# 1 非容器部署

## 1.1 软件安装

yum install git gcc gcc-c++ make automake autoconf libtool pcre pcre-devel zlib zlib-devel openssl-devel wget vim initscripts.x86\_64[[1]](#footnote-1) telnet net-tools lsof strace unzip -y

# 文件存储路径

mkdir -p /home/dfs

# 源代码路径

cd /usr/local/src

# 下载、编译fastdfs公共函数包

git clone https://github.com/happyfish100/libfastcommon.git --depth 1

cd libfastcommon/

./make.sh && ./make.sh install

# 下载、编译fastdfs

cd ..

git clone https://github.com/happyfish100/fastdfs.git --depth 1

cd fastdfs/

./make.sh && ./make.sh install

# 使用模板创建跟踪服务[[2]](#footnote-2)配置文件

cp /etc/fdfs/tracker.conf.sample /etc/fdfs/tracker.conf

# 使用模板创建存储服务[[3]](#footnote-3)配置文件

cp /etc/fdfs/storage.conf.sample /etc/fdfs/storage.conf

# fastdfs客户端配置文件

cp /etc/fdfs/client.conf.sample /etc/fdfs/client.conf

cp /usr/local/src/fastdfs/conf/http.conf /etc/fdfs/

cp /usr/local/src/fastdfs/conf/mime.types /etc/fdfs/

cd ..

# 安装fastdfs和nginx的关联模块

git clone https://github.com/happyfish100/fastdfs-nginx-module.git --depth 1

cp /usr/local/src/fastdfs-nginx-module/src/mod\_fastdfs.conf /etc/fdfs

# 安装nginx

# 删除原有nginx

yum erase nginx -y

wget <http://nginx.org/download/nginx-1.15.4.tar.gz>

tar -zxvf nginx-1.15.4.tar.gz

cd nginx-1.15.4

./configure --add-module=/usr/local/src/fastdfs-nginx-module/src/

make && make install

## 1.2 集群模式配置

# 跟踪服务监听端口，默认22122

sed -i "s/^port.\*=.\*/port = 22122/g" /etc/fdfs/tracker.conf

# 存储日志和数据的根目录

sed -i "s/^base\_path.\*=.\*/base\_path = \/home\/dfs/g" /etc/fdfs/tracker.conf

# 存储服务默认端口23000

sed -i "s/^port.\*=.\*/port = 23000/g" /etc/fdfs/storage.conf

# 存储服务根路径

sed -i "s/^base\_path.\*=.\*/base\_path = \/home\/dfs/g" /etc/fdfs/storage.conf

sed -i 's/^store\_path0.\*=.\*/store\_path0 = \/home\/dfs/g' /etc/fdfs/storage.conf

# 分别替换第1、2、3个tracker节点地址

sed -i '0,/^tracker\_server.\*=.\*/s//tracker\_server = 192.168.100.62:22122/' /etc/fdfs/storage.conf

sed -i '/**^tracker\_server.\*=.\***/{x;s/^/./;/^\.\{**2**\}$/{x;s/.\*/**tracker\_server = 192.168.100.63:22122**/;x};x;}' /etc/fdfs/storage.conf

sed -i '/**^tracker\_server.\*=.\***/{x;s/^/./;/^\.\{**3**\}$/{x;s/.\*/**tracker\_server = 192.168.100.63:22122**/;x};x;}' /etc/fdfs/storage.conf

sed -i "s/^http.server\_port=.\*/http.server\_port=8888/g" /etc/fdfs/storage.conf

# 客户端配置

sed -i "s/^base\_path.\*=.\*/base\_path = \/home\/dfs/g" /etc/fdfs/client.conf

sed -i '0,/^tracker\_server.\*=.\*/s//tracker\_server = 192.168.100.62:22122/' /etc/fdfs/client.conf

sed -i '/**^tracker\_server.\*=.\***/{x;s/^/./;/^\.\{**2**\}$/{x;s/.\*/**tracker\_server = 192.168.100.63:22122**/;x};x;}' /etc/fdfs/client.conf

sed -i '/**^tracker\_server.\*=.\***/{x;s/^/./;/^\.\{**3**\}$/{x;s/.\*/**tracker\_server = 192.168.100.64:22122**/;x};x;}' /etc/fdfs/client.conf

# 启动服务

/etc/init.d/fdfs\_trackerd start

/etc/init.d/fdfs\_storaged start

chkconfig fdfs\_trackerd on

chkconfig fdfs\_storaged on

# 客户端上传文件

fdfs\_upload\_file /etc/fdfs/client.conf /home/Nginx-GUI-For-Linux\_X64\_v1.6.zip

# 上传完成后的回显结果（等待Nginx配置完成后测试）

group1/M00/00/00/wKhkPV5qB3uAe8ISBrIcyZNcp5E1.6.zip

# 配置Nginx访问

sed -i '0,/^tracker\_server.\*=.\*/s//tracker\_server = 192.168.100.62:22122/' /etc/fdfs/mod\_fastdfs.conf

sed -i '/**^tracker\_server.\*=.\***/{x;s/^/./;/^\.\{**2**\}$/{x;s/.\*/**tracker\_server = 192.168.100.63:22122**/;x};x;}' /etc/fdfs/mod\_fastdfs.conf

sed -i '/**^tracker\_server.\*=.\***/{x;s/^/./;/^\.\{**3**\}$/{x;s/.\*/**tracker\_server = 192.168.100.64:22122**/;x};x;}' /etc/fdfs/mod\_fastdfs.conf

sed -i "s/^url\_have\_group\_name.\*=.\*/url\_have\_group\_name = true/g" /etc/fdfs/mod\_fastdfs.conf

sed -i "s/^store\_path0.\*=.\*/store\_path0 = \/home\/dfs/g" /etc/fdfs/mod\_fastdfs.conf

# 添加8888虚拟机

vi /usr/local/nginx/conf/nginx.conf

# 配置内容：

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;

}

}

# 测试下载

wget <http://192.168.100.61:8888/group1/M00/00/00/wKhkPV5qB3uAe8ISBrIcyZNcp5E1.6.zip>

# 2 容器部署（编写Dockerfile）

## 2.1 虚拟机列表

|  |  |
| --- | --- |
| 192.168.100.62 | cent7\_cntr\_fdfs1 |
| 192.168.100.63 | cent7\_cntr\_fdfs2 |
| 192.168.100.64 | cent7\_cntr\_fdfs3 |

## 2.2 docker安装

yum install -y telnet yum-utils device-mapper-persistent-data lvm2

yum-config-manager --add-repo <http://mirrors.aliyun.com/docker-ce/linux/centos/docker-ce.repo>

yum makecache fast

yum -y update

yum -y install <http://mirror.centos.org/centos/7/extras/x86_64/Packages/container-selinux-2.107-3.el7.noarch.rpm>

yum -y install containerd.io.x86\_64 docker-ce.x86\_64 docker-ce-cli.x86\_64

mkdir -p /etc/docker /etc/systemd/system/docker.service.d

cat > /etc/docker/daemon.json <<EOF

{

"exec-opts": ["native.cgroupdriver=systemd"],

"log-driver": "json-file",

"log-opts": {

"max-size": "100m"

},

"storage-driver": "overlay2",

"storage-opts": [

"overlay2.override\_kernel\_check=true"

],

"graph": "/data/docker",

"registry-mirrors": ["https://ya78xgj1.mirror.aliyuncs.com"]

}

EOF

systemctl daemon-reload

systemctl enable docker

systemctl restart docker

docker images

shutdown now

## 2.3 虚拟机复制

# 克隆虚拟机

vmrun clone /home/vboxes/cent7\_cntr\_fdfs1/cent7\_cntr\_fdfs1.vmx /home/vboxes/cent7\_cntr\_fdfs2/cent7\_cntr\_fdfs2.vmx full -cloneName=cent7\_cntr\_fdfs2

vmrun clone /home/vboxes/cent7\_cntr\_fdfs1/cent7\_cntr\_fdfs1.vmx /home/vboxes/cent7\_cntr\_fdfs3/cent7\_cntr\_fdfs3.vmx full -cloneName=cent7\_cntr\_fdfs3

# 启动虚拟机

vmrun -T ws start /home/vboxes/cent7\_cntr\_fdfs1/cent7\_cntr\_fdfs1.vmx

vmrun -T ws start /home/vboxes/cent7\_cntr\_fdfs2/cent7\_cntr\_fdfs2.vmx

vmrun -T ws start /home/vboxes/cent7\_cntr\_fdfs3/cent7\_cntr\_fdfs3.vmx

## 2.4 配置dockerfile

# 在三台虚机上分别执行

mkdir -p /usr/local/docker/fastdfs/env

cd /usr/local/docker/fastdfs/env

# 将tools.zip拷贝到此目录下，tools.zip为提前下载好的源码、配置文件等



# 编写DockerFile

cat > /usr/local/docker/fastdfs/env/Dockerfile <<EOF

FROM centos

MAINTAINER hugh

#安装编译环境

RUN yum install git gcc gcc-c++ make automake autoconf libtool pcre pcre-devel zlib zlib-devel openssl-devel wget vim unzip -y

RUN mkdir -p /home/dfs

WORKDIR /usr/local/src

#复制并解压缩各种源码

ADD tools.zip /usr/local/src

RUN unzip tools.zip

# 安装 libfastcommon

WORKDIR /usr/local/src/libfastcommon

RUN ./make.sh && ./make.sh install

# 安装 FastDFS

WORKDIR /usr/local/src/fastdfs

RUN ./make.sh && ./make.sh install

RUN cp /usr/local/src/tracker.conf /etc/fdfs/tracker.conf

RUN cp /usr/local/src/storage.conf /etc/fdfs/storage.conf

RUN cp /usr/local/src/client.conf /etc/fdfs/client.conf

RUN cp /usr/local/src/fastdfs/conf/http.conf /etc/fdfs/

RUN cp /usr/local/src/fastdfs/conf/mime.types /etc/fdfs/

#安装fastdfs-nginx-module

RUN cp /usr/local/src/mod\_fastdfs.conf /etc/fdfs

#安装nginx

WORKDIR /usr/local/src/nginx-1.15.4

RUN ./configure --add-module=/usr/local/src/fastdfs-nginx-module/src/

RUN make && make install

RUN cp /usr/local/src/nginx.conf /usr/local/nginx/conf

RUN cp /usr/local/src/start.sh /usr/local/bin/

ENTRYPOINT ["/usr/local/bin/start.sh"]

WORKDIR /

EXPOSE 8888

EOF

# 创建docker-compose配置文件

cd /usr/local/docker/fastdfs

cat > docker-compose.yml <<EOF

version: '3.1'

services:

fastdfs:

build: env

restart: always

container\_name: fastdfs

volumes:

- ./dfs:/home/dfs

network\_mode: host

EOF

# 执行docker-compose

docker-compose build

# 启动容器

docker-compose up -d

# 进入容器

docker exec -it fastdfs /bin/bash

# 容器内执行

/usr/bin/fdfs\_upload\_file /etc/fdfs/client.conf /usr/local/src/fastdfs/INSTALL

# 检测集群

/usr/bin/fdfs\_monitor /etc/fdfs/storage.conf

# 3 容器部署（模板安装）

# docker安装参见2.2节

# 拉取fastdfs镜像

docker pull mypjb/fastdfs

# 运行并进入容器

docker run --add-host fastdfs.net:192.168.100.65 --name fastdfs --net=host -e TRACKER\_ENABLE=1 -e NGINX\_PORT=81 -v /home/fastdfs:/storage/fastdfs -it mypjb/fastdfs

# 退出容器

exit

# 在宿主机上重启容器

docker restart fastdfs

# 添加防火墙策略

firewall-cmd --zone=public --add-port=81/tcp --permanent;firewall-cmd --reload;

# 切换到存储目录

cd /home/fastdfs/data/00/00

# 拷贝day.jpg到当前目录

cp /usr/share/backgrounds/day.jpg .

# 改名，不改名无法成功

mv day.jpg wKgByFmn1iGAUsF1AAL4cszpkW0032.jpg

# 测试

wget http://192.168.100.65:81/M00/00/00/wKgByFmn1iGAUsF1AAL4cszpkW0032.jpg

# 4 容器部署（容器内手动安装）

# 拉取centos最新镜像

**docker pull centos:7.7.1908**

docker images

# 运行容器[[4]](#footnote-4)

**docker run -itd --privileged --name="dfs1" --network host -v dfs:/home/dfs centos:7.7.1908 /usr/sbin/init**

**docker exec -it dfs1 /bin/bash**

# 在容器内执行1.1、1.2节内容，内容略

# 从容器提交一个新的镜像，并复制到另外两个节点

**docker commit dfs1 fdfs\_cent7:v0.1**

**docker save -o dfs\_img.tar fdfs\_cent7:v0.1**

**gzip dfs\_img.tar**

**scp -rp dfs\_img.tar.gz 192.168.100.63:/home**

**scp -rp dfs\_img.tar.gz 192.168.100.64:/home**

# 关闭所有节点上的防火墙

**systemctl stop firewalld**

**systemctl disable firewalld**

# 在另外两个节点上执行

**docker load -i dfs\_img.tar.gz**

**docker run -itd --privileged --name="dfs2" --network host -v dfs:/home/dfs fdfs\_cent7:v0.1 /usr/sbin/init**

**docker run -itd --privileged --name="dfs3" --network host -v dfs:/home/dfs fdfs\_cent7:v0.1 /usr/sbin/init**

# 检查所有服务状态

/usr/bin/fdfs\_monitor /etc/fdfs/storage.conf

# 测试：修改/etc/fdfs/client.conf，只保留第一个tracker节点，在1号节点上传文件

fdfs\_upload\_file /etc/fdfs/client.conf Wildlife.wmv

# 测试：在2号节点下载

wget http://192.168.100.63:8888/group1/M00/00/00/wKhkPl5xUByAXvIGAZB7imLO1SE958.wmv

# 5 去重存储

## 5.1 安装berkeleydb

# 登陆Oracle，获取下载地址类似

wget <https://download.oracle.com/otn/berkeley-db/db-6.2.38.tar.gz?AuthParam=1584512779_e725b87f84da862e9fb2e2ce0edb3d48>

cd db-6.2.38/build\_unix

../dist/configure --prefix=/usr/local/db-6.2.38

make && make install

## 5.2 安装FastDHT

unzip fastdht-master.zip

cd fastdht-master

sed -i "s/^CFLAGS=/CFLAGS='-Wall -D\_FILE\_OFFSET\_BITS=64 -D\_GNU\_SOURCE -I\/usr\/local\/db-6.2.38\/include\/ -L\/usr\/local\/db-6.2.38/lib\/'/g" make.sh

./make.sh

./make.sh install

mkdir -p /home/dht

sed -i "s/^base\_path.\*=.\*/base\_path = \/home\/dht/g" /etc/fdht/fdht\_client.conf

# 启用持久性连接

sed -i "s/^keep\_alive.\*=.\*/keep\_alive = 1/g" /etc/fdht/fdht\_client.conf

# 配置fdht\_servers.conf

cat > /etc/fdht/fdht\_servers.conf <<EOF

group\_count = 1

group0 = 192.168.100.62:11411

group0 = 192.168.100.63:11411

group0 = 192.168.100.64:11411

EOF

# 配置fdhtd.conf

sed -i "s/^port.\*=.\*/port = 11411/g" /etc/fdht/fdhtd.conf

sed -i "s/^base\_path.\*=.\*/base\_path = \/home\/dht/g" /etc/fdht/fdhtd.conf

sed -i "s/^cache\_size.\*=.\*/cache\_size = 64MB/g" /etc/fdht/fdhtd.conf

sed -i "s/^##include fdht\_servers.conf/#include fdht\_servers.conf/g" /etc/fdht/fdhtd.conf

# 配置storaged.conf

# 检测重复文件

sed -i "s/^check\_file\_duplicate.\*=.\*/check\_file\_duplicate = 1/g" /etc/fdfs/storage.conf

# 在FastDHT中的命名空间

sed -i "s/^key\_namespace.\*=.\*/key\_namespace = FastDFS/g" /etc/fdfs/storage.conf

# 长连接

sed -i "s/^keep\_alive.\*=.\*/keep\_alive = 1/g" /etc/fdfs/storage.conf

sed -i "s/^\##include .\*fdht\_servers.conf.\*/#include \/etc\/fdht\/fdht\_servers.conf/g" /etc/fdfs/storage.conf

# 拷贝共享库

cp /usr/local/db-6.2.38/lib/libdb-6.2.so /usr/lib/

cat >> /etc/ld.so.conf <<EOF

include /usr/local/lib

EOF

/sbin/ldconfig

# 启动fdhtd，以下两个命令均可

fdhtd /etc/fdht/fdhtd.conf

fdhtd /etc/fdht/fdhtd.conf restart

# 检查端口是否开放

ss -lntup | grep 11411

1. Initscripts软件包中含有/etc/init.d/functions，缺少此文件会造成/etc/init.d/fdfs\_trackerd status执行报错：/etc/init.d/fdfs\_trackerd: line 46: status: command not found [↑](#footnote-ref-1)
2. 跟踪服务器，主要做调度工作，起负载均衡的作用。在内存中记录集群中所有存储组和存储服务器的状态信息，是客户端和数据服务器交互的枢纽。相比GFS中的master更为精简，不记录文件索引信息，占用的内存量很少。 [↑](#footnote-ref-2)
3. 存储服务器（又称：存储节点或数据服务器） ，文件和文件属性（metadata）都保存到存储服务器上。Storage server直接利用OS的文件系统调用管理文件。 [↑](#footnote-ref-3)
4. 采用网络类型host，和宿主机共享同一网络，并且无需指定-p参数人工指定映射地址，-v参数指定目录映射关系。 [↑](#footnote-ref-4)