

# Docker

## Docker Hello World

Docker 允许你在容器内运行应用程序，使用 **docker run** 命令来在容器内运行一个应用程序。

输出Hello world

```
runoob@runoob:~$ docker run ubuntu:15.10 /bin/echo "Hello world"
```

各个参数解析：

- **docker:** Docker 的二进制执行文件。
- **run:** 与前面的 docker 组合来运行一个容器。
- **ubuntu:15.10** 指定要运行的镜像，Docker 首先从本地主机上查找镜像是否存在，如果不存在，Docker 就会从镜像仓库 Docker Hub 下载公共镜像。
- **/bin/echo "Hello world":** 在启动的容器里执行的命令

以上命令完整的意思可以解释为：Docker 以 ubuntu15.10 镜像创建一个新容器，然后在容器里执行 bin/echo "Hello world"，然后输出结果。

## 运行交互式的容器

我们通过docker的两个参数 **-i -t**，让docker运行的容器实现"对话"的能力

```
runoob@runoob:~$ docker run -i -t ubuntu:15.10 /bin/bash
```

各个参数解析：

- **-t:** 在新容器内指定一个伪终端或终端。
- **-i:** 允许你对容器内的标准输入 (STDIN) 进行交互。

我们尝试在容器中运行命令 **cat /proc/version** 和 **ls** 分别查看当前系统的版本信息和当前目录下的文件列表

我们可以通过运行 **exit** 命令或者使用 **CTRL+D** 来退出容器

## 启动容器（后台模式）

使用以下命令创建一个以进程方式运行的容器

```
runoob@runoob:~$ docker run -d ubuntu:15.10 /bin/sh -c "while true; do echo hello world; sleep 1; done"
2b1b7a428627c51ab8810d541d759f072b4fc75487eed05812646b8534a2fe63
```

在输出中，我们没有看到期望的"hello world"，而是一串长字符

**2b1b7a428627c51ab8810d541d759f072b4fc75487eed05812646b8534a2fe63**

这个长字符串叫做容器ID，对每个容器来说都是唯一的，我们可以通过容器ID来查看对应的容器发生了什么。

首先，我们需要确认容器有在运行，可以通过 **docker ps** 来查看

```
runoob@runoob:~$ docker ps
```

在容器内使用docker logs命令，查看容器内的标准输出

```
runoob@runoob:~$ docker logs 2b1b7a428627
```

## 停止容器

我们使用 **docker stop** 命令来停止容器: