docker-compose一键部署FastDFS

部署环境

部署环境docker和docker-compose. Ubuntu安装docker和docker-compose.

创建Dokcerfile

本DockerFile采用的软件版本：

- fastdfs-5.11：master 最新版(截至2018/8/11)
- libfastcommon：1.0.38
- fastdfs-nginx-module-1.20： 1.20
- nginx：1.13.6

软件均来自GitHub项目.

为了方便操作请求使用root 用户操作Linux.

创建目录

```
 mkdir  -p  /user/local/docker/fastdfs/soft && cd /user/local/docker/fastdfs
```

将下载的软件压缩包放在/user/local/docker/fastdfs/soft 目录

软件可以从 百度云下载.

```
root@server:/usr/local/docker/fastdfs-single-5.11/soft# ll
total 2516
drwxr-xr-x 2 root root    4096 Aug 11 00:56 ./
drwxr-xr-x 4 root root    4096 Aug 11 01:53 ../
-rw-r--r-- 1 root root 1108472 Aug 11 00:56 fastdfs-5.11.tar.gz
-rw-r--r-- 1 root root   19825 Aug 11 00:56 fastdfs-nginx-module-1.20.tar.gz
-rw-r--r-- 1 root root  444656 Aug 11 00:56 libfastcommon-1.0.38.tar.gz
-rw-r--r-- 1 root root  989760 Aug 11 00:56 nginx-1.13.6.tar.gz
root@server:/usr/local/docker/fastdfs-single-5.11/soft#
```

创建Dockerfile

在fastdfs目录下创建Dockerfile

```
vim Dockerfile
```

Dockerfile内容如下：

```
# 使用超小的Linux镜像alpine
FROM alpine:3.6

MAINTAINER meiko-zhang<https://github.com/meiko-zhang>

ENV HOME /root

# 安装准备
RUN    apk update \
        && apk add --no-cache --virtual .build-deps bash gcc libc-dev mak
e openssl-dev pcre-dev zlib-dev linux-headers curl gnupg libxslt-dev gd-d
ev geoip-dev

# 复制工具
ADD soft ${HOME}

RUN     cd ${HOME} \
        && tar zxf libfastcommon-1.0.38.tar.gz \
        && tar zxf fastdfs-5.11.tar.gz \
        && tar zxf fastdfs-nginx-module-1.20.tar.gz

# 安装libfastcommon
RUN     cd ${HOME}/libfastcommon-1.0.38/ \
        && ./make.sh \
        && ./make.sh install

# 安装fastdfs
RUN     cd ${HOME}/fastdfs-5.11/ \
        && ./make.sh \
        && ./make.sh install

# 配置fastdfs: base_dir
RUN     cd /etc/fdfs/ \
        && cp storage.conf.sample storage.conf \
        && cp tracker.conf.sample tracker.conf \
        && cp client.conf.sample client.conf \
        && sed -i "s|/home/yuqing/fastdfs|/var/local/fdfs/tracker|g" /et
c/fdfs/tracker.conf \
        && sed -i "s|/home/yuqing/fastdfs|/var/local/fdfs/storage|g" /et
c/fdfs/storage.conf \
```

```
        && sed -i "s|/home/yuqing/fastdfs|/var/local/fdfs/storage|g" /et
c/fdfs/client.conf

# 获取nginx源码，与fastdfs插件一起编译
RUN     cd ${HOME} \
        && tar zxf nginx-1.13.6.tar.gz \
        && chmod u+x ${HOME}/fastdfs-nginx-module-1.20/src/config \
        && cd nginx-1.13.6 \
        && ./configure --add-module=${HOME}/fastdfs-nginx-module-1.20/src
 \
        && make && make install

# 设置nginx和fastdfs联合环境，并配置nginx
RUN     cp ${HOME}/fastdfs-nginx-module-1.20/src/mod_fastdfs.conf /etc/fd
fs/ \
        && sed -i "s|^store_path0.*$|store_path0=/var/local/fdfs/storage|
g" /etc/fdfs/mod_fastdfs.conf \
        && sed -i "s|^url_have_group_name =.*$|url_have_group_name = true
|g" /etc/fdfs/mod_fastdfs.conf \
        && cd ${HOME}/fastdfs-5.11/conf/ \
        && cp http.conf mime.types anti-steal.jpg /etc/fdfs/ \
        && echo -e "\
           events {\n\
           worker_connections  1024;\n\
           }\n\
           http {\n\
           include       mime.types;\n\
           default_type  application/octet-stream;\n\
           server {\n\
               listen 8888;\n\
               server_name localhost;\n\

               location ~ /group[0-9]/M00 {\n\
                 ngx_fastdfs_module;\n\
               }\n\
             }\n\
            }" >/usr/local/nginx/conf/nginx.conf

# 清理文件
RUN rm -rf ${HOME}/*
RUN apk del .build-deps gcc libc-dev make openssl-dev linux-headers curl
 gnupg libxslt-dev gd-dev geoip-dev
RUN apk add bash pcre-dev zlib-dev


# 配置启动脚本，在启动时中根据环境变量替换nginx端口、fastdfs端口
# 默认nginx端口
ENV WEB_PORT 8888
# 默认fastdfs端口
ENV FDFS_PORT 22122
# 创建启动脚本
RUN     echo -e "\
```

```
mkdir -p /var/local/fdfs/storage/data /var/local/fdfs/tracker; \n\
ln -s /var/local/fdfs/storage/data/ /var/local/fdfs/storage/data/M00;
\n\n\
sed -i \"s/listen\ .*$/listen\ \$WEB_PORT;/g\" /usr/local/nginx/conf/ngin
x.conf; \n\
sed -i \"s/http.server_port=.*$/http.server_port=\$WEB_PORT/g\" /etc/fdf
s/storage.conf; \n\n\
if [ \"\$IP\" = \"\" ]; then \n\
    IP=`ifconfig eth0 | grep inet | awk '{print \$2}'| awk -F: '{print
\$2}'`; \n\
fi \n\
sed -i \"s/^tracker_server=.*$/tracker_server=\$IP:\$FDFS_PORT/g\" /etc/f
dfs/client.conf; \n\
sed -i \"s/^tracker_server=.*$/tracker_server=\$IP:\$FDFS_PORT/g\" /etc/f
dfs/storage.conf; \n\
sed -i \"s/^tracker_server=.*$/tracker_server=\$IP:\$FDFS_PORT/g\" /etc/f
dfs/mod_fastdfs.conf; \n\n\
/etc/init.d/fdfs_trackerd start; \n\
/etc/init.d/fdfs_storaged start; \n\
/usr/local/nginx/sbin/nginx; \n\
tail -f /usr/local/nginx/logs/access.log \
">/start.sh \
&& chmod u+x /start.sh

# 暴露端口。改为采用host网络，不需要单独暴露端口
# EXPOSE 80 22122

ENTRYPOINT ["/bin/bash","/start.sh"]
```

创建docker-compose.yml

在fastdfs目录下创建docker-compose.yml

```
vim docker-compose.yml
```

docker-compose.yaml内容如下：

```
version: '3.0'

services:
    fastdfs:
        build: .
        image: meiko/fastdfs-single:5.11
        # 该容器是否需要开机启动+自动重启。若需要，则取消注释。
        restart: always
        container_name: fastdfs-single
```

```
        environment:
            # nginx服务端口
            - WEB_PORT=8888
            # docker所在主机的IP地址
            - IP=192.168.73.141
        volumes:
            # 将本地目录映射到docker容器内的fastdfs数据存储目录，将fastdfs文件存储到主机上，以免每次重建docker容器，之前存储的文件就丢失了。
            - ${HOME}/docker-data/fdfs:/var/local/fdfs
        # 使docker具有root权限以读写主机上的目录
        privileged: true
        # 网络模式为host，即直接使用主机的网络接口
        network_mode: "host"
```

准备

目录结构:

```
root@server:/usr/local/docker/fastdfs-single-5.11# tree
.
├── docker-compose.yml
├── Dockerfile
├── log.sh
├── README.md
└── soft
    ├── fastdfs-5.11.tar.gz
    ├── fastdfs-nginx-module-1.20.tar.gz
    ├── libfastcommon-1.0.38.tar.gz
    └── nginx-1.13.6.tar.gz

1 directory, 8 files
root@server:/usr/local/docker/fastdfs-single-5.11#
```

本地fastdfs数据存储目录,如果没有需要先创建.(可以自行修改为自己的目录)

```
mkdir -p ${HOME}/docker-data/fdfs
```

启动

```
$ docker-compose up -d
# 因为要下载软件包和源码，并编译，所以过程比较漫长，期间可能会出现红字的warning，不必理会。若报错，请根据提示排查。
...
Successfully built 4baafa5d2e75
Successfully tagged meiko/fastdfs-single:5.11
WARNING: Image for service fastdfs was built because it did not already exist. To rebuild this image you must use `docker-compose build` or `docke
```

```
r-compose up --build`.
Creating fastdfs ...
Creating fastdfs ... done

# 查看容器运行状态
$ docker ps 或 $ sudo docker-compose ps
CONTAINER ID        IMAGE                          COMMAND                  C
REATED              STATUS              PORTS               NAMES
2a294bc410bd        meiko/fastdfs-single:5.11    "/bin/bash /start.sh"
 4 minutes ago       Up 4 minutes                                fastdfs-sing
le
```

测试

```
# 进入docker容器内终端
$ docker exec -it fastdfs-single或ID的前几位 /bin/bash

# 在容器内部终端执行上传测试
bash-4.3$ echo "Hello FastDFS!">index.html
bash-4.3$ fdfs_test /etc/fdfs/client.conf upload index.html
This is FastDFS client test program v5.11
...
[2017-11-28 14:05:25] DEBUG - base_path=/var/local/fdfs/storage, connect_
timeout=30, network_timeout=60, tracker_server_count=1, anti_steal_token=
0, anti_steal_secret_key length=0, use_connection_pool=0, g_connection_po
ol_max_idle_time=3600s, use_storage_id=0, storage server id count: 0
tracker_query_storage_store_list_without_group:
        server 1. group_name=, ip_addr=192.168.56.110, port=23000
group_name=group1, ip_addr=192.168.56.110, port=23000
storage_upload_by_filename
group_name=group1, remote_filename=M00/00/00/wKg4blodbSWAImI9AAAADwA12ic7
1.html
source ip address: 192.168.73.141
file timestamp=2017-11-28 14:05:25
file size=15
file crc32=3529255
example file url: http://192.168.73.141/group1/M00/00/00/wKg4blodbSWAImI9
AAAADwA12ic71.html

# 下载测试
bash-4.3$ fdfs_test /etc/fdfs/client.conf download group1 M00/00/00/wKg4b
lodbSWAImI9AAAADwA12ic71.html
...
storage=192.168.56.110:23000
download file success, file size=15, file save to wKg4blodbSWAImI9AAAADwA
12ic71.html

# 外部http访问测试
```

```
# 在主机或其他同局域网内机器上用浏览器访问上面的url。注意：若nginx的端口设置不为80，则需加上端口号
http://192.168.73.141:8888/group1/M00/00/00/wKg4blodbSWAImI9AAAADwA12ic71.html
# 网页显示：Hello FastDFS!

# 删除测试
bash-4.3$ fdfs_test /etc/fdfs/client.conf delete group1 M00/00/00/wKg4blodbSWAImI9AAAADwA12ic71.html
storage=192.168.73.141:23000
delete file success

# 退出容器终端
bash-4.3$ exit
```

管理容器

```
# 停止容器
$ docker stop <容器NAMES，也可以为容器ID的前几位>
或 $ docker-compose stop

# 更改compose或Dockerfile后重新生成并运行
$ docker-compose stop
$ docker-compose build
$ docker-compose up -d
或 $ docker-compose up -d --build #本条命令可代替上述三条命令

# 删除容器
$ docker rm <容器NAMES，也可以为容器ID的前几位>
或 $ docker-compose rm
```

查看日志

为避免每次需要查看日志都要执行 docker exec -it /usr/bin/tail -f 命令，我将常见的查看日志命令封装到一个脚本中，每次只需要执行脚本就能查看不同服务的日志了。

新建log.sh，用来快速查看日志。内容如下：

```
#!/bin/bash
STORAGE=/var/local/fdfs/storage/logs/storaged.log
TRACKER=/var/local/fdfs/tracker/logs/trackerd.log
NGINX=/usr/local/nginx/logs/access.log

ID=`docker ps|grep fastdfs|awk '{print $1}'`
echo fastdfs.ID:$ID
```

```
echo 'Use param tracker|storage|nginx to see log of each service such as
 "./log.sh tracker". No param equals to "storage".'
CAT=$1
LOG=""
if [[ "${CAT}" = "tracker" ]];then
    LOG=${TRACKER}
elif [[ "${CAT}" = "nginx" ]]; then
    LOG=${NGINX}
else
    LOG=${STORAGE}
fi

docker exec -it $ID /usr/bin/tail -f ${LOG}
```

给log.sh添加执行权限

```
chmod u+x log.sh
```

查看日志

```
$ ./log.sh tracker或storage或nginx

# 查看nginx日志，可以看到刚刚从外部http方式访问fastdfs文件的日志
$ ./log.sh nginx
```

至此以及单机版的FastDFS 服务已经搭建完成!可以使用了!!!!

项目Github 地址.

项目百度云分享链接.

[1].参考文章.