Docker 应用部署-`FastDFS`

1 准备目录

安装前在`home`目录下面新建`dfs`目录，然后将资源里面的`docker.zip`上传到`dfs`目录下面，并使用`unzip docker.zip`命令解压它。

下面分别演示，本地文件构建和在线文件构建，根据你自身情况选择其中一种方式就可以了

推荐使用本地文件方式，因为线上构建方式可能遇到版本更新后构建脚本不同步的问题。

2 开放端口

开放防火墙端口，***注意：如果是云服务器，请在云控制台网站上面配置防火墙入方向规则放行端口，而不是执行下面命令放行端口。***

```
#开放端口
firewall-cmd --add-port 8888/tcp --add-port 22122/tcp --add-port 23000/tcp --permanent
#重新加载防火墙
firewall-cmd --reload
```

3 本地文件构建

参考链接：https://gitee.com/fastdfs100/fastdfs/tree/master/docker/dockerfile_local-v6.0.9

3.1 构建镜像

TIPS：如果已经有现成的`dfs`镜像，可以跳过此步骤，直接使用现有的镜像。

```
cd /home/dfs/docker/dockerfile_local/
docker build -f ./Dockerfile -t awei/dfs:6.09 .
```

3.2 修改配置文件

主要修改`/home/dfs/docker/dockerfile_local/conf`目录下面的配置：

- storage_ids.conf

找到类似下面的配置，修改对应ip为你服务器的ip地址

```
100001   group1  192.168.220.128
```

- storage.conf文件

找到类似下面的配置，修改对应ip为你服务器的ip地址

```
tracker_server = 192.168.220.128:22122
```

- mod_fastdfs.conf文件

找到类似下面的配置，修改对应ip为你服务器的ip地址

```
tracker_server = 192.168.220.128:22122
```

3.3 服务编排

在`/home/dfs/docker/dockerfile_local`目录下面新建`docker-compose.yml`文件，并写入如下内容：

```yaml
version: '3'
services:
  dfs-tracker:
    image: awei/dfs:6.09
    container_name: dfs-tracker
    network_mode: host
    volumes:
      - /etc/localtime:/etc/localtime
      - /home/dfs/docker/dockerfile_local/tracker/data/:/data/fastdfs_data
      - /home/dfs/docker/dockerfile_local/conf/:/etc/fdfs
    command: tracker
  dfs-storage:
    image: awei/dfs:6.09
    container_name: dfs-storage
    network_mode: host
    privileged: true
    volumes:
      - /etc/localtime:/etc/localtime
      - /home/dfs/docker/dockerfile_local/storage/metadata/:/data/fastdfs_data
      - /home/dfs/docker/dockerfile_local/conf/:/etc/fdfs
      - /home/dfs/docker/dockerfile_local/upload/:/data/fastdfs/upload
      - /home/dfs/docker/dockerfile_local/nginx_conf/nginx.conf:/usr/local/nginx/conf/nginx.conf
      - /home/dfs/docker/dockerfile_local/nginx_conf.d:/usr/local/nginx/conf.d
    command: storage
    depends_on:
      - dfs-tracker
```

当然提供给大家的压缩包里面已经有服务编排文件了，可以直接使用。

3.4 创建存储目录

执行下面命令创建存储目录

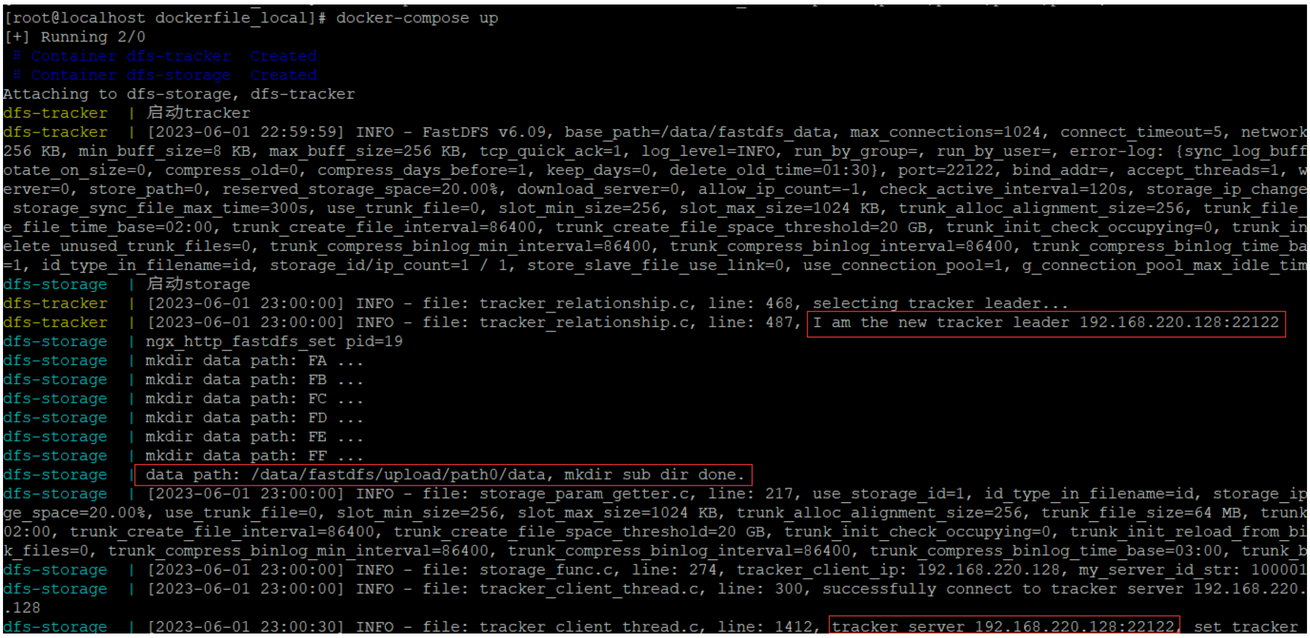
```
mkdir -p /home/dfs/docker/dockerfile_local/upload/{path0,path1,path2,path3}
```

3.5 启动测试

首先执行前台启动

```
cd /home/dfs/docker/dockerfile_local/
docker-compose up
```

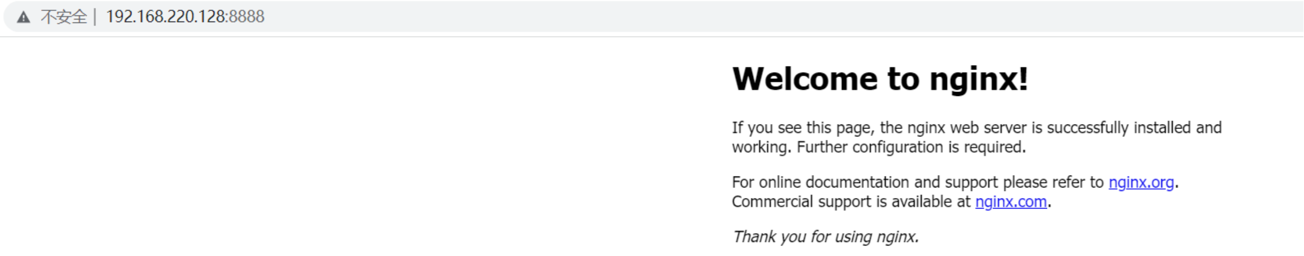
观察启动启动日志，如果没有明显错误表启动成功，示例如下图所示



```
[root@localhost dockerfile_local]# docker-compose up
[+] Running 2/0
 # Container dfs-tracker  Created
 # Container dfs-storage  Created
Attaching to dfs-storage, dfs-tracker
dfs-tracker  | 启动tracker
dfs-tracker  | [2023-06-01 22:59:59] INFO - FastDFS v6.09, base_path=/data/fastdfs_data, max_connections=1024, connect_timeout=5, network
256 KB, min_buff_size=8 KB, max_buff_size=256 KB, tcp_quick_ack=1, log_level=INFO, run_by_group=, run_by_user=, error-log: {sync_log_buff
otate_on_size=0, compress_old=0, compress_days_before=1, keep_days=0, delete_old_time=01:30}, port=22122, bind_addr=, accept_threads=1, w
erver=0, store_path=0, reserved_storage_space=20.00%, download_server=0, allow_ip_count=-1, check_active_interval=120s, storage_ip_change
 storage_sync_file_max_time=300s, use_trunk_file=0, slot_min_size=256, slot_max_size=1024 KB, trunk_alloc_alignment_size=256, trunk_file_
e_file_time_base=02:00, trunk_create_file_interval=86400, trunk_create_file_space_threshold=20 GB, trunk_init_check_occupying=0, trunk_in
elete_unused_trunk_files=0, trunk_compress_binlog_min_interval=86400, trunk_compress_binlog_interval=86400, trunk_compress_binlog_time_ba
=1, id_type_in_filename=id, storage_id/ip_count=1 / 1, store_slave_file_use_link=0, use_connection_pool=1, g_connection_pool_max_idle_tim
dfs-storage  | 启动storage
dfs-tracker  | [2023-06-01 23:00:00] INFO - file: tracker_relationship.c, line: 468, selecting tracker leader...
dfs-tracker  | [2023-06-01 23:00:00] INFO - file: tracker_relationship.c, line: 487, I am the new tracker leader 192.168.220.128:22122
dfs-storage  | ngx_http_fastdfs_set pid=19
dfs-storage  | mkdir data path: FA ...
dfs-storage  | mkdir data path: FB ...
dfs-storage  | mkdir data path: FC ...
dfs-storage  | mkdir data path: FD ...
dfs-storage  | mkdir data path: FE ...
dfs-storage  | mkdir data path: FF ...
dfs-storage  | data path: /data/fastdfs/upload/path0/data, mkdir sub dir done.
dfs-storage  | [2023-06-01 23:00:00] INFO - file: storage_param_getter.c, line: 217, use_storage_id=1, id_type_in_filename=id, storage_ip
ge_space=20.00%, use_trunk_file=0, slot_min_size=256, slot_max_size=1024 KB, trunk_alloc_alignment_size=256, trunk_file_size=64 MB, trunk
02:00, trunk_create_file_interval=86400, trunk_create_file_space_threshold=20 GB, trunk_init_check_occupying=0, trunk_init_reload_from_bi
k_files=0, trunk_compress_binlog_min_interval=86400, trunk_compress_binlog_interval=86400, trunk_compress_binlog_time_base=03:00, trunk_b
dfs-storage  | [2023-06-01 23:00:00] INFO - file: storage_func.c, line: 274, tracker_client_ip: 192.168.220.128, my_server_id_str: 100001
dfs-storage  | [2023-06-01 23:00:00] INFO - file: tracker_client_thread.c, line: 300, successfully connect to tracker server 192.168.220.
.128
dfs-storage  | [2023-06-01 23:00:30] INFO - file: tracker_client_thread.c, line: 1412, tracker server 192.168.220.128:22122, set tracker
```

控制台没有明显错误表示启动成功。

然后在浏览器中访问Nginx服务器，检查服务状态，访问地址：http://ip:8888，如果能够正常访问，可以看到如下图所示的示例效果：



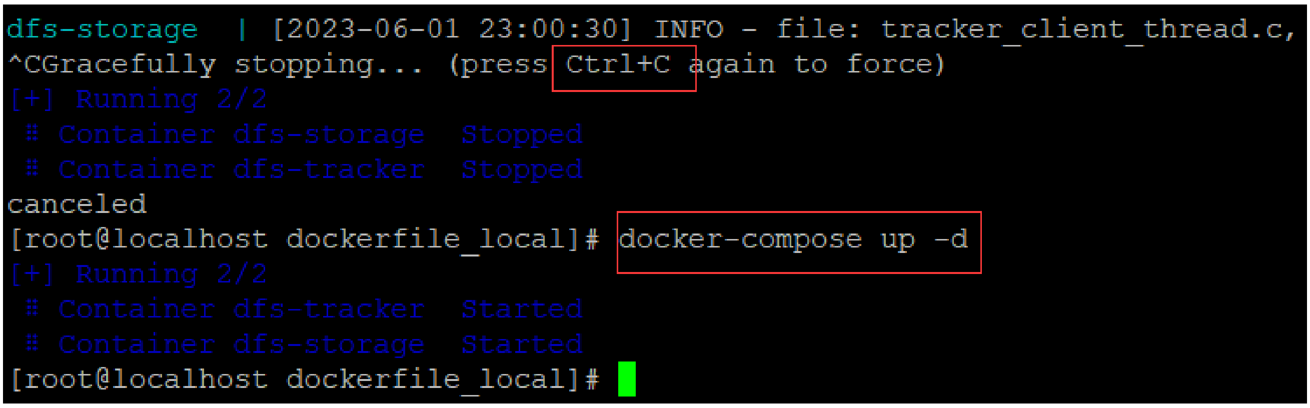
Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

然后Ctrl+C结束前台启动，切换成后台启动即可，如下图所示：



```
dfs-storage  | [2023-06-01 23:00:30] INFO - file: tracker_client_thread.c,
^CGracefully stopping... (press Ctrl+C again to force)
[+] Running 2/2
 # Container dfs-storage  Stopped
 # Container dfs-tracker  Stopped
canceled
[root@localhost dockerfile_local]# docker-compose up -d
[+] Running 2/2
 # Container dfs-tracker  Started
 # Container dfs-storage  Started
[root@localhost dockerfile_local]# 
```

到此使用本地文件搭建FastDFS服务器完成，更多配置参数优化请参考官网。

4 在线文件构建

参考链接：https://gitee.com/fastdfs100/fastdfs/tree/master/docker/dockerfile_network

4.1 构建镜像

```
cd /home/dfs/docker/dockerfile_network/
docker build -f ./Dockerfile -t awei/dfs .
```

4.2 创建容器

注意：需要将FASTDFS_IPADDR对应的IP地址对应成你宿主机的IP地址，如果是云服务器使用公网IP

```
docker run -d \
-e FASTDFS_IPADDR=192.168.220.128 \
--net=host \
--name fast-dfs \
-v /home/dfs/docker/dockerfile_network/:/home/dfs/ \
awei/dfs
```

4.3 启动失败

通过`docker logs fast-dfs`查看日志，如果出现下列错误提示。

```
exec /home/fastdfs.sh: no such file or directory
```

是因为文件格式编码的问题，可以通过`vi`指令修改`/home/dfs/docker/dockerfile_network/fastdfs.sh`的文件编码格式为unix格式。

```
# 编辑文件
vi fastdfs.sh

# 查看文件编码格式
:set ff
# 会看到是这个样子的：fileformat=dos

# 修改编码
:set ff=unix

# 保存退出
:wq
```

修改完成后执行删除镜像重新构建一次。

```
docker rm fast-dfs
docker rmi awei/dfs:latest
# 执行镜像构建指令，参考："4.1 构建镜像"
# 执行创建容器指令，参考："4.2 创建容器"
```

如果上面方式不能解决问，可以到gitee上面下载最新的脚本完成安装。