



链滴

Docker 搭建单机 FastDFS 服务 —— Docker 成长之路

作者: [724555508](#)

原文链接: <https://ld246.com/article/1594012692728>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



1.拉取镜像

```
docker pull delron/fastdfs
```

2.使用 docker 镜像构建 tracker 容器（跟踪服务器，起到调度的作用）：

```
docker run -dti --network=host --name tracker -v /var/fdfs/tracker:/var/fdfs -v /etc/localtime:/etc/localtime delron/fastdfs tracker
```

3.使用 docker 镜像构建 storage 容器（存储服务器，提供容量和备份服务）：

```
docker run -dti --network=host --name storage -e TRACKER_SERVER=192.168.56.1:22122 -v /var/fdfs/storage:/var/fdfs -v /etc/localtime:/etc/localtime delron/fastdfs storage
```

TRACKER_SERVER= 本机的 ip 地址：22122 本机 ip 地址不要使用 127.0.0.1 最好写成外网地址

4.配置

进入 storage 容器，到 storage 的配置文件中配置 http 访问的端口，配置文件在/etc/fdfs 目录下的 storage.conf

```
# the port of the web server on this storage server
http_server_port=8888
█
```

默认端口是 8888，也可以不进行更改。

进入 storage 配置 nginx，在/usr/local/nginx 目录下，修改 nginx.conf 文件，默认配置不修改也以

```
server {
    listen      8888;
    server_name localhost;
    location ~/group[0-9]/ {
        ngx_fastdfs_module;
    }
    error_page 500 502 503 504 /50x.html;
    location = /50x.html {
        root html;
    }
}
ginx.conf" 81L, 1709C
```

5.测试上传文件

将一张照片 (test.png) 放置在/var/dfs/storage 目录下, 进入 storage 容器, 进入/var/dfs 目录运行下面命令:

```
[root@localhost dfs]# mv 888827
[root@localhost dfs]# ls
data logs test.png
[root@localhost dfs]#
```

```
/usr/bin/dfs_upload_file /etc/dfs/client.conf test.png
```

此时将该图片已上传至文件系统, 并在执行该语句后返回图片存储的 uri

```
[root@localhost dfs]# /usr/bin/dfs_upload_file /etc/dfs/client.conf test.png
group1/M00/00/00/CgACD1z7SEuAXrIqAA1eBLGVLow043.png
```

通过IP:8888/group1.....png访问即可查到图片

6.开放端口

```
firewall-cmd --zone=public --permanent --add-port=8888/tcp
```

```
firewall-cmd --zone=public --permanent --add-port=22122/tcp
```

```
firewall-cmd --zone=public --permanent --add-port=23000/tcp
```

```
systemctl restart firewalld //重启防火墙
```

7.设置开机启动容器

```
docker update --restart=always tracker
```

```
docker update --restart=always storage
```

8.over