

## Docker 容器连接

前面我们实现了通过网络端口来访问运行在 docker 容器内的服务。下面我们来实现通过端口连接到一个 docker 容器

### 网络端口映射

我们创建了一个 python 应用的容器。

```
runoob@runoob:~$ docker run -d -P training/webapp python app.py
fce072cc88cee71b1cdceb57c2821d054a4a59f67da6b416fceb5593f059fc6d
```

另外，我们可以指定容器绑定的网络地址，比如绑定 127.0.0.1。

我们使用 **-P** 参数创建一个容器，使用 **docker ps** 来看到端口 5000 绑定主机端口 32768。

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

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORT
S	NAMES				
fce072cc88ce	training/webapp	"python app.py"	4 minutes ago	Up 4 minutes	0.0.
0.0:32768->5000/tcp	grave_hopper				

我们也可以使用 **-p** 标识来指定容器端口绑定到主机端口。

两种方式的区分是：

- **-P** :是容器内部端口随机映射到主机的高端口。
- **-p** : 是容器内部端口绑定到指定的主机端口。

```
runoob@runoob:~$ docker run -d -p 5000:5000 training/webapp python app.py
33e4523d30aaf0258915c368e66e03b49535de0ef20317d3f639d40222ba6bc0
```

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

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	POR
TS	NAMES				
33e4523d30aa	training/webapp	"python app.py"	About a minute ago	Up About a minute	0.0
.0.0:5000->5000/tcp	berserk_bartik				
fce072cc88ce	training/webapp	"python app.py"	8 minutes ago	Up 8 minutes	0.0
.0.0:32768->5000/tcp	grave_hopper				

另外，我们可以指定容器绑定的网络地址，比如绑定 127.0.0.1。

```
runoob@runoob:~$ docker run -d -p 127.0.0.1:5001:5000 training/webapp python app.py
95c6ceef88ca3e71eaf303c2833fd6701d8d1b2572b5613b5a932dfdf8a857c
```

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

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORT
S	NAMES				
95c6ceef88ca	training/webapp	"python app.py"	6 seconds ago	Up 6 seconds	5000
/tcp, 127.0.0.1:5001->5000/tcp	adoring_stonebraker				
33e4523d30aa	training/webapp	"python app.py"	3 minutes ago	Up 3 minutes	0.0.
0.0:5000->5000/tcp	berserk_bartik				
fce072cc88ce	training/webapp	"python app.py"	10 minutes ago	Up 10 minutes	0.0.
0.0:32768->5000/tcp	grave_hopper				

这样我们就可以通过访问 127.0.0.1:5001 来访问容器的 5000 端口。

上面的例子中，默认都是绑定 tcp 端口，如果要绑定 UDP 端口，可以在端口后面加上 **/udp**。

```
runoob@runoob:~$ docker run -d -p 127.0.0.1:5000:5000/udp training/webapp python app.py
6779686f06f6204579c1d655dd8b2b31e8e809b245a97b2d3a8e35abe9dcd22a
```

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

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORT
S	NAMES				
6779686f06f6	training/webapp	"python app.py"	4 seconds ago	Up 2 seconds	5000
/tcp, 127.0.0.1:5000->5000/udp	drunk_visvesvaraya				
95c6ceef88ca	training/webapp	"python app.py"	2 minutes ago	Up 2 minutes	5000
/tcp, 127.0.0.1:5001->5000/tcp	adoring_stonebraker				
33e4523d30aa	training/webapp	"python app.py"	5 minutes ago	Up 5 minutes	0.0.
0.0:5000->5000/tcp	berserk_bartik				
fce072cc88ce	training/webapp	"python app.py"	12 minutes ago	Up 12 minutes	0.0.
0.0:32768->5000/tcp	grave_hopper				

**docker port** 命令可以让我们快捷地查看端口的绑定情况。

```
runoob@runoob:~$ docker port adoring_stonebraker 5000
127.0.0.1:5001
```

## Docker容器连接

端口映射并不是唯一把 docker 连接到另一个容器的方法。

docker 有一个连接系统允许将多个容器连接在一起，共享连接信息。

docker 连接会创建一个父子关系，其中父容器可以看到子容器的信息。

### 容器命名

当我们创建一个容器的时候，docker 会自动对它进行命名。另外，我们也可以使用 **--name** 标识来命名容器，例如：

```
runoob@runoob:~$ docker run -d -P --name runoob training/webapp python app.py
43780a6eabaaf14e590b6e849235c75f3012995403f97749775e38436db9a441
```

我们可以使用 **docker ps** 命令来查看容器名称。

runoob@runoob:~\$ docker ps -l

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORT
S	NAMES				
43780a6eabaa	training/webapp	"python app.py"	3 minutes ago	Up 3 minutes	0.0.
0.0:32769->5000/tcp	runoob				

[← Docker 镜像使用](#)

[Docker 安装 Nginx →](#)

[📝 点我分享笔记](#)