

[提问](#)[实验台边...](#)[首](#) [话题](#) [发现](#) [消息](#)

245

[« 去我的收藏](#)

## Docker

[添加评论](#) • [编辑](#) • [删除](#)

### 为什么需要Docker?

**Java3y**, 原创技术公众号: Java3y

245 人赞

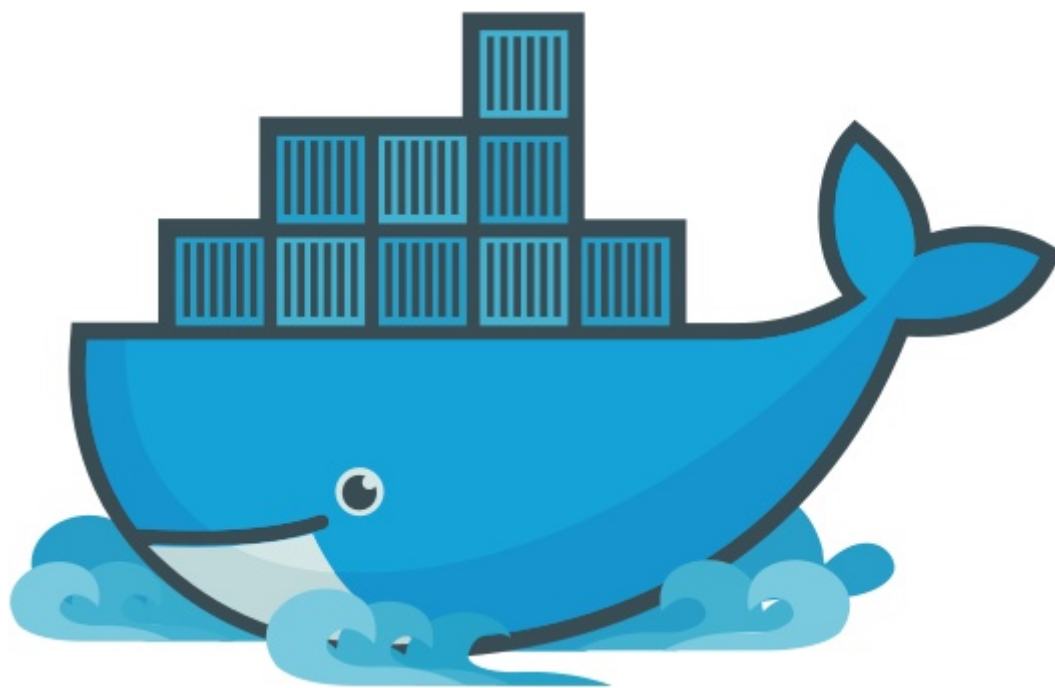
#### 前言

只有光头才能变强。

文本已收录至我的GitHub仓库, 欢迎Star: [github.com/ZhongFuCheng...](https://github.com/ZhongFuCheng...)

估计大家也可能听过Docker这项技术(在论坛上、招聘技能上、交流群上等等), 要是不了解Docker, 都不好意思在网上冲浪的时候吹牛逼了。

所以这几天学了一下Docker, 总结了Docker入门的相关知识, 分享给大家(好让我们一起吹牛逼)。



I need a doctor, call me a doctor. I need a doctor, doctor, to bring me back to life .

#### 一、为什么需要Docker

官方介绍(中文版): [docker-cn.com/what-dock...](https://docker-cn.com/what-dock...)

Docker 是世界领先的软件容器平台。

开发人员利用 Docker 可以消除协作编码时“在我的机器上可正常工作”的问题。

运维人员利用 Docker 可以在隔离容器中并行运行和管理应用，获得更好的计算密度。

企业利用 Docker 可以构建敏捷的软件交付管道，以更快的速度、更高的安全性和可靠的信誉为 Linux 和 Windows Server 应用发布新功能。

245

## 1.1 环境(切换/配置)麻烦

一般我们写程序的，能接触到好几个环境：

- 自己写代码的环境叫做开发环境。
- 给测试去跑的环境叫做测试环境。
- 测试完可以对外使用的叫做生产环境。

其实我们在学习编程中，很多时间都浪费在“环境”上：

- 如果我现在重装了系统，我想要跑我的war/jar包，我得去安装一下JDK、Tomcat、MySQL等配置各种的环境变量才能跑起来。
- 开开心心地跟着博主给出的步骤去写Demo，但总是有Bug。(这里我将版本/依赖也归纳在环境的范畴里边)。
- 好不容易在测试环境下跑起来了，在生产环境就各种出错！
- 跟着教学视频做分布式/集群的项目，跑一堆的虚拟机，每个虚拟机都要安装对应的环境。

所以就有个笑话《千万不要跟程序员说，你的代码有bug》：

- 他的第一反应是你的环境有问题，第二就是你是傻逼不会用吧。
- 你要跟他这么说：“这个程序运行的怎么运行的跟预期不一样，是我操作有问题吗？”。
- 这货就会第一反应“我擦，这是不是出bug了？”

## 1.2 应用之间需要隔离

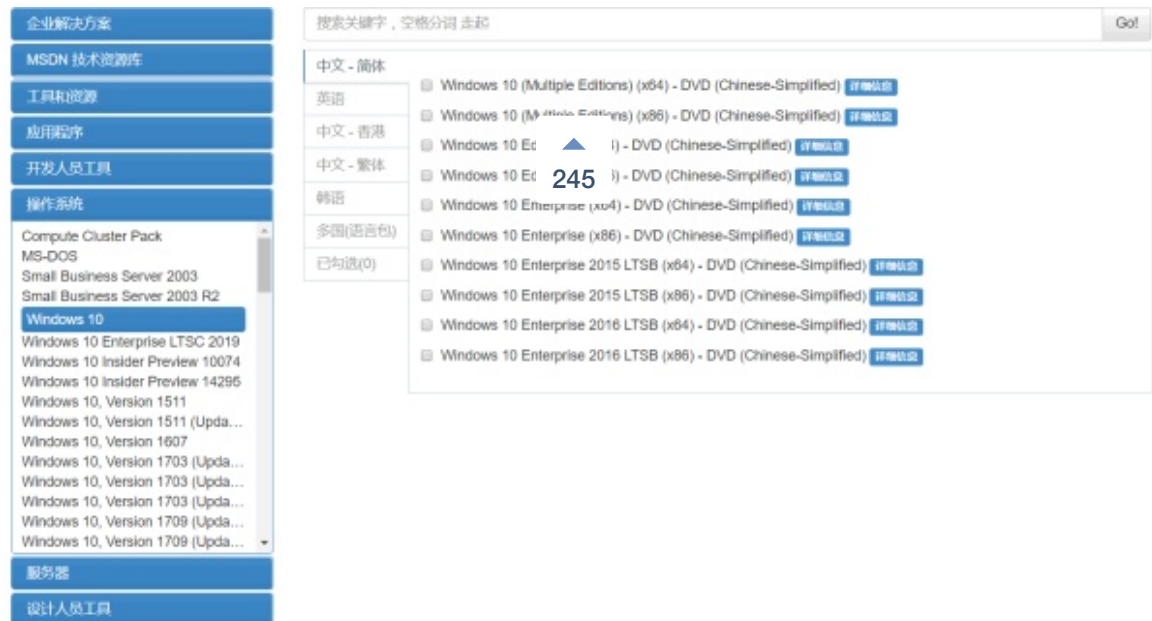
比如我写了两个应用(网站)，这两个应用部署在同一台服务器上，那可能会出现什么问题？

- 如果一个应用出现了问题，导致CPU占100%。那另一个应用也会受到关联，跟着一起凉凉了。
- 这两个应用是完全不同技术栈的应用，比如一个PHP，一个.NET。这两个应用各种的依赖软件都安装在同一个服务器上，可能就会造成各种冲突/无法兼容，这可能调试就非常麻烦了。

## 二、Docker是如何解决上述的问题的

### 2.1 解决环境(切换/配置)

不知道大家有没有装过系统，比如说装Linux虚拟机，重装Windows系统，都是需要镜像的。



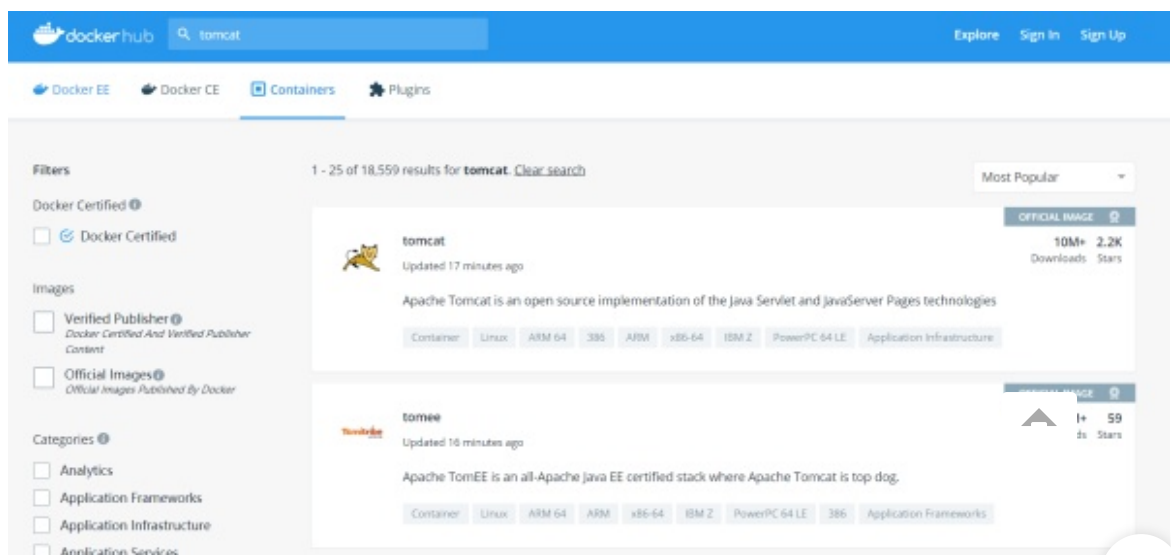
有了这个镜像，我们就可以运行这个镜像，来进行安装系统的操作(此处省略N个下一步)，于是我们的系统就装好了。一般来说，我们去官方渠道下载的镜像，都是纯净的。比如去官方下载Windows镜像，装完后之后桌面只有一个回收站。

但有了解装系统的同学可能就会知道，有的镜像装完可能还有360这些软件，但系统的的确是变了。简单来说，就是这些镜像添加了其他的东西(比如360软件、腾讯、千千静听等等软件)。

Docker也是这种思路，可以将我们的想要的环境构建(打包)成一个镜像，然后我们可以推送(发布)到网上去。想要用这个环境的时候，在网上拉取一份就好了。

有了Docker，我们在搭环境的时候，跟以前的方式就不一样了。

- 之前：在开发环境构建出了一个war包，想跑到Linux下运行。我们得先在Linux下载好Java、Tomcat、MySQL，配置好对应的环境变量，将war包丢到Tomcat的webapps文件夹下，才能跑起来。
- 现在：在Linux下直接拉取一份镜像(各种环境都配好了)，将镜像运行起来，把war包丢进去就好了。



将Docker的镜像运行起来就是一两秒的事情而已，十分方便的。

## 2.2解决应用之间隔离

说到这里，就得提出一个大家可能不认识的概念：LXC (Containers)--->Linux容器。

### 2.2.1Linux容器

在Linux内核中，提供了**cgroups**功能，来达成资源的区隔化。它同时也提供了名称空间(namespace)区隔化的功能，使应用程序看到的操作系统环境被区隔成独立区间，包括进程树，网络，用户id，以及挂载的文件系统。

简单来说就是：LXC是一个为Linux内核包含特征的用户接口。通过强大的API和简单的工具，它可以让Linux用户轻松的创建和托管系统或者应用程序容器。

### 2.2.2回到Docker

我们在翻看Docker的官方文档的时候，也很容易看见cgroup和namespace这两个名词：

#### The underlying technology

Docker is written in Go and takes advantage of several features of the Linux kernel to deliver its functionality.

##### Namespaces

Docker uses a technology called **namespaces** to provide the isolated workspace called the *container*. When you run a container, Docker creates a set of *namespaces* for that container.

These namespaces provide a layer of isolation. Each aspect of a container runs in a separate namespace and its access is limited to that namespace.

Docker Engine uses namespaces such as the following on Linux:

- The **pid namespace**: Process isolation (PID: Process ID).
- The **net namespace**: Managing network interfaces (NET: Networking).
- The **ipc namespace**: Managing access to IPC resources (IPC: InterProcess Communication).
- The **mnt namespace**: Managing filesystem mount points (MNT: Mount).
- The **uts namespace**: Isolating kernel and version identifiers. (UTS: Unix Timesharing System).

##### Control groups

Docker Engine on Linux also relies on another technology called *control groups* ( **cgroups** ). A cgroup limits an application to a specific set of resources. Control groups allow Docker Engine to share available hardware resources to containers and optionally enforce limits and constraints. For example, you can limit the memory available to a specific container.

来源维基百科：

Early versions of Docker used LXC as the container execution driver, though LXC was made optional in v0.9 and support was dropped in Docker v1.10.

lxc是早期版本docker的一个基础组件，docker 主要用到了它对 Cgroup 和 Namespace 两个内核特性的控制。新的Docker版本已经移除了对LXC的支持。

### 2.2.3Docker在Windows和Mac

上面说了，Docker底层用的Linux的cgroup和namespace这两项技术来实现应用隔离，那Windows和Mac用户能用Docker吗？

- 之前，Windows和Mac使用Docker实际上就是跑了一层Linux虚拟机。
  - 比如在Windows下安装的是Docker Toolbox，它需要Oracle Virtual Box来跑Docker

- 现在，Windows和Mac都已经原生支持Docker了。但需要一些安装的条件，详情可以查看官网
  - 比如Windows: Docker for Windows requires 64bit Windows 10 Pro and Microsoft Hyper-V

▲

245

参考资料：

- Windows 原生 Docker 正式商用
  - [blog.daocloud.io/window...](https://blog.daocloud.io/window...)

三、虚拟机和Docker

说到应用隔离和镜像，我就想起了虚拟机。今年下半年(此处省略.....)，文体两开花(此处省略.....)，要是我写文章写得不好，我是需要向XX谢罪的。

估计大家都用过虚拟机，虚拟机也能实现对应用的隔离，安装特定的镜像也能跑出我们想要的环境。虚拟机已经发展了很久了，为什么我们还需要Docker呢？

这部分内容在官网也有相关的介绍：

- [docker-cn.com/what-cont...](https://docker-cn.com/what-cont...)



一句话总结：Docker容器比虚拟机轻量多了！

最后

Docker可以干嘛？

- 将一整套环境打包封装成镜像，无需重复配置环境，解决环境带来的种种问题。
- Docker容器间是进程隔离的，谁也不会影响谁。

其实这篇文章主要是讲为什么我们需要Docker(在学习一项技术之前，必须要知道这项技术是用干什么的)，Docker的一些概念和命令我还没介绍(留到下一篇啦)。如果还没看过【生活现场】从搬家到容器技术docker应用场景解析，可以先去看看~

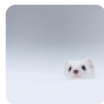
我在学习Docker的时候也找到了不少的资源，想要获取Docker入门资源的同学可在公众号下回复“Docker”

乐于分享和输出干货的Java技术公众号：**Java3y**。关注即可领取海量的视频资源！

觉得我的文章写得不错，不妨点一下赞！

关注专栏 16 条评论 分享 · 举报 · 去文章 245

## 关于创建者



实验台边吃早饭

#来自墨尔本的追梦者#五颗柠檬...

## 收藏夹状态

最近活动于 2019-03-04

0 人关注了该收藏夹

举报收藏夹

## 热门收藏

换一换

程序员的工作

3993 人关注 · 209 条内容

有营养的下饭菜

104 人关注 · 38 条内容

育儿、生活经验感悟

117 人关注 · 235 条内容

收藏了我觉得有意思的东西

321 人关注 · 118 条内容

五叔购物回答辑〔公众号：拾无缺〕

40757 人关注 · 153 条内容