



链滴

7、容器

作者: [sunjvhui](#)

原文链接: <https://ld246.com/article/1562825323182>

来源网站: [链滴](#)

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



新建并启动容器

所需要的命令主要为

`docker run`

例如：输出hello world后终止容器

```
$ docker run ubuntu:18.04 /bin/echo 'Hello world'  
Hello world
```

启动一个 bash 终端，允许用户进行交互

```
$ docker run -t -i ubuntu:18.04 /bin/bash  
root@af8bae53bdd3:/#
```

- `-t` 选项让Docker分配一个伪终端（pseudo-tty）并绑定到容器的标准输入上
- `-i` 则让容器的标准输入保持打开

当docker run 创建容器的时候，docker后台操作标准

- 检查本地是否有指定镜像，没有就去共有仓库下载
- 利用镜像创建并启动一个容器
- 分配一个文件系统，并在只读的镜像层外面挂载一层可读写层
- 从宿主主机配置的网桥接口中桥接一个虚拟接口到容器中
- 从地址池配置一个ip地址给容器

- 执行用户指定的应用程序
- 执行完毕后容器被终止

启动已终止容器

`docker container start`

将一个已经终止的容器启动运行。

后台运行

需要让 Docker 在后台运行而不是直接把执行命令的结果输出在当前宿主机下。此时，可以通过添加 `-d` 参数来实现。

如果使用了 `-d` 参数运行容器。此时容器会在后台运行并不会把输出的结果 (STDOUT) 打印到宿主机面(输出结果可以用 `docker logs`查看)。

```
$ docker run -d ubuntu:18.04 /bin/sh -c "while true; do echo hello world; sleep 1; done"
77b2dc01fe0f3f1265df143181e7b9af5e05279a884f4776ee75350ea9d8017a
```

查看容器信息

`docker container ls`

获取容器的输出信息 `docker container logs`

```
$ docker container logs [container ID or NAMES]
hello world
hello world
hello world
...
```

终止容器

`docker container stop` 来终止一个运行中的容器。

终止状态的容器可以用 `docker container ls -a` 命令看到

```
docker container ls -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS
PORTS         NAMES
ba267838cc1b  ubuntu:18.04  "/bin/bash"            30 minutes ago Exited (0) Abo
t a minute ago          trusting_newton
98e5efa7d997   training/webapp:latest "python app.py"       About an hour ago Exited (
) 34 minutes ago

backstabbing_pike
```

处于终止状态的容器，可以通过 `docker container start` 命令来重新启动。

此外，`docker container restart` 命令会将一个运行态的容器终止，然后再重新启动它。

进入容器

在使用 `-d` 参数时，容器启动后会进入后台。

attach 命令

下面示例如何使用 `docker attach` 命令。

```
$ docker run -dit ubuntu
243c32535da7d142fb0e6df616a3c3ada0b8ab417937c853a9e1c251f499f550

$ docker container ls
CONTAINER ID   IMAGE                COMMAND                  CREATED          STATUS          PO
TS            NAMES
243c32535da7   ubuntu:latest       "/bin/bash"            18 seconds ago  Up 17 seconds
nostalgic_hypatia

$ docker attach 243c
root@243c32535da7:/#
```

如果从这个 `stdin` 中 `exit`，会导致容器的停止。

exec 命令

```
$ docker run -dit ubuntu
69d137adef7a8a689cbcb059e94da5489d3cddd240ff675c640c8d96e84fe1f6

$ docker container ls
CONTAINER ID   IMAGE                COMMAND                  CREATED          STATUS          PO
TS            NAMES
69d137adef7a   ubuntu:latest       "/bin/bash"            18 seconds ago  Up 17 seconds
zealous_swirles

$ docker exec -i 69d1 bash
ls
bin
boot
dev
...

$ docker exec -it 69d1 bash
root@69d137adef7a:/#
```

只用 `-i` 参数时，由于没有分配伪终端，界面没有我们熟悉的 Linux 命令提示符，但命令执行结果仍可以返回。 `
`

当 `-i -t` 参数一起使用时，则可以看到我们熟悉的 Linux 命令提示符。 `
`

如果从这个 `stdin` 中 `exit`，不会导致容器的停止

导出和导入容器

导出容器

如果要导出本地某个容器，可以使用 `docker export` 命令。

```
$ docker container ls -a
CONTAINER ID   IMAGE          COMMAND        CREATED        STATUS
PORTS         NAMES
7691a814370e  ubuntu:18.04  "/bin/bash"   36 hours ago  Exited (0) 21 hours a
o             test
$ docker export 7691a814370e > ubuntu.tar
```

导入容器快照

可以使用 `docker import` 从容器快照文件中再导入为镜像，例如

```
$ cat ubuntu.tar | docker import - test/ubuntu:v1.0
$ docker image ls
REPOSITORY    TAG          IMAGE ID      CREATED        VIRTUAL SIZE
test/ubuntu   v1.0        9d37a6082e97 About a minute ago  171.3 MB
```

此外，也可以通过指定 URL 或者某个目录来导入

```
$ docker import http://example.com/exampleimage.tgz example/imagerepo
```

删除容器

可以使用 `docker container rm` 来删除一个处于终止状态的容器。例如

```
$ docker container rm trusting_newton
trusting_newton
```

如果要删除一个运行中的容器，可以添加 `-f` 参数。Docker 会发送 `SIGKILL` 信号给容器。

清理所有处于终止状态的容器

用 `docker container ls -a` 命令可以查看所有已经创建的包括终止状态的容器，如果数量太多要一个删除可能会很麻烦，用下面的命令可以清理掉所有处于终止状态的容器。

```
$ docker container prune
```