复用云技术

云计算的基石是虚拟化技术

容器技术

基本概念

和容器技术比较靠近的是虚拟技术，容器技术从某种程度上来开也是一种虚拟技术。比如我们的虚拟机，在vmware上使用ubuntu的虚拟机来构建一些环境，这时在设备上完全搭建了一个完全不一样的操作系统，但是虚拟技术的效率和资源使用率上面都不是太好，所以才出现了容器技术(Container), 容器技术在操作系统镜像上构建一个容器，那么应用程序可以直接运行在容器中。

特点

- 速度快
- 性能好
- 资源利用率高
- 可移植性好

对云技术的意义

云技术对于虚拟化是非常看重的，因为必须要保证环境的一致性，才能保证云技术的互操作性，可移植性，可复用性。

比如你自己搭建了一个服务，它是依赖于某些操作系统的，如果你需要把这个服务部署到多个服务器上，那么一种比较常用的方法时，在每个服务器上都配置环境，但是这样很容易导致一些问题，比如配置环境的流程不一致，操作系统内核不一样，物理设备不一样，这样可能会到底测试开发环境和最后发布的环境不一样，导致软件的表现不一样。另外一种方法就是直接把你的测试操作系统当做一个虚拟机直接在服务器上运行，但是运行一个虚拟机开销太大，而且版本控制已经移植性其实都不太好。那么，这时我们可以使用容器技术（例如dokcer），在一个镜像（image）比如ubuntu上搭建一个容器，然后提交到共有仓库上docker hub上面，那么，以后我的服务器只要是ubuntu（其实其他linux也可以，只要你的环境不依赖于操作系统的内核），那么我只需要安装一个docker，然后把image pull下来，就可以完美地得到一模一样的环境

实例

Docker

介绍

Docker 是一个开源的应用容器引擎，让开发者可以打包他们的应用以及依赖包到一个可移植的容器中，然后发布到任何流行的 Linux 机器上，也可以实现虚拟化。容器是完全使用沙箱机制，相互之间不会有任何接口。

原理

Docker核心解决的问题是利用LXC来实现类似VM的功能，从而利用更加节省的硬件资源提供给用户更多的计算资源。同VM的方式不同, LXC 其并不是一套硬件虚拟化方法 - 无法归属到全虚拟化、部分虚拟化和半虚拟化中的任意一个，而是一个操作系统级虚拟化方法, 理解起来可能并不像VM那样直观。所以我们从虚拟化到docker要解决的问题出发，看看他是怎么满足用户虚拟化需求的。

用户需要考虑虚拟化方法，尤其是硬件虚拟化方法，需要借助其解决的主要是以下4个问题:

1. 隔离性 - 每个用户实例之间相互隔离, 互不影响。 硬件虚拟化方法给出的方法是VM, LXC给出的方法是container，更细一点是kernel namespace
2. 可配额/可度量 - 每个用户实例可以按需提供其计算资源，所使用的资源可以被计量。硬件虚拟化方法因为虚拟了CPU, memory可以方便实现, LXC则主要是利用cgroups来控制资源
3. 移动性 - 用户的实例可以很方便地复制、移动和重建。硬件虚拟化方法提供snapshot和image来实现, docker(主要)利用AUFS实现
4. 安全性 - 这个话题比较大，这里强调是host主机的角度尽量保护container。硬件虚拟化的方法因为虚拟化的水平比较高，用户进程都是在KVM等虚拟机容器中翻译运行的, 然而对于LXC, 用户的进程是lxc-start进程的子进程, 只是在Kernel的namespace中隔离的, 因此需要一些kernel的patch来保证用户的运行环境不会受到来自host主机的恶意入侵, dotcloud(主要是)利用kernel grsec patch解决的.

一些特点

1. Docker是基于Linux 64bit的，无法在32bit的linux/Windows/unix环境下使用
2. LXC是基于cgroup等linux kernel功能的，因此container的guest系统只能是linux base的
3. 隔离性相比KVM之类的虚拟化方案还是有些欠缺，所有container公用一部分的运行库
4. 网络管理相对简单，主要是基于namespace隔离
5. cgroup的cpu和cpuset提供的cpu功能相比KVM的等虚拟化方案相比难以度量(所以dotcloud主要是按内存收费)
6. docker对disk的管理比较有限
7. container随着用户进程的停止而销毁，container中的log等用户数据不便收集

缺陷

1. 针对特点1-2，有windows base应用的需求的基本可以pass了; 3-5主要是看用户的需求，到底是需要一个container还是一个VM, 同时也决定了docker作为 IaaS 不太可行。
2. Docker并非适合所有应用场景，Docker只能虚拟基于Linux的服务。Windows Azure 服务能够运行 Docker实例，但到目前为止Windows服务还不能被虚拟化。

使用docker的主要步骤

分解

一般来说，应用程序都是复杂的，它们都有很多的组件。例如，大多数应用程序都需要数据库或中间件服务的支持以实现对数据的存储、检索和集成。所以，需要通过设计和部署把这些服务拆分成为它们自己的容器。如果一个应用程序能够被拆分成为越多的分布式组件，那么应用程序扩展的选择则越多。但是，分布式组件越多也意味着管理的复杂性越高。

选择一个基础映像

当执行应用程序迁移时，应尽量避免推倒重来的做法。搜索Docker注册库找到一个基本的Docker映像并将其作为应用程序的基础来使用。

解决安全性和管理问题

安全性和管理应当是一个高优先级的考虑因素；企业用户不应再把它们当作应用程序迁移至容器的最后一步。反之，企业必须从一开始就做好安全性和管理的规划，把它们的功能纳入应用程序的开发过程中，并在应用程序运行过程中积极主动地关注这些方面。这就是企业应当花大功夫的地方。 基于容器的应用程序是分布式应用程序。企业应当更新较老的应用程序以支持联合身份管理方法，这将非常有利于确保分布式应用程序的安全性。为了做到这一点，应为每一个应用程序组件和数据提供一个唯一的标识符，这个标识符可允许企业在一个细粒度的级别上进行安全性管理。企业用户还应当增加一个日志记录的方法。

增加代码

为了创建映像，企业用户需要使用一个Dockerfile来定义映像开发的必要步骤。一旦创建了映像，企业用户就应将其添加至Docer Hub。

配置、测试、部署

应对在容器中运行的应用程序进行配置，以便于让应用程序知道可以在哪里连接外部资源或者应用程序集群中的其他容器。企业用户可以把这些配置部署在容器中或使用环境变量。 对基于容器的应用程序进行测试类似于对其他分布式应用程序的测试。企业可以对每个容器进行组件测试，并将容器集群作为一个整体进行测试。 确定应用程序应如何能够在负载增加的情况下进行扩展。如果用户正在使用一个集群管理器（例如Swarm），则可测试其性能。 最后，把容器部署到实际生产环境中。为了积极主动地关注基于容器的应用程序的运行状况，可考虑实施必要的监控和管理机制 。确保打开日志记录功能。 很多应用程序迁移至云计算都是采用容器技术的。虽然迁移有一点复杂，但是容器可以保护应用程序投资并赋予了它一个更长的使用寿命