

Docker入门教程（二）命令

【编者的话】DockerOne组织翻译了Flux7的Docker入门教程，本文是系列入门教程的第二篇，介绍了Docker的基本命令以及命令的用法和功能。

在[Docker系列教程的第一篇文章](#)中，我们了解了Docker的基础知识，知道了它是如何工作以及如何安装的。在这篇文章中，我们将学习15个Docker命令，并通过实践来学习它是如何工作的。

首先，让我们通过下面的命令来检查Docker的安装是否正确：

```
docker info
```

如果没有找到这条命令，则表示Docker安装错误。如果安装正确，则会输出类似下面的内容：

```
docker@boot2docker:~$ docker info
Containers: 0
Images: 0
Storage Driver: aufs
  Root Dir: /var/lib/docker/aufs
  Dirs: 0
Execution Driver: native-0.2
Kernel Version: 3.13.3-tingycore64
Debug mode (server): true
Debug mode (client): false
Fds: 10
Goroutines: 12
EventsListeners: 0
Init Path: /usr/local/bin/docker
```

到这一步Docker里还没有镜像或是容器。所以，让我们通过使用命令预先构建的镜像来创建来一个：

```
sudo docker pull busybox
```

```
docker@boot2docker:~$ sudo docker pull busybox
Pulling repository busybox
2d8e5b282c81: Pulling image (latest) from busybox, endpoint: https://cdn-registr
2d8e5b282c81: Download complete
511136ea3c5a: Download complete
6c991eb93460: Download complete
9f4e93171ec5: Download complete
docker@boot2docker:~$ _
```

BusyBox是一个最小的Linux系统，它提供了该系统的主要功能，不包含一些与GNU相关的功能和选项。

下一步我们将运行一个“Hello World”的例子，我们暂且叫它“Hello Docker”吧。

```
docker run busybox /bin/echo Hello Docker
```

```
docker@boot2docker:~$ docker run busybox /bin/echo Hello Docker
Hello Docker
docker@boot2docker:~$ _
```

现在，让我们以后台进程的方式运行 `hello docker`：

```
sample_job=$(docker run -d busybox /bin/sh -c "while true; do echo Docker; sleep 1; done")
```

```
docker@boot2docker:~$ sample_job=$(docker run -d busybox /bin/sh -c "while true;
do echo Docker; sleep 1; done")
docker@boot2docker:~$ docker logs $sample_job
Docker
Docker
Docker
Docker
Docker
Docker
docker@boot2docker:~$ _
```

`sample_job` 命令会隔一秒打印一次 Docker，使用 `docker logs` 可以查看输出的结果。如果没有给这个 job 起名字，那这个 job 就会被分配一个 id，以后使用命令例如 `docker logs` 查看日志就会变得比较麻烦。

运行 `docker logs` 命令来查看 job 的当前状态：

```
docker logs $sample_job
```

所有 Docker 命令可以用以下命令查看：

```
docker help
```

名为 `sample_job` 的容器，可以使用以下命令来停止：

```
docker stop $sample_job
```

使用以下命令可以重新启动该容器：

```
docker restart $sample_job
```

如果要完全移除容器，需要先将该容器停止，然后才能移除。像这样：

```
docker stop $sample_job
docker rm $sample_job
```

将容器的状态保存为镜像，使用以下命令：

```
docker commit $sample_job job1
```

注意，镜像名称只能取字符[a-z]和数字[0-9]。

现在，你就可以使用以下命令查看所有镜像的列表：

```
docker images
```

在[我们之前的Docker教程](#)中，我们学习过镜像存储在Docker registry。在registry中的镜像可以使用以下命令查找到：

```
docker search (image-name)
```

查看镜像的历史版本可以执行以下命令：

```
docker history (image_name)
```

最后，使用以下命令将镜像推送到registry：

```
docker push (image_name)
```

非常重要的一点是，你必须要知道存储库不是根存储库，它应该使用此格式(user)/(repo_name)。

这都是一些非常基本的Docker命令。在我们[Docker教程系列的第六章](#)，我们将讨论如何使用Docker运行Python的Web应用程序，以及一些进阶的Docker命令。

原文链接：[Part 2: The 15 Commands](#)（翻译：[田浩浩](#) 审校：李颖杰）

=====

译者介绍

田浩浩，[悉尼大学USYD](#)硕士研究生，目前在珠海从事Android应用开发工作。业余时间专注Docker的学习与研究，希望通过[DockerOne](#)把最新最优秀的译文贡献给大家，与读者一起畅游Docker的海洋。

[Docker入门教程（一）介绍](#)

[Docker入门教程（二）命令](#)

[Docker入门教程（三）DockerFile](#)

[Docker入门教程（四）Docker Registry](#)

[Docker入门教程（五）Docker安全](#)

[Docker入门教程（六）另外的15个Docker命令](#)

[Docker入门教程（七）Docker API](#)

[Docker入门教程（八）Docker Remote API](#)

[Docker入门教程（九）10个镜像相关的API](#)