
云原生训练营 | Docker 常见面试题汇总

1. Docker 和虚拟机有啥不同？



答：Docker 是轻量级的沙盒，在其中运行的只是应用，虚拟机里面还有额外的系统。

2. Docker 安全么？

答：Docker 利用了 Linux 内核中很多安全特性来保证不同容器之间的隔离，并且通过签名机制来对镜像进行验证。大量生产环境的部署证明，Docker 虽然隔离性无法与虚拟机相比，但仍然具有极高的安全性。

3. 如何清理后台停止的容器？

答：可以使用 `sudo docker rm $(sudo docker ps -a -q)` 命令。

4. 如何查看镜像支持的环境变量？

答：可以使用 `docker run IMAGE env` 命令。

5. 当启动容器的时候提示：exec format error？如何解决问题

答：检查启动命令是否有可执行权限，进入容器手工运行脚本进行排查。

6. 本地的镜像文件都存放在哪里？

答：与 Docker 相关的本地资源都存放在/var/lib/docker/目录下，其中 container 目录存放容器信息，graph 目录存放镜像信息，aufs 目录下存放具体的内容文件。

7. 如何退出一个镜像的 bash，而不终止它？

答：按 Ctrl-p Ctrl-q。

8. 退出容器时候自动删除？

答：使用 -rm 选项，例如 `sudo docker run -rm -it ubuntu`

9. 怎么快速查看本地的镜像和容器？

答：可以通过 `docker images` 来快速查看本地镜像；通过 `docker ps -a` 快速查看本地容器。

镜像相关：

1. 如何批量清理临时镜像文件？

答：可以使用 `sudo docker rmi $(sudo docker images -q -f dangling=true)`命令

2. 如何查看镜像支持的环境变量？

答：使用 `sudo docker run IMAGE env`

3. 本地的镜像文件都存放在哪里

答：于 Docker 相关的本地资源存放在 `/var/lib/docker/` 目录下，其中 `container` 目录存放容器信息，`graph` 目录存放镜像信息，`aufs` 目录下存放具体的镜像底层文件。

4. 构建 Docker 镜像应该遵循哪些原则？

答：整体远侧上，尽量保持镜像功能的明确和内容的精简，要点包括：

- 尽量选取满足需求但较小的基础系统镜像，建议选择 `debian:wheezy` 镜像，仅有 86MB 大小
- 清理编译生成文件、安装包的缓存等临时文件
- 安装各个软件时候要指定准确的版本号，并避免引入不需要的依赖
- 从安全的角度考虑，应用尽量使用系统的库和依赖
- 使用 Dockerfile 创建镜像时候要添加 `.dockerignore` 文件或使用干净的工作目录

容器相关

1. 容器退出后，通过 `docker ps` 命令查看不到，数据会丢失么？

答：容器退出后会处于终止（`exited`）状态，此时可以通过 `docker ps -a` 查看，其中数据不会丢失，还可以通过 `docker start` 来启动，只有删除容器才会清除数据。

2. 如何停止所有正在运行的容器？

答：使用 `docker kill $(sudo docker ps -q)`

3. 如何清理批量后台停止的容器？

答：使用 `docker rm $(sudo docker ps -a -q)`

4. 如何临时退出一个正在交互的容器的终端，而不终止它？

答：按 `Ctrl+p`，后按 `Ctrl+q`，如果按 `Ctrl+c` 会使容器内的应用进程终止，进而会使容器终止。

5. 很多应用容器都是默认后台运行的，怎么查看它们的输出和日志信息？

答：使用 `docker logs`，后面跟容器的名称或者 ID 信息

6. 使用 `docker port` 命令映射容器的端口时，系统报错 `Error: No public port '80' published for ...`，是什么意思？

答：创建镜像时 `Dockerfile` 要指定正确的 `EXPOSE` 的端口，容器启动时指定 `PublishAllport=true`

7. 可以在一个容器中同时运行多个应用进程吗？

答：一般不推荐在同一个容器内运行多个应用进程，如果有类似需求，可以通过额外的进程管理机制，比如 `supervisord` 来管理所运行的进程

8. 如何控制容器占用系统资源（CPU，内存）的份额？

答：在使用 `docker create` 命令创建容器或使用 `docker run` 创建并运行容器的时候，可以使用 `-c|--cpu-shares[=0]` 参数来调整同期使用 CPU 的权重，使用 `-m|--memory` 参数来调整容器使用内存的大小。

仓库相关

1. 仓库 (Repository) 、注册服务器 (Registry) 、注册索引 (Index) 有何关系 ?

答 : 首先 , 仓库是存放一组关联镜像的集合 , 比如同一个应用的不同版本的镜像 , 注册服务器是存放实际的镜像的地方 , 注册索引则负责维护用户的账号 , 权限 , 搜索 , 标签等管理。注册服务器利用注册索引来实现认证等管理。

2. 从非官方仓库 (如 : <http://dl.dockerpool.com>) 下载镜像的时候 , 有时候会提示 "Error : Invalid registry endpoint <https://dl.docker.com:5000/v1/...> " ?

答 : Docker 自 1.3.0 版本往后以来 , 加强了对镜像安全性的验证 , 需要手动添加对非官方仓库的信任。

`DOCKER_OPTS=" -insecure-registry dl.dockerpool.com:5000"`

重启 docker 服务

配置相关

1. Docker 的配置文件放在那里。如何修改配置 ?

答 : Ubuntu 系统下 Docker 的配置文件是 `/etc/default/docker` , CentOS 系统配置文件存放在 `/etc/sysconfig/docker`

2. 如何更改 Docker 的默认存储设置 ?

答：Docker 的默认存放位置是/var/lib/docker,如果希望将 Docker 的本地文件存储到其他分区，可以使用 Linux 软连接的方式来做。

Docker 与虚拟化

1. Docker 与 LXC (Linux Container) 有何不同？

答：LXC 利用 Linux 上相关技术实现容器，Docker 则在如下的几个方面进行了改进：

- 移植性：通过抽象容器配置，容器可以实现一个平台移植到另一个平台；
- 镜像系统：基于 AUFS 的镜像系统为容器的分发带来了很多的便利，同时共同的镜像层只需要存储一份，实现高效率的存储；
- 版本管理：类似于 GIT 的版本管理理念，用户可以更方面的创建、管理镜像文件；
- 仓库系统：仓库系统大大降低了镜像的分发和管理的成本；
- 周边工具：各种现有的工具（配置管理、云平台）对 Docker 的支持，以及基于 Docker 的 Pass、CI 等系统，让 Docker 的应用更加方便和多样化。

2. Docker 与 Vagrant 有何不同？

答：两者的定位完全不同

Vagrant 类似于 Boot2Docker（一款运行 Docker 的最小内核），是一套虚拟机的管理环境，Vagrant 可以在多种系统上和虚拟机软件中运行，可以在 Windows。Mac 等非 Linux 平台上为 Docker 支持，自身具有较好的包装性和移植性。

原生 Docker 自身只能运行在 Linux 平台上，但启动和运行的性能都比虚拟机要快，往往更适合快速开发和部署应用的场景。

3. 开发环境中 Docker 与 Vagrant 该如何选择？

答：Docker 不是虚拟机，而是进程隔离，对于资源的消耗很少，单一开发环境下

Vagrant 是虚拟机上的封装，虚拟机本身会消耗资源。

Other FAQ

1. Docker 能在非 Linux 平台（Windows+MacOS）上运行吗？

答：可以

2. 如何将一台宿主机的 docker 环境迁移到另外一台宿主机？

答：停止 Docker 服务，将整个 docker 存储文件复制到另外一台宿主机上，然后调整另外一台宿主机的配置即可

3. Docker 容器创建后，删除了/var/run/netns 目录下的网络名字空间文件，可以

手动恢复它：—— 训练营 ——

答：查看容器进程 ID，比如 1234 `sudo docker inspect --format '{{. State.pid}}'`

`$container_id 1234 #` 到 `proc` 目录下，把对应的网络名字空间文件链接到

`/var/run/netns`，然后通过正常的系统命令查看操作容器的名字空间。