# FastDFS安装配置手册

XjAcKs

xjacks@gmail.com

2014/3/27

目录

[FastDFS安装配置手册 1](#_Toc383683153)

[一、 安装 3](#_Toc383683154)

[(一) 下载FastDFS安装包 3](#_Toc383683155)

[(二) 安装tracker 3](#_Toc383683156)

[1. 安装 3](#_Toc383683157)

[2. 配置 5](#_Toc383683158)

[3. 运行 5](#_Toc383683159)

[(三) 安装storage 6](#_Toc383683160)

[1. 安装 6](#_Toc383683161)

[2. 配置 8](#_Toc383683162)

[3. 运行 8](#_Toc383683163)

[(四) 在storage上安装nginx 10](#_Toc383683164)

[1. 安装 10](#_Toc383683165)

[2. 配置 12](#_Toc383683166)

[3. 运行 13](#_Toc383683167)

[(五) 在tracker上安装nginx 14](#_Toc383683168)

[1. 安装 15](#_Toc383683169)

[2. 配置 16](#_Toc383683170)

[3. 运行 18](#_Toc383683171)

[二、 使用 20](#_Toc383683172)

[1. 上传文件 20](#_Toc383683173)

[2. 下载文件 21](#_Toc383683174)

[3. 监视服务器资源 21](#_Toc383683175)

[4. 其他功能 27](#_Toc383683176)

[三、 附录 28](#_Toc383683177)

## 安装

### 下载FastDFS安装包

FastDFS官方论坛：<http://www.csource.org>

下载1：<http://sourceforge.net/projects/fastdfs/files/>

下载2：<https://code.google.com/p/fastdfs/downloads/list>

本手册使用CentOS 6.5 x86\_64版操作系统，按照以下网络结构进行部署：



所需要下载的压缩包有：

FastDFS源代码：FastDFS\_v5.01.tar.gz

nginx模块源代码：fastdfs-nginx-module\_v1.15.tar.gz

nginx服务器源代码：nginx-1.4.7.tar.gz

nginx cache purge插件源代码：ngx\_cache\_purge-2.1.tar.gz

nginx依赖的pcre库源代码：pcre-8.34.tar.gz

nginx依赖的zlib库源代码：zlib-1.2.8.tar.gz

### 安装tracker

### 安装

首先在172.16.1.202上安装FastDFS tracker，使用FastDFS\_v5.01.tar.gz源代码包。

可以直接从Linux系统下载源代码包并解压安装。如果是在Windows系统下载，可以在Linux系统中挂载Windows的共享目录并且将源代码包复制进Linux系统内。挂载命令如下

|  |
| --- |
| [root@centos-db01 ~]# mount -o ro //Windows主机的IP地址/共享目录名 /挂载目标目录 |

* 首先将代码包复制到系统的/usr/local/src内（这一步可选），然后使用tar命令解压

|  |
| --- |
| [root@tracker opt]# cp FastDFS\_v5.01.tar.gz /usr/local/src/  [root@tracker opt]# cd /usr/local/src/  [root@tracker src]# tar zxf FastDFS\_v5.01.tar.gz  [root@tracker src]# cd FastDFS  [root@tracker FastDFS]# ll  总用量 128  drwxrwxr-x. 3 500 500 4096 2月 6 18:07 client  drwxrwxr-x. 2 500 500 4096 2月 6 18:07 common  drwxrwxr-x. 2 500 500 4096 2月 6 18:07 conf  -rw-rw-r--. 1 500 500 35067 7月 28 2008 COPYING-3\_0.txt  -rw-rw-r--. 1 500 500 29691 2月 2 13:17 HISTORY  drwxrwxr-x. 2 500 500 4096 2月 6 18:07 init.d  -rw-rw-r--. 1 500 500 7639 1月 5 14:08 INSTALL  -rwxrwxr-x. 1 500 500 5531 12月 7 15:19 make.sh  drwxrwxr-x. 2 500 500 4096 2月 6 18:07 php\_client  -rw-rw-r--. 1 500 500 2380 7月 28 2008 README  -rwxrwxr-x. 1 500 500 1768 4月 12 2010 restart.sh  -rwxrwxr-x. 1 500 500 1680 4月 10 2010 stop.sh  drwxrwxr-x. 4 500 500 4096 2月 6 18:07 storage  drwxrwxr-x. 2 500 500 4096 2月 6 18:07 test  drwxrwxr-x. 2 500 500 4096 2月 6 18:07 tracker |

* 运行make.sh，确认make成功。期间如果有错误，可能会是缺少依赖的软件包，需安装后再次make。

|  |
| --- |
| [root@tracker FastDFS]# ./make.sh |

* 运行make.sh install，确认install成功。

|  |
| --- |
| [root@tracker FastDFS]# ./make.sh install |

安装完成后，所有可执行文件在/usr/local/bin下，以fdfs开头：

|  |
| --- |
| [root@tracker FastDFS]# ll /usr/local/bin/fdfs\*  -rwxr-xr-x. 1 root root 522918 3月 25 14:57 /usr/local/bin/fdfs\_appender\_test  -rwxr-xr-x. 1 root root 522871 3月 25 14:57 /usr/local/bin/fdfs\_appender\_test1  -rwxr-xr-x. 1 root root 514023 3月 25 14:57 /usr/local/bin/fdfs\_append\_file  -rwxr-xr-x. 1 root root 513433 3月 25 14:57 /usr/local/bin/fdfs\_crc32  -rwxr-xr-x. 1 root root 513967 3月 25 14:57 /usr/local/bin/fdfs\_delete\_file  -rwxr-xr-x. 1 root root 514377 3月 25 14:57 /usr/local/bin/fdfs\_download\_file  -rwxr-xr-x. 1 root root 514133 3月 25 14:57 /usr/local/bin/fdfs\_file\_info  -rwxr-xr-x. 1 root root 525064 3月 25 14:57 /usr/local/bin/fdfs\_monitor  -rwxr-xr-x. 1 root root 1179682 3月 25 14:57 /usr/local/bin/fdfs\_storaged  -rwxr-xr-x. 1 root root 529845 3月 25 14:57 /usr/local/bin/fdfs\_test  -rwxr-xr-x. 1 root root 527774 3月 25 14:57 /usr/local/bin/fdfs\_test1  -rwxr-xr-x. 1 root root 655809 3月 25 14:57 /usr/local/bin/fdfs\_trackerd  -rwxr-xr-x. 1 root root 514213 3月 25 14:57 /usr/local/bin/fdfs\_upload\_appender  -rwxr-xr-x. 1 root root 514999 3月 25 14:57 /usr/local/bin/fdfs\_upload\_file |

所有配置文件在/etc/fdfs下：

|  |
| --- |
| [root@tracker FastDFS]# ll /etc/fdfs/  总用量 60  -rw-r--r--. 1 root root 1461 3月 13 15:15 client.conf  -rw-r--r--. 1 root root 858 3月 13 15:15 http.conf  -rw-r--r--. 1 root root 31172 3月 13 15:15 mime.types  -rw-r--r--. 1 root root 3837 3月 25 10:03 mod\_fastdfs.conf  -rw-r--r--. 1 root root 7515 3月 24 10:36 storage.conf  -rw-r--r--. 1 root root 6989 3月 13 15:15 tracker.conf |

* 至此tracker安装完成。

### 配置

* 编辑配置文件目录下的tracker.conf，设置相关信息并保存。

|  |
| --- |
| [root@tracker FastDFS]# vim /etc/fdfs/tracker.conf |

一般只需改动以下几个参数即可：

disabled=false #启用配置文件

port=22122 #设置tracker的端口号

base\_path=/fdfs/tracker #设置tracker的数据文件和日志目录（需预先创建）

http.server\_port=8080 #设置http端口号

如需要进行性能调优，可以参照附录的配置文件的详细说明。

### 运行

* 运行tracker之前，先要把防火墙中对应的端口打开（本例中为22122）。

|  |
| --- |
| [root@tracker FastDFS]# iptables -I INPUT -p tcp -m state --state NEW -m tcp --dport 22122 -j ACCEPT  [root@tracker FastDFS]# /etc/init.d/iptables save  iptables：将防火墙规则保存到 /etc/sysconfig/iptables：[确定] |

* 启动tracker，确认启动是否成功。（查看是否对应端口22122是否开始监听）

|  |
| --- |
| [root@tracker FastDFS]# /usr/local/bin/fdfs\_trackerd /etc/fdfs/tracker.conf restart  [root@tracker FastDFS]# netstat -unltp | grep fdfs  tcp 0 0.0.0.0:22122 0.0.0.0:\* LISTEN 1766/fdfs\_trackerd |

也可查看tracker的日志是否启动成功或是否有错误。

|  |
| --- |
| [root@tracker FastDFS]# cat /fdfs/tracker/logs/trackerd.log |

* 设置开机自动启动。

|  |
| --- |
| [root@tracker FastDFS]# vim /etc/rc.d/rc.local |

将运行命令行添加进文件：/usr/local/bin/fdfs\_trackerd /etc/fdfs/tracker.conf restart

### 安装storage

### 安装

首先在172.16.1.203上安装FastDFS storage，使用FastDFS\_v5.01.tar.gz源代码包。

* 首先将代码包复制到系统的/usr/local/src内（这一步可选），然后使用tar命令解压

|  |
| --- |
| [root@storage1 opt]# cp FastDFS\_v5.01.tar.gz /usr/local/src/  [root@storage1 opt]# cd /usr/local/src/  [root@storage1 src]# tar zxf FastDFS\_v5.01.tar.gz  [root@storage1 src]# cd FastDFS  [root@storage1 FastDFS]# ll  总用量 128  drwxrwxr-x. 3 500 500 4096 2月 6 18:07 client  drwxrwxr-x. 2 500 500 4096 2月 6 18:07 common  drwxrwxr-x. 2 500 500 4096 2月 6 18:07 conf  -rw-rw-r--. 1 500 500 35067 7月 28 2008 COPYING-3\_0.txt  -rw-rw-r--. 1 500 500 29691 2月 2 13:17 HISTORY  drwxrwxr-x. 2 500 500 4096 2月 6 18:07 init.d  -rw-rw-r--. 1 500 500 7639 1月 5 14:08 INSTALL  -rwxrwxr-x. 1 500 500 5531 12月 7 15:19 make.sh  drwxrwxr-x. 2 500 500 4096 2月 6 18:07 php\_client  -rw-rw-r--. 1 500 500 2380 7月 28 2008 README  -rwxrwxr-x. 1 500 500 1768 4月 12 2010 restart.sh  -rwxrwxr-x. 1 500 500 1680 4月 10 2010 stop.sh  drwxrwxr-x. 4 500 500 4096 2月 6 18:07 storage  drwxrwxr-x. 2 500 500 4096 2月 6 18:07 test  drwxrwxr-x. 2 500 500 4096 2月 6 18:07 tracker |

* 运行make.sh，确认make成功。期间如果有错误，可能会是缺少依赖的软件包，需安装后再次make。

|  |
| --- |
| [root@storage1 FastDFS]# ./make.sh |

* 运行make.sh install，确认install成功。

|  |
| --- |
| [root@storage1 FastDFS]# ./make.sh install |

安装完成后，所有可执行文件在/usr/local/bin下，以fdfs开头：

|  |
| --- |
| [root@storage1 FastDFS]# ll /usr/local/bin/fdfs\*  -rwxr-xr-x. 1 root root 522918 3月 25 14:57 /usr/local/bin/fdfs\_appender\_test  -rwxr-xr-x. 1 root root 522871 3月 25 14:57 /usr/local/bin/fdfs\_appender\_test1  -rwxr-xr-x. 1 root root 514023 3月 25 14:57 /usr/local/bin/fdfs\_append\_file  -rwxr-xr-x. 1 root root 513433 3月 25 14:57 /usr/local/bin/fdfs\_crc32  -rwxr-xr-x. 1 root root 513967 3月 25 14:57 /usr/local/bin/fdfs\_delete\_file  -rwxr-xr-x. 1 root root 514377 3月 25 14:57 /usr/local/bin/fdfs\_download\_file  -rwxr-xr-x. 1 root root 514133 3月 25 14:57 /usr/local/bin/fdfs\_file\_info  -rwxr-xr-x. 1 root root 525064 3月 25 14:57 /usr/local/bin/fdfs\_monitor  -rwxr-xr-x. 1 root root 1179682 3月 25 14:57 /usr/local/bin/fdfs\_storaged  -rwxr-xr-x. 1 root root 529845 3月 25 14:57 /usr/local/bin/fdfs\_test  -rwxr-xr-x. 1 root root 527774 3月 25 14:57 /usr/local/bin/fdfs\_test1  -rwxr-xr-x. 1 root root 655809 3月 25 14:57 /usr/local/bin/fdfs\_trackerd  -rwxr-xr-x. 1 root root 514213 3月 25 14:57 /usr/local/bin/fdfs\_upload\_appender  -rwxr-xr-x. 1 root root 514999 3月 25 14:57 /usr/local/bin/fdfs\_upload\_file |

所有配置文件在/etc/fdfs下：

|  |
| --- |
| [root@storage1 FastDFS]# ll /etc/fdfs/  总用量 60  -rw-r--r--. 1 root root 1461 3月 13 15:15 client.conf  -rw-r--r--. 1 root root 858 3月 13 15:15 http.conf  -rw-r--r--. 1 root root 31172 3月 13 15:15 mime.types  -rw-r--r--. 1 root root 3837 3月 25 10:03 mod\_fastdfs.conf  -rw-r--r--. 1 root root 7515 3月 24 10:36 storage.conf  -rw-r--r--. 1 root root 6989 3月 13 15:15 tracker.conf |

* 至此storage安装完成。

### 配置

* 编辑配置文件目录下的storage.conf，设置相关信息并保存。

|  |
| --- |
| [root@storage1 FastDFS]# vim /etc/fdfs/storage.conf |

一般只需改动以下几个参数即可：

disabled=false #启用配置文件

group\_name=group1 #组名，根据实际情况修改

port=23000 #设置storage的端口号

base\_path=/fdfs/storage #设置storage的日志目录（需预先创建）

store\_path\_count=1 #存储路径个数，需要和store\_path个数匹配

store\_path0=/fdfs/storage #存储路径

tracker\_server=172.16.1.202:22122 #tracker服务器的IP地址和端口号

http.server\_port=8080 #设置http端口号

如需要进行性能调优，可以参照附录的配置文件的详细说明。

### 运行

* 运行storage之前，先要把防火墙中对应的端口打开（本例中为23000）。

|  |
| --- |
| [root@storage1 FastDFS]# iptables -I INPUT -p tcp -m state --state NEW -m tcp --dport 23000 -j ACCEPT  [root@storage1 FastDFS]# /etc/init.d/iptables save  iptables：将防火墙规则保存到 /etc/sysconfig/iptables：[确定] |

* 启动storage，会根据配置文件的设置自动创建多级存储目录，确认启动是否成功。（查看是否对应端口23000是否开始监听）

|  |
| --- |
| [root@storage1 FastDFS]# /usr/local/bin/fdfs\_storaged /etc/fdfs/storage.conf restart  data path: /fdfs/storage/data, mkdir sub dir...  mkdir data path: 00 ...  mkdir data path: 01 ...  mkdir data path: 02 ...  mkdir data path: 03 ...  ...  data path: /fdfs/storage/data, mkdir sub dir done.  [root@storage1 FastDFS]# netstat -unltp | grep fdfs  tcp 0 0.0.0.0:23000 0.0.0.0:\* LISTEN 1766/fdfs\_storaged |

也可查看storage的日志是否启动成功或是否有错误。

|  |
| --- |
| [root@storage1 FastDFS]# cat /fdfs/storage/logs/storaged.log |

确认启动成功后，可以运行fdfs\_monitor查看storage服务器是否已经登记到tracker服务器。

|  |
| --- |
| [root@storage1 FastDFS]# /usr/local/bin/fdfs\_monitor /etc/fdfs/storage.conf  [2014-03-26 01:51:20] DEBUG - base\_path=/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\_pool\_max\_idle\_time=3600s, use\_storage\_id=0, storage server id count: 0  server\_count=1, server\_index=0  tracker server is 172.16.1.202:22122  group count: 1  Group 1:  group name = group1  disk total space = 27789 MB  disk free space = 23920 MB  trunk free space = 0 MB  storage server count = 1  active server count = 1  storage server port = 23000  storage HTTP port = 8080  store path count = 1  subdir count per path = 256  current write server index = 0  current trunk file id = 0  Storage 1:  id = 172.16.1.203  ip\_addr = 172.16.1.203 ACTIVE  http domain =  version = 5.01  join time = 2014-03-26 01:48:19  up time = 2014-03-26 01:48:19  total storage = 27789 MB  free storage = 23920 MB  upload priority = 10  store\_path\_count = 1  subdir\_count\_per\_path = 256  storage\_port = 23000  storage\_http\_port = 8080  current\_write\_path = 0  source storage id=  if\_trunk\_server= 0  total\_upload\_count = 0  success\_upload\_count = 0  total\_append\_count = 0  success\_append\_count = 0  total\_modify\_count = 0  success\_modify\_count = 0  total\_truncate\_count = 0  success\_truncate\_count = 0  total\_set\_meta\_count = 0  success\_set\_meta\_count = 0  total\_delete\_count = 0  success\_delete\_count = 0  total\_download\_count = 0  success\_download\_count = 0  total\_get\_meta\_count = 0  success\_get\_meta\_count = 0  total\_create\_link\_count = 0  success\_create\_link\_count = 0  total\_delete\_link\_count = 0  success\_delete\_link\_count = 0  total\_upload\_bytes = 0  success\_upload\_bytes = 0  total\_append\_bytes = 0  success\_append\_bytes = 0  total\_modify\_bytes = 0  success\_modify\_bytes = 0  stotal\_download\_bytes = 0  success\_download\_bytes = 0  total\_sync\_in\_bytes = 0  success\_sync\_in\_bytes = 0  total\_sync\_out\_bytes = 0  success\_sync\_out\_bytes = 0  total\_file\_open\_count = 0  success\_file\_open\_count = 0  total\_file\_read\_count = 0  success\_file\_read\_count = 0  total\_file\_write\_count = 0  success\_file\_write\_count = 0  last\_heart\_beat\_time = 2014-03-26 01:51:03  last\_source\_update = 1970-01-01 08:00:00  last\_sync\_update = 1970-01-01 08:00:00  last\_synced\_timestamp = 1970-01-01 08:00:00 |

看到“172.16.1.203 ACTIVE”即可确认storage运行正常。

* 设置开机自动启动。

|  |
| --- |
| [root@storage1 FastDFS]# vim /etc/rc.d/rc.local |

将运行命令行添加进文件：/usr/local/bin/fdfs\_storaged /etc/fdfs/storage.conf restart

* 之后依次在172.16.1.204~208上全部安装上storage并确认运行正常。注意配置文件中group名参数需要根据实际情况调整，本例中group是这样分配的：

group1：172.16.1.203，172.16.1.204

group2：172.16.1.205，172.16.1.206

group3：172.16.1.207，172.16.1.208

另外每个group中所有storage的端口号必须一致。

### 在storage上安装nginx

在storage上安装的nginx主要为了提供http的访问服务，同时解决group中storage服务器的同步延迟问题。

### 安装

首先在172.16.1.203上安装nginx，使用nginx-1.4.7.tar.gz源代码包以及FastDFS的nginx插件fastdfs-nginx-module\_v1.15.tar.gz。

* 首先将代码包和插件复制到系统的/usr/local/src内（可选），然后使用tar命令解压

|  |
| --- |
| [root@storage1 opt]# cp nginx-1.4.7.tar.gz /usr/local/src  [root@storage1 opt]# cp fastdfs-nginx-module\_v1.15.tar.gz /usr/local/src  [root@storage1 opt]# cp pcre-8.34.tar.gz /usr/local/src  [root@storage1 opt]# cp zlib-1.2.8.tar.gz /usr/local/src  [root@storage1 opt]# cd /usr/local/src/  [root@storage1 src]# tar zxf nginx-1.4.7.tar.gz  [root@storage1 src]# tar zxf fastdfs-nginx-module\_v1.15.tar.gz  [root@storage1 src]# tar zxf pcre-8.34.tar.gz  [root@storage1 src]# tar zxf zlib-1.2.8.tar.gz  [root@storage1 src]# cd nginx-1.4.7  [root@storage1 nginx-1.4.7]# ll  总用量 592  drwxr-xr-x. 6 1001 1001 4096 3月 26 02:34 auto  -rw-r--r--. 1 1001 1001 225213 3月 18 21:17 CHANGES  -rw-r--r--. 1 1001 1001 343040 3月 18 21:17 CHANGES.ru  drwxr-xr-x. 2 1001 1001 4096 3月 26 02:34 conf  -rwxr-xr-x. 1 1001 1001 2369 3月 18 21:17 configure  drwxr-xr-x. 3 1001 1001 4096 3月 26 02:34 contrib  drwxr-xr-x. 2 1001 1001 4096 3月 26 02:34 html  -rw-r--r--. 1 1001 1001 1397 3月 18 21:17 LICENSE  drwxr-xr-x. 2 1001 1001 4096 3月 26 02:34 man  -rw-r--r--. 1 1001 1001 49 3月 18 21:17 README  drwxr-xr-x. 8 1001 1001 4096 3月 26 02:34 src |

* 运行./configure进行安装前的设置，主要设置安装路径、FastDFS插件模块目录、pcre库目录、zlib库目录。

如果提示错误，可能缺少依赖的软件包，需先安装依赖包，再次运行./configure

|  |
| --- |
| [root@storage1 nginx-1.4.7]# ./configure --prefix=/usr/local/nginx --add-module=/usr/local/src/fastdfs-nginx-module/src --with-pcre=/usr/local/src/pcre-8.34/ --with-zlib=/usr/local/src/zlib-1.2.8 |

* 运行make进行编译，确保编译成功。

|  |
| --- |
| [root@storage1 nginx-1.4.7]# make |

* 运行make install进行安装。

|  |
| --- |
| [root@storage1 nginx-1.4.7]# make install |

* 将FastDFS的nginx插件模块的配置文件copy到FastDFS配置文件目录。

|  |
| --- |
| [root@storage1 nginx-1.4.7]# cp /usr/local/src/fastdfs-nginx-module/src/mod\_fastdfs.conf /etc/fdfs/ |

安装完成后，nginx所有文件在/usr/local/nginx下：

|  |
| --- |
| [root@storage1 nginx-1.4.7]# ll /usr/local/nginx/  总用量 16  drwxr-xr-x. 2 root root 4096 3月 26 03:11 conf  drwxr-xr-x. 2 root root 4096 3月 26 03:11 html  drwxr-xr-x. 2 root root 4096 3月 26 03:11 logs  drwxr-xr-x. 2 root root 4096 3月 26 03:11 sbin |

* 至此nginx以及FastDFS的nginx插件模块安装完成。

### 配置

* 编辑/usr/local/nginx/conf配置文件目录下的nginx.conf，设置添加storage信息并保存。

|  |
| --- |
| [root@storage1 nginx-1.4.7]# vim /usr/local/nginx/conf/nginx.conf |

将server段中的listen端口号改为8080：

listen 8080;

在server段中添加：

location ~/group[1-3]/M00 {

root /fdfs/storage/data;

ngx\_fastdfs\_module;

}

* 编辑/etc/fdfs配置文件目录下的mod\_fastdfs.conf，设置storage信息并保存。

|  |
| --- |
| [root@storage1 nginx-1.4.7]# vim /etc/fdfs/mod\_fastdfs.conf |

一般只需改动以下几个参数即可：

base\_path=/fdfs/storage #保存日志目录

tracker\_server=172.16.1.202:22122 #tracker服务器的IP地址以及端口号

storage\_server\_port=23000 #storage服务器的端口号

group\_name=group1 #当前服务器的group名

url\_have\_group\_name = true #文件url中是否有group名

store\_path\_count=1 #存储路径个数，需要和store\_path个数匹配

store\_path0=/fdfs/storage #存储路径

http.need\_find\_content\_type=true #从文件扩展名查找文件类型（nginx时为true）

group\_count = 3 #设置组的个数

在末尾增加3个组的具体信息：

[group1]

group\_name=group1

storage\_server\_port=23000

store\_path\_count=1

store\_path0=/fdfs/storage

[group2]

group\_name=group2

storage\_server\_port=23000

store\_path\_count=1

store\_path0=/fdfs/storage

[group3]

group\_name=group3

storage\_server\_port=23000

store\_path\_count=1

store\_path0=/fdfs/storage

* 建立M00至存储目录的符号连接。

|  |
| --- |
| [root@storage1 nginx-1.4.7]# ln -s /fdfs/storage/data /fdfs/storage/data/M00  [root@storage1 nginx-1.4.7]# ll /fdfs/storage/data/M00  lrwxrwxrwx. 1 root root 19 3月 26 03:44 /fdfs/storage/data/M00 -> /fdfs/storage/data/ |

* 至此，nginx以及FastDFS插件模块设置完成。

### 运行

* 运行nginx之前，先要把防火墙中对应的端口打开（本例中为8080）。

|  |
| --- |
| [root@storage1 nginx-1.4.7]# iptables -I INPUT -p tcp -m state --state NEW -m tcp --dport 8080 -j ACCEPT  [root@storage1 nginx-1.4.7]# /etc/init.d/iptables save  iptables：将防火墙规则保存到 /etc/sysconfig/iptables：[确定] |

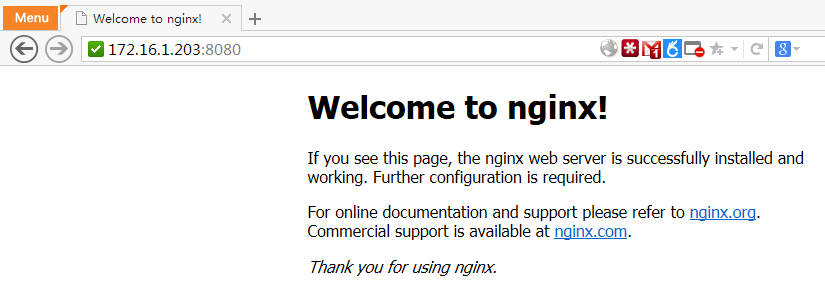
* 启动nginx，确认启动是否成功。（查看是否对应端口8080是否开始监听）

|  |
| --- |
| [root@storage1 nginx-1.4.7]# /usr/local/nginx/sbin/nginx  ngx\_http\_fastdfs\_set pid=40638  [root@storage1 nginx-1.4.7]# netstat -unltp | grep nginx  tcp 0 0.0.0.0:8080 0.0.0.0:\* LISTEN 40639/nginx |

也可查看nginx的日志是否启动成功或是否有错误。

|  |
| --- |
| [root@storage1 nginx-1.4.7]# cat /usr/local/nginx/logs/error.log  ngx\_http\_fastdfs\_process\_init pid=40640  [2014-03-26 03:47:17] INFO - local\_host\_ip\_count: 2, 127.0.0.1 172.16.1.203  [2014-03-26 03:47:17] INFO - fastdfs apache / nginx module v1.15, response\_mode=proxy, base\_path=/tmp, url\_have\_group\_name=1, group\_count=3, connect\_timeout=2, network\_timeout=30, tracker\_server\_count=1, if\_alias\_prefix=, local\_host\_ip\_count=2, need\_find\_content\_type=1, default\_content\_type=application/octet-stream, anti\_steal\_token=0, token\_ttl=0s, anti\_steal\_secret\_key length=0, token\_check\_fail content\_type=, token\_check\_fail buff length=0, load\_fdfs\_parameters\_from\_tracker=1, storage\_sync\_file\_max\_delay=86400s, use\_storage\_id=0, storage server id count=0, flv\_support=1, flv\_extension=flv  [2014-03-26 03:47:17] INFO - group 1. group\_name=group1, storage\_server\_port=23000, path\_count=1, store\_path0=/fdfs/storage  [2014-03-26 03:47:17] INFO - group 2. group\_name=group2, storage\_server\_port=23000, path\_count=1, store\_path0=/fdfs/storage  [2014-03-26 03:47:17] INFO - group 3. group\_name=group3, storage\_server\_port=23000, path\_count=1, store\_path0=/fdfs/storage |

在error.log中没有错误，既启动成功。可以打开浏览器，直接访问<http://172.16.1.203:8080>，查看是否弹出nginx欢迎页面。



之后依次在172.16.1.204~208上全部安装上nginx并确认运行正常。

* 设置开机自动启动。

|  |
| --- |
| [root@storage1 nginx-1.4.7]# vim /etc/rc.d/rc.local |

将运行命令行添加进文件：/usr/local/nginx/sbin/nginx

### 在tracker上安装nginx

在tracker上安装的nginx主要为了提供http访问的反向代理、负载均衡以及缓存服务。

### 安装

* 首先将代码包和插件复制到系统的/usr/local/src内（可选），然后使用tar命令解压

|  |
| --- |
| [root@tracker opt]# cp nginx-1.4.7.tar.gz /usr/local/src  [root@tracker opt]# cp ngx\_cache\_purge-2.1.tar.gz /usr/local/src  [root@tracker opt]# cp pcre-8.34.tar.gz /usr/local/src  [root@tracker opt]# cp zlib-1.2.8.tar.gz /usr/local/src  [root@tracker opt]# cd /usr/local/src/  [root@tracker src]# tar zxf nginx-1.4.7.tar.gz  [root@tracker src]# tar zxf ngx\_cache\_purge-2.1.tar.gz  [root@tracker src]# tar zxf pcre-8.34.tar.gz  [root@tracker src]# tar zxf zlib-1.2.8.tar.gz  [root@tracker src]# cd nginx-1.4.7  [root@tracker nginx-1.4.7]# ll  总用量 592  drwxr-xr-x. 6 1001 1001 4096 3月 26 02:34 auto  -rw-r--r--. 1 1001 1001 225213 3月 18 21:17 CHANGES  -rw-r--r--. 1 1001 1001 343040 3月 18 21:17 CHANGES.ru  drwxr-xr-x. 2 1001 1001 4096 3月 26 02:34 conf  -rwxr-xr-x. 1 1001 1001 2369 3月 18 21:17 configure  drwxr-xr-x. 3 1001 1001 4096 3月 26 02:34 contrib  drwxr-xr-x. 2 1001 1001 4096 3月 26 02:34 html  -rw-r--r--. 1 1001 1001 1397 3月 18 21:17 LICENSE  drwxr-xr-x. 2 1001 1001 4096 3月 26 02:34 man  -rw-r--r--. 1 1001 1001 49 3月 18 21:17 README  drwxr-xr-x. 8 1001 1001 4096 3月 26 02:34 src |

* 运行./configure进行安装前的设置，主要设置安装路径、nginx cache purge插件模块目录、pcre库目录、zlib库目录。

如果提示错误，可能缺少依赖的软件包，需先安装依赖包，再次运行./configure

|  |
| --- |
| [root@storage1 nginx-1.4.7]# ./configure --prefix=/usr/local/nginx --add-module=/usr/local/src/ngx\_cache\_purge-2.1 --with-pcre=/usr/local/src/pcre-8.34/ --with-zlib=/usr/local/src/zlib-1.2.8 |

* 运行make进行编译，确保编译成功。

|  |
| --- |
| [root@storage1 nginx-1.4.7]# make |

* 运行make install进行安装。

|  |
| --- |
| [root@storage1 nginx-1.4.7]# make install |

* 至此nginx以及nginx cache purge插件模块安装完成。

### 配置

* 编辑/usr/local/nginx/conf配置文件目录下的nginx.conf，设置负载均衡以及缓存。

|  |
| --- |
| [root@storage1 nginx-1.4.7]# vim /usr/local/nginx/conf/nginx.conf |

worker\_processes 4; #根据CPU核心数而定

events {

worker\_connections 65535; #最大链接数

use epoll; #新版本的Linux可使用epoll加快处理性能

}

http {

#设置缓存参数

server\_names\_hash\_bucket\_size 128;

client\_header\_buffer\_size 32k;

large\_client\_header\_buffers 4 32k;

client\_max\_body\_size 300m;

sendfile on;

tcp\_nopush on;

proxy\_redirect off;

proxy\_set\_header Host $http\_host;

proxy\_set\_header X-Real-IP $remote\_addr;

proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

proxy\_connect\_timeout 90;

proxy\_send\_timeout 90;

proxy\_read\_timeout 90;

proxy\_buffer\_size 16k;

proxy\_buffers 4 64k;

proxy\_busy\_buffers\_size 128k;

proxy\_temp\_file\_write\_size 128k;

#设置缓存存储路径、存储方式、分配内存大小、磁盘最大空间、缓存期限

proxy\_cache\_path /var/cache/nginx/proxy\_cache levels=1:2 keys\_zone=http-cache:500m max\_size=10g inactive=30d;

proxy\_temp\_path /var/cache/nginx/proxy\_cache/tmp;

#设置group1的服务器

upstream fdfs\_group1 {

server 172.16.1.203:8080 weight=1 max\_fails=2 fail\_timeout=30s;

server 172.16.1.204:8080 weight=1 max\_fails=2 fail\_timeout=30s;

}

#设置group2的服务器

upstream fdfs\_group2 {

server 172.16.1.205:8080 weight=1 max\_fails=2 fail\_timeout=30s;

server 172.16.1.206:8080 weight=1 max\_fails=2 fail\_timeout=30s;

}

#设置group3的服务器

upstream fdfs\_group3 {

server 172.16.1.207:8080 weight=1 max\_fails=2 fail\_timeout=30s;

server 172.16.1.208:8080 weight=1 max\_fails=2 fail\_timeout=30s;

}

server {

#设置服务器端口

listen 8080;

#设置group1的负载均衡参数

location /group1/M00 {

proxy\_next\_upstream http\_502 http\_504 error timeout invalid\_header;

proxy\_cache http-cache;

proxy\_cache\_valid 200 304 12h;

proxy\_cache\_key $uri$is\_args$args;

proxy\_pass http://fdfs\_group1;

expires 30d;

}

#设置group2的负载均衡参数

location /group2/M00 {

#如果后端的服务器返回502、504、执行超时等错误。自动将请求转发到upsream负载均衡池中的另一台服务器，实现故障转移。

proxy\_next\_upstream http\_502 http\_504 error timeout invalid\_header;

proxy\_cache http-cache;

#对不同的HITP状悉码设置不同的缓有时间

proxy\_cache\_valid 200 304 12h;

#以域名、URI、参数组合成We缓存的Key伯、Nginx根据Key伯哈希，存储缓存内容到二级缓存目录内

proxy\_cache\_key $uri$is\_args$args;

proxy\_pass http://fdfs\_group2;

expires 30d;

}

#设置group3的负载均衡参数

location /group3/M00 {

proxy\_next\_upstream http\_502 http\_504 error timeout invalid\_header;

proxy\_cache http-cache;

proxy\_cache\_valid 200 304 12h;

proxy\_cache\_key $uri$is\_args$args;

proxy\_pass http://fdfs\_group3;

expires 30d;

}

#设置清除缓存的访问权限

location ~ /purge(/.\*) {

allow 127.0.0.1;

allow 172.16.1.0/24;

deny all;

proxy\_cache\_purge http-cache $1$is\_args$args;

}

* 至此，nginx以及nginx cache purge插件模块设置完成。

### 运行

运行nginx之前，先要把防火墙中对应的端口打开（本例中为8080）。

|  |
| --- |
| [root@tracker nginx-1.4.7]# iptables -I INPUT -p tcp -m state --state NEW -m tcp --dport 8080 -j ACCEPT  [root@tracker nginx-1.4.7]# /etc/init.d/iptables save  iptables：将防火墙规则保存到 /etc/sysconfig/iptables：[确定] |

启动nginx，确认启动是否成功。（查看是否对应端口8080是否开始监听）

|  |
| --- |
| [root@tracker nginx-1.4.7]# /usr/local/nginx/sbin/nginx  ngx\_http\_fastdfs\_set pid=40638  [root@tracker nginx-1.4.7]# netstat -unltp | grep nginx  tcp 0 0.0.0.0:8080 0.0.0.0:\* LISTEN 40639/nginx |

* 尝试上传一个文件到FastDFS，然后访问试试。先配置client.conf文件。

|  |
| --- |
| [root@tracker nginx-1.4.7]# vim /etc/fdfs/client.conf |

修改以下参数：

base\_path=/fdfs/tracker #日志存放路径

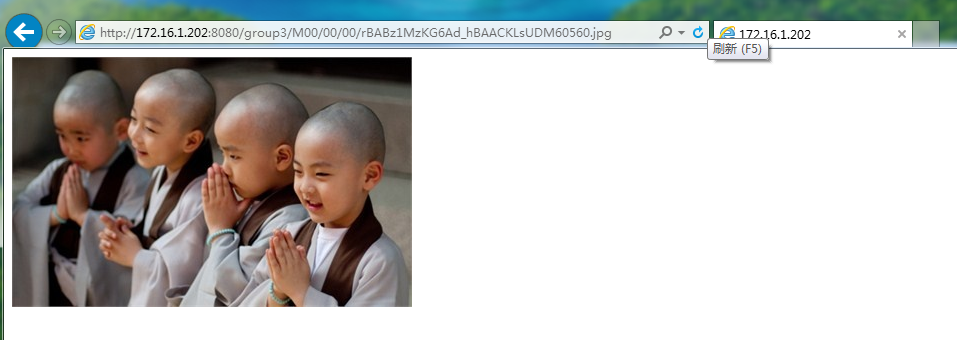
tracker\_server=172.16.1.202:22122 #tracker服务器IP地址和端口号

http.tracker\_server\_port=8080 #tracker服务器的http端口号

使用/usr/local/bin/fdfs\_upload\_file上传一个文件，程序会自动返回文件的URL。

|  |
| --- |
| [root@tracker nginx-1.4.7]# /usr/local/bin/fdfs\_upload\_file /etc/fdfs/client.conf /mnt/monk.jpg  group3/M00/00/00/rBABz1MzKG6Ad\_hBAACKLsUDM60560.jpg |

然后使用浏览器访问：



可以看到文件被正确读取出来了。

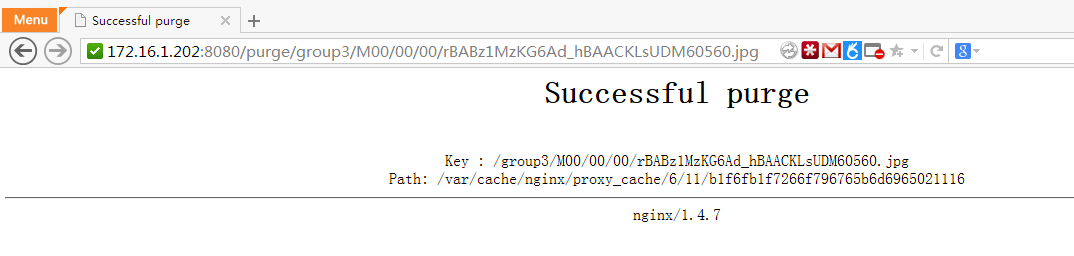
查看nginx的access.log日志，可以看到访问返回200成功。

|  |
| --- |
| [root@tracker nginx-1.4.7]# tail -n 10 -f /usr/local/nginx/logs/access.log  172.16.1.201 - - [26/Mar/2014:13:15:00 +0800] "GET /group3/M00/00/00/rBABz1MzKG6Ad\_hBAACKLsUDM60560.jpg HTTP/1.1" 200 35374 "-" "Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko" |

查看nginx的cache目录，可以看到已经生成了缓存文件。

|  |
| --- |
| [root@tracker nginx-1.4.7]# ll /var/cache/nginx/proxy\_cache/ -R  /var/cache/nginx/proxy\_cache/:  总用量 8  drwx------. 3 nobody nobody 4096 3月 26 12:57 6  drwxr-xr-x. 2 nobody root 4096 3月 26 13:14 tmp  /var/cache/nginx/proxy\_cache/6:  总用量 4  drwx------. 2 nobody nobody 4096 3月 26 13:08 11  /var/cache/nginx/proxy\_cache/6/11:  总用量 36  -rw-------. 1 nobody nobody 35686 3月 26 13:08 b1f6fb1f7266f796765b6d6965021116 |

如果要手动清除缓存，可以在文件URL之前加上purge：



* 设置开机自动启动。

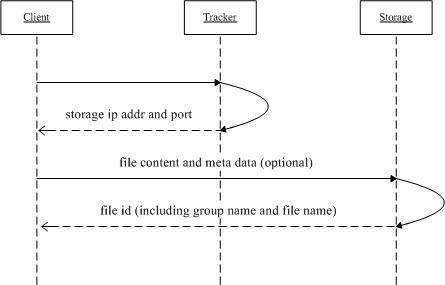
|  |
| --- |
| [root@tracker nginx-1.4.7]# vim /etc/rc.d/rc.local |

将运行命令行添加进文件：/usr/local/nginx/sbin/nginx

至此，tracker服务器上的http反向代理+负载均衡+缓存已经安装完成。

## 使用

### 上传文件

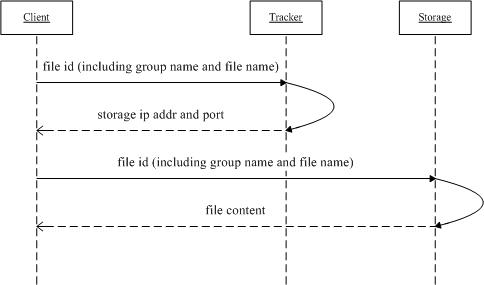


FastDFS提供了多种方式上传文件：

* 使用fdfs\_upload\_file上传（之前已经有演示）
* 使用C语言客户端接口上传
* 使用PHP客户端接口上传
* 使用Java客户端接口上传
* 使用Python客户端接口上传
* 使用.NET客户端接口上传

根据具体情况选择使用。上传均支持断点续传。

### 下载文件



* 使用fdfs\_download\_file通过tracker服务器下载
* 使用nginx通过http方式直接从storage下载（支持断点续传，作者推荐）

### 监视服务器资源

* 使用fdfs\_monitor查看tracker和所有group的运行情况。

|  |
| --- |
| [root@tracker tmp]# /usr/local/bin/fdfs\_monitor /etc/fdfs/client.conf  2014-03-26 15:01:07] DEBUG - base\_path=/fdfs/tracker, 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\_pool\_max\_idle\_time=3600s, use\_storage\_id=0, storage server id count: 0  server\_count=1, server\_index=0  tracker server is 172.16.1.202:22122  group count: 3  Group 1:  group name = group1  disk total space = 27789 MB  disk free space = 23825 MB  trunk free space = 0 MB  storage server count = 2  active server count = 2  storage server port = 23000  storage HTTP port = 8080  store path count = 1  subdir count per path = 256  current write server index = 1  current trunk file id = 0  Storage 1:  id = 172.16.1.203  ip\_addr = 172.16.1.203 ACTIVE  http domain =  version = 5.01  join time = 2014-03-26 01:48:19  up time =  total storage = 27789 MB  free storage = 23844 MB  upload priority = 10  store\_path\_count = 1  subdir\_count\_per\_path = 256  storage\_port = 23000  storage\_http\_port = 8080  current\_write\_path = 0  source storage id=  if\_trunk\_server= 0  total\_upload\_count = 1  success\_upload\_count = 1  total\_append\_count = 0  success\_append\_count = 0  total\_modify\_count = 0  success\_modify\_count = 0  total\_truncate\_count = 0  success\_truncate\_count = 0  total\_set\_meta\_count = 0  success\_set\_meta\_count = 0  total\_delete\_count = 0  success\_delete\_count = 0  total\_download\_count = 0  success\_download\_count = 0  total\_get\_meta\_count = 0  success\_get\_meta\_count = 0  total\_create\_link\_count = 0  success\_create\_link\_count = 0  total\_delete\_link\_count = 0  success\_delete\_link\_count = 0  total\_upload\_bytes = 24  success\_upload\_bytes = 24  total\_append\_bytes = 0  success\_append\_bytes = 0  total\_modify\_bytes = 0  success\_modify\_bytes = 0  stotal\_download\_bytes = 0  success\_download\_bytes = 0  total\_sync\_in\_bytes = 0  success\_sync\_in\_bytes = 0  total\_sync\_out\_bytes = 24  success\_sync\_out\_bytes = 24  total\_file\_open\_count = 1  success\_file\_open\_count = 1  total\_file\_read\_count = 0  success\_file\_read\_count = 0  total\_file\_write\_count = 1  success\_file\_write\_count = 1  last\_heart\_beat\_time = 2014-03-26 15:00:47  last\_source\_update = 2014-03-26 13:13:55  last\_sync\_update = 1970-01-01 08:00:00  last\_synced\_timestamp = 1970-01-01 08:00:00  Storage 2:  id = 172.16.1.204  ip\_addr = 172.16.1.204 ACTIVE  http domain =  version = 5.01  join time = 2014-03-26 18:53:19  up time = 2014-03-26 18:53:19  total storage = 27789 MB  free storage = 23825 MB  upload priority = 10  store\_path\_count = 1  subdir\_count\_per\_path = 256  storage\_port = 23000  storage\_http\_port = 8080  current\_write\_path = 0  source storage id= 172.16.1.203  if\_trunk\_server= 0  total\_upload\_count = 0  success\_upload\_count = 0  total\_append\_count = 0  success\_append\_count = 0  total\_modify\_count = 0  success\_modify\_count = 0  total\_truncate\_count = 0  success\_truncate\_count = 0  total\_set\_meta\_count = 0  success\_set\_meta\_count = 0  total\_delete\_count = 0  success\_delete\_count = 0  total\_download\_count = 0  success\_download\_count = 0  total\_get\_meta\_count = 0  success\_get\_meta\_count = 0  total\_create\_link\_count = 0  success\_create\_link\_count = 0  total\_delete\_link\_count = 0  success\_delete\_link\_count = 0  total\_upload\_bytes = 0  success\_upload\_bytes = 0  total\_append\_bytes = 0  success\_append\_bytes = 0  total\_modify\_bytes = 0  success\_modify\_bytes = 0  stotal\_download\_bytes = 0  success\_download\_bytes = 0  total\_sync\_in\_bytes = 24  success\_sync\_in\_bytes = 24  total\_sync\_out\_bytes = 0  success\_sync\_out\_bytes = 0  total\_file\_open\_count = 1  success\_file\_open\_count = 1  total\_file\_read\_count = 0  success\_file\_read\_count = 0  total\_file\_write\_count = 1  success\_file\_write\_count = 1  last\_heart\_beat\_time = 2014-03-26 15:00:40  last\_source\_update = 1970-01-01 08:00:00  last\_sync\_update = 2014-03-27 03:38:23  last\_synced\_timestamp = 2014-03-26 13:13:56 (-1s delay)  Group 2:  group name = group2  disk total space = 27789 MB  disk free space = 23825 MB  trunk free space = 0 MB  storage server count = 2  active server count = 2  storage server port = 23000  storage HTTP port = 8080  store path count = 1  subdir count per path = 256  current write server index = 0  current trunk file id = 0  Storage 1:  id = 172.16.1.205  ip\_addr = 172.16.1.205 ACTIVE  http domain =  version = 5.01  join time = 2014-03-26 19:37:36  up time = 2014-03-26 19:37:36  total storage = 27789 MB  free storage = 23825 MB  upload priority = 10  store\_path\_count = 1  subdir\_count\_per\_path = 256  storage\_port = 23000  storage\_http\_port = 8080  current\_write\_path = 0  source storage id=  if\_trunk\_server= 0  total\_upload\_count = 0  success\_upload\_count = 0  total\_append\_count = 0  success\_append\_count = 0  total\_modify\_count = 0  success\_modify\_count = 0  total\_truncate\_count = 0  success\_truncate\_count = 0  total\_set\_meta\_count = 0  success\_set\_meta\_count = 0  total\_delete\_count = 0  success\_delete\_count = 0  total\_download\_count = 0  success\_download\_count = 0  total\_get\_meta\_count = 0  success\_get\_meta\_count = 0  total\_create\_link\_count = 0  success\_create\_link\_count = 0  total\_delete\_link\_count = 0  success\_delete\_link\_count = 0  total\_upload\_bytes = 0  success\_upload\_bytes = 0  total\_append\_bytes = 0  success\_append\_bytes = 0  total\_modify\_bytes = 0  success\_modify\_bytes = 0  stotal\_download\_bytes = 0  success\_download\_bytes = 0  total\_sync\_in\_bytes = 0  success\_sync\_in\_bytes = 0  total\_sync\_out\_bytes = 0  success\_sync\_out\_bytes = 0  total\_file\_open\_count = 0  success\_file\_open\_count = 0  total\_file\_read\_count = 0  success\_file\_read\_count = 0  total\_file\_write\_count = 0  success\_file\_write\_count = 0  last\_heart\_beat\_time = 2014-03-26 15:01:02  last\_source\_update = 1970-01-01 08:00:00  last\_sync\_update = 1970-01-01 08:00:00  last\_synced\_timestamp = 1970-01-01 08:00:00  Storage 2:  id = 172.16.1.206  ip\_addr = 172.16.1.206 ACTIVE  http domain =  version = 5.01  join time = 2014-03-26 22:38:04  up time = 2014-03-26 22:38:04  total storage = 27789 MB  free storage = 23825 MB  upload priority = 10  store\_path\_count = 1  subdir\_count\_per\_path = 256  storage\_port = 23000  storage\_http\_port = 8080  current\_write\_path = 0  source storage id= 172.16.1.205  if\_trunk\_server= 0  total\_upload\_count = 0  success\_upload\_count = 0  total\_append\_count = 0  success\_append\_count = 0  total\_modify\_count = 0  success\_modify\_count = 0  total\_truncate\_count = 0  success\_truncate\_count = 0  total\_set\_meta\_count = 0  success\_set\_meta\_count = 0  total\_delete\_count = 0  success\_delete\_count = 0  total\_download\_count = 0  success\_download\_count = 0  total\_get\_meta\_count = 0  success\_get\_meta\_count = 0  total\_create\_link\_count = 0  success\_create\_link\_count = 0  total\_delete\_link\_count = 0  success\_delete\_link\_count = 0  total\_upload\_bytes = 0  success\_upload\_bytes = 0  total\_append\_bytes = 0  success\_append\_bytes = 0  total\_modify\_bytes = 0  success\_modify\_bytes = 0  stotal\_download\_bytes = 0  success\_download\_bytes = 0  total\_sync\_in\_bytes = 0  success\_sync\_in\_bytes = 0  total\_sync\_out\_bytes = 0  success\_sync\_out\_bytes = 0  total\_file\_open\_count = 0  success\_file\_open\_count = 0  total\_file\_read\_count = 0  success\_file\_read\_count = 0  total\_file\_write\_count = 0  success\_file\_write\_count = 0  last\_heart\_beat\_time = 2014-03-26 15:01:02  last\_source\_update = 1970-01-01 08:00:00  last\_sync\_update = 1970-01-01 08:00:00  last\_synced\_timestamp = 1970-01-01 08:00:00  Group 3:  group name = group3  disk total space = 27789 MB  disk free space = 23825 MB  trunk free space = 0 MB  storage server count = 2  active server count = 2  storage server port = 23000  storage HTTP port = 8080  store path count = 1  subdir count per path = 256  current write server index = 1  current trunk file id = 0  Storage 1:  id = 172.16.1.207  ip\_addr = 172.16.1.207 ACTIVE  http domain =  version = 5.01  join time = 2014-03-26 22:50:28  up time = 2014-03-26 22:50:28  total storage = 27789 MB  free storage = 23825 MB  upload priority = 10  store\_path\_count = 1  subdir\_count\_per\_path = 256  storage\_port = 23000  storage\_http\_port = 8080  current\_write\_path = 0  source storage id= 172.16.1.208  if\_trunk\_server= 0  total\_upload\_count = 1  success\_upload\_count = 1  total\_append\_count = 0  success\_append\_count = 0  total\_modify\_count = 0  success\_modify\_count = 0  total\_truncate\_count = 0  success\_truncate\_count = 0  total\_set\_meta\_count = 0  success\_set\_meta\_count = 0  total\_delete\_count = 0  success\_delete\_count = 0  total\_download\_count = 0  success\_download\_count = 0  total\_get\_meta\_count = 0  success\_get\_meta\_count = 0  total\_create\_link\_count = 0  success\_create\_link\_count = 0  total\_delete\_link\_count = 0  success\_delete\_link\_count = 0  total\_upload\_bytes = 35374  success\_upload\_bytes = 35374  total\_append\_bytes = 0  success\_append\_bytes = 0  total\_modify\_bytes = 0  success\_modify\_bytes = 0  stotal\_download\_bytes = 0  success\_download\_bytes = 0  total\_sync\_in\_bytes = 0  success\_sync\_in\_bytes = 0  total\_sync\_out\_bytes = 0  success\_sync\_out\_bytes = 0  total\_file\_open\_count = 1  success\_file\_open\_count = 1  total\_file\_read\_count = 0  success\_file\_read\_count = 0  total\_file\_write\_count = 1  success\_file\_write\_count = 1  last\_heart\_beat\_time = 2014-03-26 15:01:20  last\_source\_update = 2014-03-27 03:20:13  last\_sync\_update = 1970-01-01 08:00:00  last\_synced\_timestamp = 1970-01-01 08:00:00  Storage 2:  id = 172.16.1.208  ip\_addr = 172.16.1.208 ACTIVE  http domain =  version = 5.01  join time = 2014-03-26 22:49:37  up time = 2014-03-26 22:49:37  total storage = 27789 MB  free storage = 23825 MB  upload priority = 10  store\_path\_count = 1  subdir\_count\_per\_path = 256  storage\_port = 23000  storage\_http\_port = 8080  current\_write\_path = 0  source storage id=  if\_trunk\_server= 0  total\_upload\_count = 0  success\_upload\_count = 0  total\_append\_count = 0  success\_append\_count = 0  total\_modify\_count = 0  success\_modify\_count = 0  total\_truncate\_count = 0  success\_truncate\_count = 0  total\_set\_meta\_count = 0  success\_set\_meta\_count = 0  total\_delete\_count = 0  success\_delete\_count = 0  total\_download\_count = 0  success\_download\_count = 0  total\_get\_meta\_count = 0  success\_get\_meta\_count = 0  total\_create\_link\_count = 0  success\_create\_link\_count = 0  total\_delete\_link\_count = 0  success\_delete\_link\_count = 0  total\_upload\_bytes = 0  success\_upload\_bytes = 0  total\_append\_bytes = 0  success\_append\_bytes = 0  total\_modify\_bytes = 0  success\_modify\_bytes = 0  stotal\_download\_bytes = 0  success\_download\_bytes = 0  total\_sync\_in\_bytes = 35374  success\_sync\_in\_bytes = 35374  total\_sync\_out\_bytes = 0  success\_sync\_out\_bytes = 0  total\_file\_open\_count = 1  success\_file\_open\_count = 1  total\_file\_read\_count = 0  success\_file\_read\_count = 0  total\_file\_write\_count = 1  success\_file\_write\_count = 1  last\_heart\_beat\_time = 2014-03-26 15:01:32  last\_source\_update = 1970-01-01 08:00:00  last\_sync\_update = 2014-03-27 03:20:19  last\_synced\_timestamp = 2014-03-27 03:20:14 (-1s delay) |

### 其他功能

* 使用fdfs\_crc32获取文件CRC
* 使用fdfs\_delete\_file删除文件
* 使用fdfs\_file\_info查看文件属性信息

以上所有功能均可使用接口在代码中调用。

## 附录

