

Docker 容器的使用

docker 客户端非常简单,我们可以直接输入 docker 命令来查看到 Docker 客户端的所有命令选项。

```
runoob@runoob:~# docker
```

可以通过命令 **docker command --help** 更深入的了解指定的 Docker 命令使用方法。

例如我们要查看 **docker stats** 指令的具体使用方法:

```
runoob@runoob:~# docker stats --help
```

运行一个web应用

我们将在docker容器中运行一个 Python Flask 应用来运行一个web应用。

```
runoob@runoob:~# docker pull training/webapp # 载入镜像
runoob@runoob:~# docker run -d -P training/webapp python app.py
```

参数说明:

- **-d**:让容器在后台运行。
- **-P**:将容器内部使用的网络端口映射到我们使用的主机上。

查看 WEB 应用容器

使用 docker ps 来查看我们正在运行的容器

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

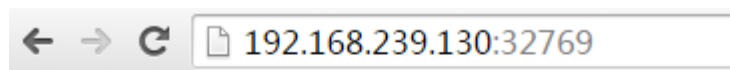
```
runoob@runoob:~$ docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                    NAMES
7ecb9ce111d   training/webapp "python app.py"         18 seconds ago Up 18 seconds  0.0.0.0:32769->5000/tcp   stupefied_rosalind
```

这里多了端口信息。

```
PORTS
0.0.0.0:32769->5000/tcp
```

Docker 开放了 5000 端口（默认 Python Flask 端口）映射到主机端口 32769 上。

这时我们可以通过浏览器访问WEB应用



Hello world!

我们也可以指定 -p 标识来绑定指定端口。

```
runoob@runoob:~$ docker run -d -p 5000:5000 training/webapp python app.py
```

docker ps查看正在运行的容器

```
runoob@runoob:~$ docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                    NAMES
7a38a1ad55c6   training/webapp "python app.py"         55 seconds ago Up 52 seconds  0.0.0.0:5000->5000/tcp   determined_swanson
```

容器内部的 5000 端口映射到我们本地主机的 5000 端口上。

查看**WEB**应用程序日志

docker logs [ID或者名字] 可以查看容器内部的标准输出。

```
runoob@runoob:~$ docker logs -f 7a38a1ad55c6
* Running on http://0.0.0.0:5000/ (Press CTRL+C to quit)
192.168.239.1 - - [09/May/2016 16:30:37] "GET / HTTP/1.1" 200 -
192.168.239.1 - - [09/May/2016 16:30:37] "GET /favicon.ico HTTP/1.1" 404 -
```

-f:让 **dokcer logs** 像使用** tail -f** 一样来输出容器内部的标准输出。

从上面，我们可以看到应用程序使用的是 5000 端口并且能够查看到应用程序的访问日志。

检查**WEB**应用程序

使用 **docker inspect** 来查看Docker的底层信息。它会返回一个 JSON 文件记录着 Docker 容器的配置和状态信息。

```
runoob@runoob:~$ docker inspect determined_swanson
[
  {
    "Id": "7a38a1ad55c6914b360b565819604733db751d86afd2575236a70a2519527361",
    "Created": "2016-05-09T16:20:45.427996598Z",
    "Path": "python",
    "Args": [
      "app.py"
    ],
    "State": {
      "Status": "running",
      .....
```

停止**WEB**应用容器

```
runoob@runoob:~$ docker stop determined_swanson
determined_swanson
```

重启**WEB**应用容器

已经停止的容器，我们可以使用命令 **docker start** 来启动。

```
runoob@runoob:~$ docker start determined_swanson
determined_swanson
```

移除**WEB**应用容器

我们可以使用 `docker rm` 命令来删除不需要的容器

```
runoob@runoob:~$ docker rm determined_swanson  
determined_swanson
```

删除容器时，容器必须是停止状态