker的官方仓库作为他们的语言的选择，但是不幸的是，如果你使用官方镜像，你会得到一个巨大的镜像，而本来你可以得到一个小镜像的。你并不需要和这些官方镜像中一起的许多复杂的东西。这也正是docker于软件复用技术的核心。