



链滴

如何限制 docker run 容器执行时长?

作者: [88250](#)

原文链接: <https://ld246.com/article/1558229770509>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

需求背景

有的时候我们需要跑一些不一定安全（不受信任）的程序时，可以通过 docker 实现资源隔离，其执行时长也需要控制，如果执行时间太长就终止运行。

Docker 本身是否支持超时控制

从 2013 年开始每隔一段时间都有人呼吁 docker 官方来实现这个特性，但官方并不打算实现。

具体讨论细节请看 <https://github.com/moby/moby/issues/1905>。

错误的方案

以上讨论中以及其他地方陆续有人提出“外围”解决方案，其中有两种常见的错误方案值得我们注意。

kill docker run

`timeout 3 docker run ... untrusted_program`

docker run 本身是一个 docker client 派生的进程，使用 kill docker run 进程来终止容器进程运行（比如通过 Linux 命令 `timeout` 或者类似发送 kill 到 docker run 进程）的方案都可能会有问题，导致容器进程无法终止。主要原因是：

- docker run 这个进程不一定能及时响应信号，即使响应了信号，也不一定能传到容器进程，比如 SIGKILL 就没法传递
- 即使通过 docker run 进程将信号传给了容器进程，但容器进程也不一定响应该信号，比如非 SIGKILL 信号会被忽略

docker run kill

`docker run ... timeout 3 untrusted_program`

在容器进程执行前进行一定包装处理，比如还是通过 `timeout` 来控制执行时间。看上去没啥问题，但实际上这个做法非常不安全，因为这个不受信程序可以派生其他进程甚至修改已有的进程空间导致包装失效。

正确的解决方案

限制 `docker run` 执行超时唯一可靠的方法是通过 docker 容器操作命令（`stop/kill/rm -f` 等）来发送 SIGKILL 信号，这样 SIGKILL 信号会直接发到容器进程上。

当然，最终容器进程能不能被终止还要看进程所处状态，比如处于 D 状态就无法终止。