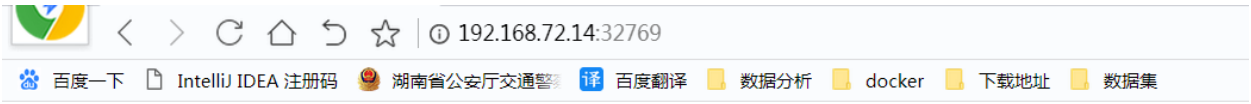
11.Docker容器绑定外部端口和IP

1.Docker允许通过外部访问容器或者容器之间互联的方式来提供网络服务。 2.首先是外部访问容器：

- 容器启动之后，容器中可以运行一些网络应用，通过-p或-P参数来指定端口映射，使用-P（大写）标记时，docker会随机选择一个端口映射到容器内部开放的网络端口上。

```
[root@bigdata-4 ~]# docker run -d -P training/webapp python app.py
60b684d32e043479b102ef6b93a5c0138da7b767727b0058ee4fff713b268418
[root@bigdata-4 ~]# docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS
                                                                                                    NAMES
60b684d32e04        training/webapp     "python app.py"     31 seconds ago      Up 30 seconds       0.0.0.0:32769->5000/tc
p                                                                                                   peaceful_euler
```

此时访问本机的32769端口就可以访问到容器内web应用提供的界面。



Hello world!

也可以使用docker logs来查看应用的信息

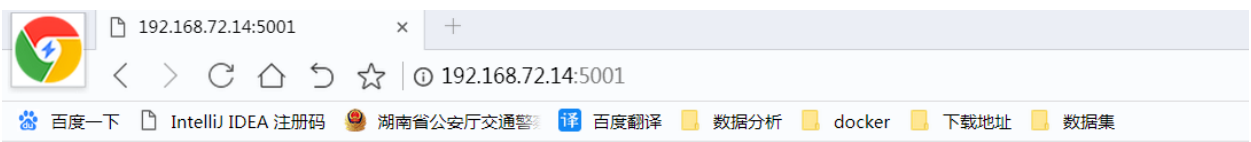
```
                                                                                                    hive
[root@bigdata-4 ~]# docker logs peaceful_euler
 * Running on http://0.0.0.0:5000/ (Press CTRL+C to quit)
192.168.72.1 - - [17/Sep/2019 14:37:54] "GET / HTTP/1.1" 200 -
192.168.72.1 - - [17/Sep/2019 14:37:57] "GET /favicon.ico HTTP/1.1" 404 -
[root@bigdata-4 ~]#
```

- 使用-p（小写)则可以指定要映射的端口，并且在一个指定端口上只可以绑定一个容器，支持的格式有：

```
ip:hostport:containerport ip::containerport hostport:containerport
docker run -d -p 5000:5000 training/webapp python app.py
```

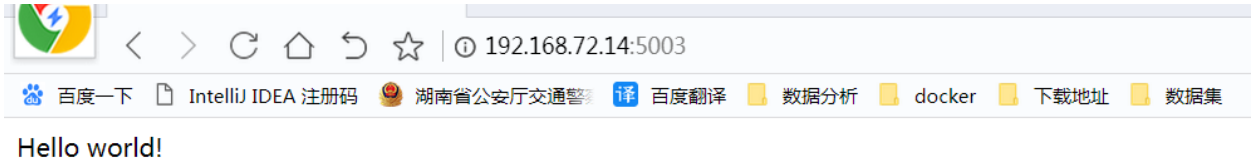
```
[root@bigdata-4 ~]# docker run -d -p 5000:5000 training/webapp python app.py
c829f830fa2c344ff5102d5ea71fdb29389ea8b7e7571d6a511cac743befb640
/usr/bin/docker-current: Error response from daemon: driver failed programming external connectivity on endpoint elastic_leavit
t (995e45b940c8b9d7866731b6fbadd6b1f5b40ac330eaac6ea915edd52c546acc): Bind for 0.0.0.0:5000 failed: port is already allocated.
[root@bigdata-4 ~]# netstat -nlpt |grep 5000
tcp6       0      0 :::5000                 :::*                    LISTEN      5476/docker-proxy-c
[root@bigdata-4 ~]# docker run -d -p 5001:5000 training/webapp python app.py
cdfd4d1e670a0040698388f8fb140e2fa8c20b13f5ad665d26ffde62229a2d76
[root@bigdata-4 ~]# docker logs cdfd4d1e670a0040698388f8fb140e2fa8c20b13f5ad665d26ffde62229a2d76
 * Running on http://0.0.0.0:5000/ (Press CTRL+C to quit)
192.168.72.1 - - [17/Sep/2019 14:48:14] "GET / HTTP/1.1" 200 -
192.168.72.1 - - [17/Sep/2019 14:48:15] "GET /favicon.ico HTTP/1.1" 404 -
[root@bigdata-4 ~]#
```

看到本地5000端口已经被映射，换一个端口5001运行成功，在页面上输入5001端口访问如下



Hello world!

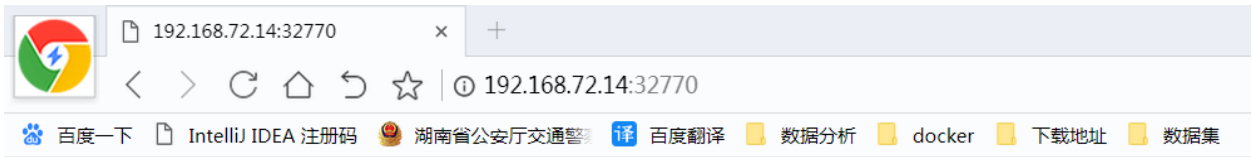
- docker 默认会映射本地所有的地址。现在我们来尝试一下映射到指定地址的指定端口 docker run -d -p 192.168.72.14:5003:5000 training/webapp python app.py



Hello world!

接下来绑定本机的任意端口到容器的5000端口 docker run -d -p 192.168.72.14::5000 training/webapp python app.py

```
[root@bigdata-4 ~]# docker run -d -p 192.168.72.14::5000 training/webapp python app.py
5602e1270e13e7638d7b960c09c9b20d75fc4acf40641d795f52c4208d743b07
[root@bigdata-4 ~]# docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS                                NAMES
5602e1270e13        training/webapp     "python app.py"     9 seconds ago       Up 8 seconds        192.168.72.14:32770->5000/tcp        pensive_curie
```



Hello world!

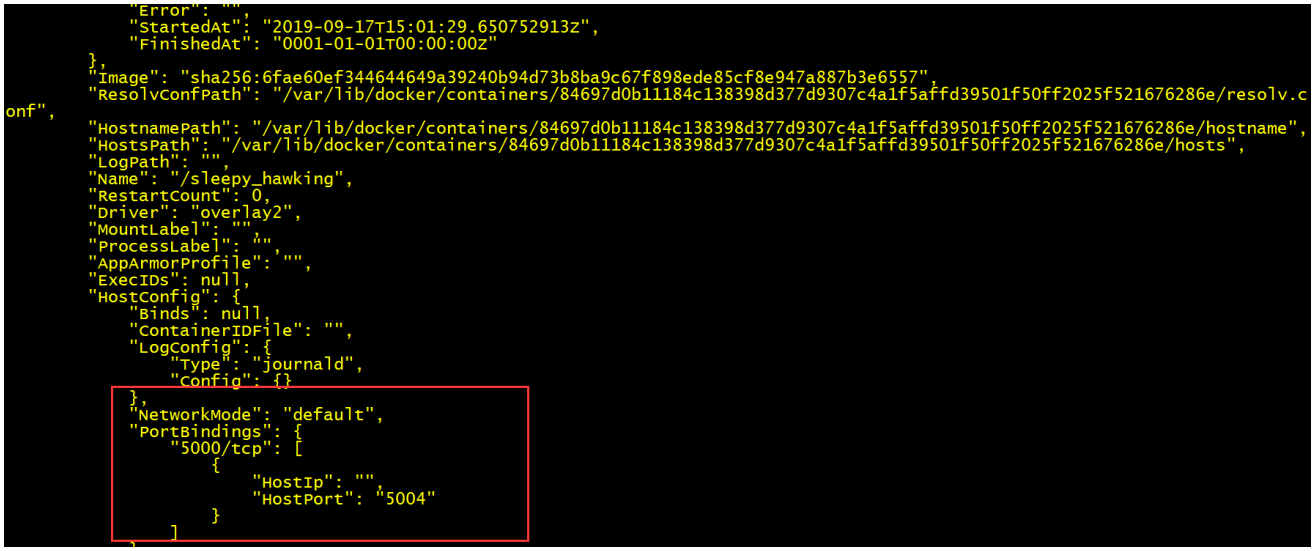
还可以指定通信协议 docker run -d -p 5003:5000/udp training/webapp python app.py

```
[root@bigdata-4 ~]# docker run -d -p 5003:5000/udp training/webapp python app.py
a48ade074be656011c081f22faebfb291db2087faf6c8a5e974becfb1eafd70e
[root@bigdata-4 ~]# docker run -d -p 5004:5000/tcp training/webapp python app.py
84697d0b11184c138398d377d9307c4a1f5affd39501f50ff2025f521676286e
[root@bigdata-4 ~]#
```

那我们怎样来查看docker里面容易绑定和映射的端口及Ip地址呢？可以使用docker port

```
[root@bigdata-4 ~]# docker port 84697d0b11184c138398d377d9307c4a1f5affd39501f50ff2025f521676286e
5000/tcp -> 0.0.0.0:5004
[root@bigdata-4 ~]#
```

需要注意的是： a.容器内部有自己的内部网络和ip地址，可以使用docker inspect来查看



b.在启动容器的时候，可以多次使用-p标记来绑定多个端口 docker run -d -p 5005:5000 -p 5006:80 training/webapp python app.py