

容器

容器实现

NameSpace

缺陷

首先，既然容器只是运行在宿主机上的一种特殊的进程，那么多个容器之间使用的就还是同一个宿主机的操作系统内核

优点

敏捷

高性能

作用

隔离

Cgroups

作用

限制

Change Root

作用

容器镜像生效，以实现环境一致性。所谓容器镜像，本质就是容器的根文件系统(rootfs)。

一个正在运行的 Docker 容器，其实就是一个启用了多个 Linux Namespace 的应用进程，而这个进程能够使用的资源量，则受 Cgroups 配置的限制。容器是一个“单进程”模型

镜像

本质

挂载在容器根目录上、用来为容器进程提供隔离后执行环境的文件系统，就是所谓的“容器镜像”。它还有一个更为专业的名字，叫作：rootfs（根文件系统）。但并不包括内核

一致性

通过多个增量 rootfs（UnionFS）实现分层镜像

overlay2文件系统

LowerDir

容器镜像层

MergedDir

容器挂载点，lowerdir和upperdir整合起来提供统一的视图给容器，作为根文件系统；

UpperDir

指向容器层，在容器中创建文件后，文件出现在此目录；

WorkDir

用于实现copy_up操作。

即下层已经生成的永远不会改变，所有的修改都通过在上层叠加

进入容器

docker exec

原理

一个进程，可以选择加入到某个进程已有的 Namespace 当中，从而达到“进入”这个进程所在容器的目的，这正是 docker exec 的实现原理。意味着docker exec 每次都会创建一个和容器共享namespace的新进程