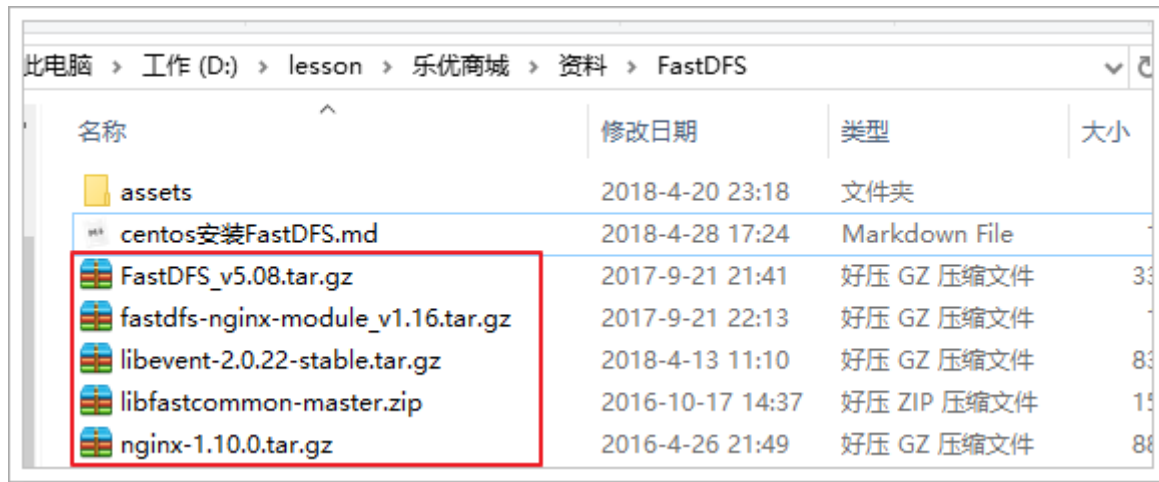


# 1. Centos下安装FastDFS

## 1.1 上传

将课前资料中的文件上传到linux下的 `/home/leyou/fdfs` 目录:



## 1.2 安装依赖

FastDFS运行需要一些依赖，在课前资料提供的虚拟中已经安装好了这些依赖，如果大家想要从头学习，可以按下面方式安装：

### 1.2.1 安装GCC依赖

GCC用来对C语言代码进行编译运行，使用yum命令安装：

```
sudo yum -y install gcc
```

### 1.2.2 安装unzip工具

unzip工具可以帮我们对压缩包进行解压

```
sudo yum install -y unzip zip
```

### 1.2.3 安装libevent

```
sudo yum -y install libevent
```

### 1.2.4 安装Nginx所需依赖

```
sudo yum -y install pcre pcre-devel zlib zlib-devel openssl openssl-devel
```

## 1.2.5 安装libfastcommon-master

这个没有yum包，只能通过编译安装：

- 解压刚刚上传的 libfastcommon-master.zip

```
tar -xvf libfastcommon-master.zip
```

- 进入解压完成的目录：

```
cd libfastcommon-master
```

- 编译并且安装：

```
sudo ./make.sh  
sudo ./makesh install
```

到这里为止，所有依赖都已经安装完毕，接下来我们安装FastDFS：

## 1.3 安装FastDFS

### 1.3.1 编译安装

这里我们也采用编译安装，步骤与刚才的编译安装方式一样：

- 解压

```
tar -xvf FastDFS_v5.08.tar.gz
```

- 进入目录

```
cd FastDFS
```

- 编译并安装

```
sudo ./make.sh  
sudo ./make.sh install
```

- 校验安装结果

1) 安装完成，我们应该能在 `/etc/init.d/` 目录，通过命令 `ll /etc/init.d/ | grep fdfs` 看到FastDFS提供的启动脚本：

```
[root@localhost init.d]# ll /etc/init.d/ | grep fdfs
-rwxr-xr-x. 1 root root 918 9月 21 21:46 fdfs_storaged
-rwxr-xr-x. 1 root root 920 9月 21 21:46 fdfs_trackerd
```

其中：

- `fdfs_trackerd` 是tracker启动脚本
- `fdfs_storaged` 是storage启动脚本

2) 我们可以在 `/etc/fdfs` 目录，通过命令查看到以下配置文件模板：

```
总用量 20
-rw-r--r--. 1 root root 1461 4月 18 17:23 client.conf.sample
-rw-r--r--. 1 root root 7927 4月 18 17:23 storage.conf.sample
-rw-r--r--. 1 root root 7200 4月 18 17:23 tracker.conf.sample
```

其中：

- `tarcker.conf.sample` 是tracker的配置文件模板
- `storage.conf.sample` 是storage的配置文件模板
- `client.conf.sample` 是客户端的配置文件模板

### 1.3.2 启动tracker

FastDFS的tracker和storage在刚刚的安装过程中，都已经被安装了，因此我们安装这两种角色的方式是一样的。不同的是，两种需要不同的配置文件。

我们要启动tracker，就修改刚刚看到的 `tarcker.conf`，并且启动 `fdfs_trackerd` 脚本即可。

- 编辑tracker配置

首先我们将模板文件进行赋值和重命名：

```
sudo cp tracker.conf.sample tracker.conf
sudo vim tracker.conf
```

打开 `tracker.conf`，修改 `base_path` 配置：

```
base_path=/leyou/fdfs/tracker # tracker的数据和日志存放目录
```

- 创建目录

刚刚配置的目录可能不存在，我们创建出来

```
sudo mkdir -p /leyou/fdfs/tracker
```

- 启动tracker

我们可以使用 `sh /etc/init.d/fdfs_trackerd` 启动，不过安装过程中，fdfs已经被设置为系统服务，我们可以采用熟悉的服务启动方式：

```
sudo service fdfs_trackerd start # 启动fdfs_trackerd服务，停止用stop
```

另外，我们可以通过以下命令，设置tracker开机启动：

```
sudo chkconfig fdfs_trackerd on
```

### 1.3.3 启动storage

我们要启动tracker，就修改刚刚看到的 `tracker.conf`，并且启动 `fdfs_trackerd` 脚本即可。

- 编辑storage配置

首先我们将模板文件进行赋值和重命名：

```
sudo cp storage.conf.sample storage.conf
sudo vim storage.conf
```

打开 `storage.conf`，修改 `base_path` 配置：

```
base_path=/leyou/fdfs/storage # storage的数据和日志存放目录
store_path0=/leyou/fdfs/storage # storage的上传文件存放路径
tracker_server=192.168.56.101:22122 # tracker的地址
```

- 创建目录

刚刚配置的目录可能不存在，我们创建出来

```
sudo mkdir -p /leyou/fdfs/storage
```

- 启动storage

我们可以使用 `sh /etc/init.d/fdfs_storaged` 启动，同样我们可以用服务启动方式：

```
sudo service fdfs_storaged start # 启动fdfs_storaged服务，停止用stop
```

另外，我们可以通过以下命令，设置tracker开机启动：

```
sudo chkconfig fdfs_storaged on
```

最后，通过 `ps -ef | grep fdfs` 查看进程：

```
[leyou@leyou fdfs]$  
[leyou@leyou fdfs]$ ps -ef | grep fdfs  
root      1931      1  0 17:39 ?        00:00:00 /usr/bin/fdfs_trackerd /etc/fdfs/tracker.conf  
root      1972      1 75 17:46 ?        00:00:05 /usr/bin/fdfs_storaged /etc/fdfs/storage.conf  
leyou     1974    1484  0 17:46 pts/1    00:00:00 grep fdfs  
[leyou@leyou fdfs]$  
[leyou@leyou fdfs]$
```

## 1.4 安装Nginx及FastDFS模块

### 1.4.1 FastDFS的Nginx模块

- 解压

```
tar -xvf fastdfs-nginx-module_v1.16.tar.gz
```

- 配置config文件

```
# 进入配置目录  
cd /home/leyou/fdfs/fastdfs-nginx-module/src/  
# 修改配置  
vim config  
# 执行下面命令（将配置中的/usr/local改为/usr）：  
:%s+/usr/local/+/usr/+g
```

- 配置mod\_fastdfs.conf

```
# 将src目录下的mod_fastdfs.conf复制到 /etc/fdfs目录：  
sudo cp mod_fastdfs.conf /etc/fdfs/  
# 编辑该文件  
sudo vim /etc/fdfs/mod_fastdfs.conf
```

- 修改一下配置：

<code>connect_timeout=10</code>	# 客户端访问文件连接超时时长（单位：秒）
<code>tracker_server=192.168.56.101:22122</code>	# tracker服务IP和端口
<code>url_have_group_name=true</code>	# 访问链接前缀加上组名
<code>store_path0=/leyou/fdfs/storage</code>	# 文件存储路径

- 复制 FastDFS的部分配置文件到/etc/fdfs目录

```
cd /home/leyou/fdfs/FastDFS/conf/
cp http.conf mime.types /etc/fdfs/
```

## 1.4.2 安装Nginx

- 解压

```
tar -xvf nginx-1.10.0.tar.gz
```

- 配置

```
sudo ./configure --prefix=/opt/nginx --sbin-path=/usr/bin/nginx --add-module=
```

- 编译安装

```
sudo make && sudo make install
```

- 配置nginx整合fastdfs-module模块

我们需要修改nginx配置文件，在/opt/nginx/config/nginx.conf文件中：

```
sudo vim /opt/nginx/conf/nginx.conf
```

将文件中，原来的 `server 80{ ...}` 部分代码替换为如下代码：

```
server {
    listen      80;
    server_name image.taotao.com;
```

```

# 监听域名中带有group的，交给FastDFS模块处理
location ~/group([0-9]+)/ {
    ngx_fastdfs_module;
}

location / {
    root    html;
    index  index.html index.htm;
}

error_page    500 502 503 504    /50x.html;
location = /50x.html {
    root    html;
}
}

```

- 启动

```

nginx # 启动
nginx -s stop # 停止
nginx -s reload # 重新加载配置

```

- 设置nginx开机启动

创建一个开机启动的脚本：

```

vim /etc/init.d/nginx

```

添加以下内容：

```

#!/bin/sh
#
# nginx - this script starts and stops the nginx daemon
#
# chkconfig:   - 85 15
# description: NGINX is an HTTP(S) server, HTTP(S) reverse \
#               proxy and IMAP/POP3 proxy server
# processname: nginx
# config:      /etc/nginx/nginx.conf
# config:      /etc/sysconfig/nginx
# pidfile:     /var/run/nginx.pid

# Source function library.
. /etc/rc.d/init.d/functions

# Source networking configuration.

```

```

. /etc/sysconfig/network

# Check that networking is up.
[ "$NETWORKING" = "no" ] && exit 0

nginx="/usr/bin/nginx"
prog=$(basename $nginx)

NGINX_CONF_FILE="/opt/nginx/conf/nginx.conf"

[ -f /etc/sysconfig/nginx ] && . /etc/sysconfig/nginx

lockfile=/var/lock/subsys/nginx

make_dirs() {
    # make required directories
    user=`$nginx -V 2>&1 | grep "configure arguments:.*--user=" | sed 's/^[^*]*'
    if [ -n "$user" ]; then
        if [ -z "`grep $user /etc/passwd`" ]; then
            useradd -M -s /bin/nologin $user
        fi
        options=`$nginx -V 2>&1 | grep 'configure arguments:'`
        for opt in $options; do
            if [ `echo $opt | grep '.*-temp-path'` ]; then
                value=`echo $opt | cut -d "=" -f 2`
                if [ ! -d "$value" ]; then
                    # echo "creating" $value
                    mkdir -p $value && chown -R $user $value
                fi
            fi
        done
    fi
}

start() {
    [ -x $nginx ] || exit 5
    [ -f $NGINX_CONF_FILE ] || exit 6
    make_dirs
    echo -n "Starting $prog: "
    daemon $nginx -c $NGINX_CONF_FILE
    retval=$?
    echo
    [ $retval -eq 0 ] && touch $lockfile
    return $retval
}

stop() {
    echo -n "Stopping $prog: "
    killproc $prog -QUIT
    retval=$?
    echo
    [ $retval -eq 0 ] && rm -f $lockfile
}

```



```

    return $retval
}

restart() {
    configtest || return $?
    stop
    sleep 1
    start
}

reload() {
    configtest || return $?
    echo -n $"Reloading $prog: "
    killproc $nginx -HUP
    RETVAL=$?
    echo
}

force_reload() {
    restart
}

configtest() {
    $nginx -t -c $NGINX_CONF_FILE
}

rh_status() {
    status $prog
}

rh_status_q() {
    rh_status >/dev/null 2>&1
}

case "$1" in
    start)
        rh_status_q && exit 0
        $1
        ;;
    stop)
        rh_status_q || exit 0
        $1
        ;;
    restart|configtest)
        $1
        ;;
    reload)
        rh_status_q || exit 7
        $1
        ;;
    force-reload)
        force_reload

```

```

        ;;
    status)
        rh_status
        ;;
    condrestart|try-restart)
        rh_status_q || exit 0
        ;;
    *)
        echo $"Usage: $0 {start|stop|status|restart|condrestart|try-restart|r
        exit 2
esac

```

- 修改文件权限，并加入服务列表

```

# 修改权限
chmod 777 /etc/init.d/nginx
# 添加到服务列表
chkconfig --add /etc/init.d/nginx

```

- 设置开机启动

```
chkconfig nginx on
```