

<http://dockone.io>登录 (<http://dockone.io/login/>)注册 (<http://dockone.io/account/register/>)<http://dockone.io/article/2626>Docker (<http://dockone.io/topic/Docker>)

Docker五大优势：持续集成、版本控制、可移植性、隔离性和安全性

【编者的话】随着Docker技术的不断成熟，越来越多的企业开始考虑使用Docker。Docker有很多的优势，本文主要讲述了Docker的五个最重要优势，即持续集成、版本控制、可移植性、隔离性和安全性。

对于Docker，应该不需要进行详细的介绍了。它是最火热的开源项目之一，通过在容器中增加一个抽象层（a layer of abstraction），就可以将应用程序部署到容器中。在看似稳定而成熟的场景下，使用Docker的好处越来越多。在这篇文章中，我不谈论Docker是什么或者Docker是怎么工作的，取而代之，我会提出使用这个不断成长的平台的五大好处。

持续部署与测试

Docker在开发与运维 (<http://iamondemand.com/blog/why-devops-is-a-role-and-not-just-a-concept/>)的世界中具有极大的吸引力，因为它能保持跨环境的一致性。在开发与发布的生命周期中，不同的环境具有细微的不同，这些差异可能是由于不同安装包的版本和依赖关系引起的。然而，Docker可以通过确保从开发到产品发布整个过程环境的一致性来解决这个问题* Docker容器通过相关配置，保持容器内部所有的配置和依赖关系始终不变。最终，你可以在开发到产品发布的整个过程中使用相同的容器来确保没有任何差异或者人工干预。

使用Docker，你还可以确保开发者不需要配置完全相同的产品环境，他们可以在他们自己的系统上通过VirtualBox建立虚拟机来运行Docker容器。Docker

(<http://iamondemand.com/blog/how-ironsource-scales-node-js-with-docker-to-support-millions-of-daily-users/>)的魅力在于它同样可以让你在亚马逊EC2 (<http://iamondemand.com/blog/ec2-virtualization-options-and-the-move-from-pv-to-hvm/>)实例上运行相同的容器。如果你需要在一个产品发布周期中完成一次升级，你可以很容易地将需要变更的东西放到Docker容器中，测试它们，并且使你已经存在的容器执行相同的变更。这种灵活性就是使用Docker的一个主要好处。和标准部署与集成过程一样，Docker可以让你构建、测试和发布镜像，这个镜像可以跨多个服务器进行部署。哪怕安装一个新的安全补丁，整个过程也是一样的。你可以安装补丁，然后测试它，并且将这个补丁发布到产品中。

多云平台

Docker最大的好处之一就是可移植性。在过去的几年里，所有主流的云计算提供商，包括亚马逊AWS和谷歌的GCP，都将Docker融入到他们的平台并增加了各自的支持。Docker容器能运行在亚马逊的EC2实例、谷歌的GCP实例、Rackspace服务器或者VirtualBox这些提供主机操作系统的平台上。举例来说，如果运行在亚马逊EC2实例上的Docker容器能够很容易地移植到其他几个平台上，比如说VirtualBox，并且达到类似的一致性和功能性，那这将允许你从基础设施层中抽象出来。除了AWS和GCP，Docker在其他不同的IaaS提供商也运行的非常好，例如微软的Azure、OpenStack和可以被具有不同配置的管理者所使用的Chef、Puppet、Ansible等。

环境标准化和版本控制

通过上面的讨论，Docker容器可以在不同的开发与产品发布生命周期中确保一致性，进而标准化你的环境。除此之外，Docker容器还可以像git仓库一样，可以让你提交变更到Docker镜像中并通过不同的版本来管理它们。设想如果你因为完成了一个组件的升级而导致你整个环境都损坏了，Docker可以让你轻松地回滚到这个镜像的前一个版本。这整个过程可以在几分钟内完成，如果和虚拟机的备份或者镜像创建流程对比，那Docker算相当快的，它可以让你快速地进行复制和实现冗余。此外，启动Docker就和运行一个进程一样快。

隔离性

Docker可以确保你的应用程序与资源是分隔开的。几个月前，Gartner发表了一篇报告，这份报告说明了运行Docker 容器进行资源隔离的效果和虚拟机（VM）管理程序一样的好，但是管理与控制方面还需要进行完善。

我们考虑这样一个场景，你在你的虚拟机中运行了很多应用程序，这些应用程序包括团队协作软件（例如Confluence）、问题追踪软件（例如JIRA）、集中身份管理系统（例如Crowd）等等。由于这些软件运行在不同的端口上，所以你必须使用Apache或者Nginx来做反向代理。到目前为止，一切都很正常，但是随着你的环境向前推进，你需要在你现有的环境中配置一个内容管理系统（例如Alfresco）。这时候有个问题发生了，这个软件需要一个不同版本的Apache

Tomcat，为了满足这个需求，你只能将你现有的软件迁移到另一个版本的Tomcat上，或者找到适合你现有Tomcat的内容管理系统（Alfresco）版本。

对于上述场景，使用Docker就不用做这些事情了。Docker能够确保每个容器都拥有自己的资源，并且和其他容器是隔离的。你可以用不同的容器来运行使用不同堆栈的应用程序。除此之外，如果你想在服务器上直接删除一些应用程序是比较困难的，因为这样可能引发依赖关系冲突。而Docker可以帮你确保应用程序被完全清除，因为不同的应用程序运行在不同的容器上，如果你不再需要一款应用程序，那你可以简单地通过删除容器来删除这个应用程序，并且在你的宿主操作系统上不会留下任何的临时文件或者配置文件。

除了上述好处，Docker还能确保每个应用程序只使用分配给它的资源（包括CPU、内存和磁盘空间）。一个特殊的软件将不会使用你全部的可用资源，要不然这将导致性能降低，甚至让其他应用程序完全停止工作。

安全性

如上所述，Gartner也承认Docker正在快速地发展。从安全角度来看，Docker确保运行在容器中的应用程序和其他容器中的应用程序是完全分隔与隔离的，在通信流量和管理上赋予你完全的控制权。Docker容器不能窥视运行在其他容器中的进程。从体系结构角度来看，每个容器只使用着自己的资源（从进程到网络堆栈）。

作为坚固安全的一种手段，Docker将宿主机操作系统上的敏感挂载点（例如/proc和/sys）作为只读挂载点，并且使用一种写时复制（<http://zh.wikipedia.org/wiki/%E5%AF%AB%E5%85%A5%E6%99%82%E8%A4%87%E8%A3%BD>）系统来确保容器不能读取其他容器的数据。Docker也限制了宿主机操作系统上的一些系统调用，并且和SELinux与AppArmor一起运行的很好。此外，在Docker Hub上可以使用的Docker镜像都通过数字签名来确保其可靠性。由于Docker容器是隔离的，并且资源是受限制的，所以即使你其中一个应用程序被黑，也不会影响运行在其它Docker容器上的应用程序。

结语

将云计算一起考虑，上面提到的这些好处能够清楚地证明Docker是一个有效的开源平台。使用Docker的好处越来越多，今天我只想强调这前五大好处。如果你使用了Docker，欢迎分享你的使用案例或者任何你觉得使用Docker带来的好处。

原文链接：5 Key Benefits of Docker: CI, Version Control, Portability, Isolation and Security (<http://java.dzone.com/articles/5-key-benefits-docker-ci>) (翻译：肖远昊 校对：魏小红)





4

[分享](#) 2015-05-23<http://dockone.io/people/%E5%AE%88%E5%80%992581314><http://dockone.io/people/%E4%B8%89%E5%A1%98%E6%B0%B4><http://dockone.io/people/fanne>  <http://dockone.io/people/grumpyAmber>

0 条评论

要回复文章请先登录 (<http://dockone.io/login/>)或注册 (<http://dockone.io/account/register/>)

DockOne.io, 专业的容器技术交流平台

关注容器相关的产品以及开源项目

2018 **DockOne**. All Rights Reserved.

DockOne, 新圈子, 新思路, 新视野。

本网站服务器由UCloud云服务 (http://www.ucloud.cn/?utm_source=zanzhu&utm_campaign=DokerOne&utm_medium=display)
提供。



