毋庸置疑，容器已经成为企业IT基础设施中必不可少的部分，它具有许多的优点，比如：

第一：容器是不可变的——操作系统，库版本，配置，文件夹和应用程序都包装在容器内。你保证在质量检查中测试过的同一镜像将以相同的行为到达生产环境。

第二：容器很轻——容器的内存占用量很小。容器将只为主要进程分配内存，而不是数百或数千MB。

第三：容器非常快——可以像启动典型Linux进程一样快地启动容器。你可以在几秒钟内启动一个新容器，而不是几分钟。

但是，许多用户仍然像对待典型虚拟机一样对待容器，而忘记了容器具有重要的特征：即容器是一次性的。

这种特征迫使用户改变他们对如何处理和管理容器的看法。那么该如何保持容器的最佳效益呢？以下将介绍Docker容器中应避免的10件事。

1、不要将数据存储在容器中

因为你可以停止，销毁或更换容器。在容器中运行的应用程序版本1.0应该容易地由版本1.1替换，而不会造成任何影响或数据丢失。因此，如果需要存储数据，请批量存储。在这种情况下，还应该注意两个容器是否在同一卷上写入数据，因为这可能会导致损坏。确保你的应用程序是为了写入共享数据存储。

2、不要将应用程序分为两部分进行交付

有些人看到像虚拟机这样的容器，大多数人倾向于认为他们应该将应用程序部署到现有的运行容器中。在开发阶段，你需要不断进行部署和调试，这是正确的。但对于一个连续传递(CD)管道QA和Production，你的应用程序应该是镜像的一部分。

3、不要创建大镜像

因为大镜像将很难分发。确保仅具有运行应用程序/进程所需的文件和库。不要安装不必要的软件包或运行将许多文件下载到新镜像层的“更新” 。

4、不要使用单层镜像

为了有效利用分层文件系统，请始终为操作系统创建自己的基础镜像层，为用户名定义创建另一层，为运行时安装创建另一层，为配置创建另一层，最后是应用程序的另一层。重新创建，管理和分发镜像将更加容易。

5、不要从正在运行的容器中创建镜像

换句话说，不要使用“ docker commit”来创建镜像。这种创建镜像的方法不可复制，应完全避免。始终使用完全可复制的Dockerfile或任何其他S2I(从源到镜像)方法，如果将Dockerfile存储在源代码控制存储库(git)中，则可以跟踪对Dockerfile的更改。

6、不要只使用“最新”标签

对于Maven用户，最新标签就像“ SNAPSHOT”一样。由于容器的分层文件系统性质，因此鼓励使用标签。几个月后生成镜像并发现你的应用程序无法运行是因为父层(Dockerfile中的FROM)被不兼容向后的新版本或错误的新版本所取代，你不会感到惊讶从构建缓存中检索了“最新”版本。在生产环境中部署容器时，也应避免使用“最新”标签，因为你无法跟踪正在运行哪个版本的镜像。

7、不要在单个容器中运行多个进程。

容器非常适合运行单个进程(http守护程序，应用程序服务器，数据库)，但是如果有多个进程，则管理起来可能会遇到更多麻烦，检索日志，并分别更新流程。

8、不要将凭据存储在镜像中。

使用环境变量，你不想对镜像中的任何用户名/密码进行硬编码。使用环境变量从容器外部检索该信息。这个原理的一个很好的例子是Postgres镜像。

9、不要以root用户身份运行进程

“默认情况下，docker容器以root用户身份运行。随着docker的成熟，可能会提供更多安全的默认选项。目前，要求root用户对其他人是危险的，可能并非在所有环境中都可用。你的镜像应使用USER指令为运行容器指定一个非root用户。

10、不要依赖IP地址

每个容器都有自己的内部IP地址，如果你启动和停止容器，它可能会更改。如果应用程序或微服务需要与另一个容器通信，请使用环境变量将正确的主机名和端口从一个容器传递到另一个容器。

所以，在Docker容器的应用中，这个10个坏习惯要不得，注意！注意！再注意！