

FastDFS的搭建----云服务器搭建

原创

一个坏小孩  于 2019-05-28 22:04:49 发布  534  收藏 4

分类专栏: [服务器环境搭建](#) 文章标签: [FastDFS的云服务器版搭建](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: https://blog.csdn.net/qq_37215474/article/details/90648133

版权



[服务器环境搭建](#) 专栏收录该内容

1 篇文章 0 订阅

订阅专栏

此教程应该是最简单的教程了你只需要直接按照下面的命令执行就好。

1. 安装 Linux(演示用使用 CentOS7) 系统 (请自行百度)

2. 使用 SecureCRT 工具 连接到的你的 Linux (请自行百度)

3. 关闭 CentOS7 的防火墙 (或者将相应的端口放行)

我采用的是关闭防护墙(也没人看得上攻击你的服务器)

```
systemctl stop firewalld.service          #停止 firewall
systemctl disable firewalld.service      #禁止 firewall 开机启动
firewall --cmd --state                   #查看默认防火墙状态(关闭后显示notrunning, 开启后显示 running)
```

4. FastDFS的环境 准备 (直接执行下面命令)

```
[root@localhost ~]# yum install make cmake gcc gcc-c++ c++
[root@localhost ~]# yum install vim --enhanced
[root@localhost ~]# yum install zip unzip
[root@localhost ~]# yum install --y pcre pcre --devel
[root@localhost ~]# yum install --y zlib zlib --devel
[root@localhost ~]# yum install --y openssl openssl --devel
[root@localhost ~]# yum install net --tools
[root@localhost ~]# yum install git
```

5. 下载所需要的安装包 (GitHub直接下载就好)

```
[root@localhost ~]# git clone https://github.com/TyCoding/build --fastdfs.git
[root@localhost ~]# mv build --fastdfs/*/root/ && rm --rf build --fastdfs
[root@localhost ~]# cd software
```

6. 安装 libfastcommon

将 libfastcommon-master.zip 解压到 /usr/local/fast 目录下, 并编译安装

```
[root@localhost software]# unzip libfastcommon --master.zip -d /usr/local/fast/
[root@localhost software]# cd /usr/local/fast/libfastcommon --master/
[root@localhost libfastcommon --master]# ./make.sh && ./make.sh install
```

由于 libfastcommon 安装好后会自动将库文件拷贝至 /usr/lib64 目录下，但因为 FastDFS 程序引用 /usr/lib 目录，所以我们可以通
过创建软连接的方式来解决 FastDFS 的引用路径问题，当然通过将 /usr/lib64 下的库文件拷贝到 /usr/lib 下也行。

```
[root@localhost libfastcommon- - master] # ln --s /usr/lib64/libfastcommon.so /usr/local/lib/libfastcommon
.so
[root@localhost libfastcommon- - master]# ln --s /usr/lib64/libfastcommon.so /usr/lib/libfastcommon.so
[root@localhost libfastcommon- - master] # ln --s /usr/lib64/libfdfsclient.so /usr/local/lib/libfdfsclient.
so
[root@localhost libfastcommon- - master] # ln --s /usr/lib64/libfdfsclient.so /usr/lib/libfdfsclient.so
```

7. 安装 FastDFS

将 FastDFS_v5.05.tar.gz 解压到 /usr/local/fast 目录下

```
[root@localhost libfastcommon-master]# cd ~/software
[root@localhost software]# tar -zxvf FastDFS_v5.05.tar.gz -C /usr/local/fast/
[root@localhost software]# cd /usr/local/fast/FastDFS/
[root@localhost FastDFS]# ./make.sh && ./make.sh install
```

因为 FastDFS 服务器脚本设置的 bin 目录为 /usr/local/bin 目录下，但是实际我们安装在 /usr/bin 下面，所以我们要修改 FastDFS 配置
文件中的路径，需要修改配置文件：/etc/init.d/fdfs_storaged 以及 /etc/init.d/fdfs_trackerd，修改方式如下：

```
[root@localhost FastDFS]# vim /etc/init.d/fdfs_storaged --这里涉及到了vim的语法，进入这个文件的编辑页面后，直接将输入
法切换至英文状态，输入
:%s+/usr/local/bin+/usr/bin --当输入完成上面的指令后直接回车会看到：7 substitutions on 7 lines，输入
:wq! --保存并退出
```

```
restart() {
    /usr/local/bin/restart.sh $CMD &
    :%s+/usr/local/bin+/usr/bin
```

一定要注意的，这里并不是在文件中增加这一行代码，如上图，我们是给 vim 输入了一串指令，这个指令的作用是批量
修改一些代码，当回车后会显示：7 substitutions on 7 lines，然后我们输入:wq!保存并退出文件的编辑。

同样的方式，修改 /etc/init.d/fdfs_trackerd，输入命令

```
[root@localhost FastDFS]# vim /etc/init.d/fdfs_trackerd
:%s+/usr/local/bin+/usr/bin
:wq!
```

8. 配置 tracker

进入/etc/fdfs目录并复制一份tracker.conf.sample并命名为tracker.conf，并修改tracker.conf，将其中的base_path参数的值修改为/fastdfs/tracker，并使用mkdir -p /fastdfs/tracker创建两个目录

点击键盘的i键进入vim的编辑模式，如果编辑完成后要先点击键盘的ESC键退出vim的编辑模式再点击键盘的键:wq!直接会在左下角写入

```
# network timeout in seconds
# default value is 30s
network_timeout=60

# the base path to store data and log files
base_path=/fastdfs/tracker

# max concurrent connections this server supported
max_connections=256
```

https://blog.csdn.net/qq_37215474

上面的文件需要我们自己创建所以在接下来在终端中输入如下命令

```
:wq!
[root@localhost fdfs]# mkdir -p /fastdfs/tracker
```

9. 启动tracker

```
[root@localhost fdfs]# /etc/init.d/fdfs_trackerd start
```

10. 配置storage

进入/etc/fdfs目录，赋值一份storage.conf.sample并命名为storage.conf，并修改storage.conf

```
[root@localhost fdfs]# cp storage.conf.sample storage.conf
[root@localhost fdfs]# vi storage.conf
```

修改如下几个参数：

```
base_path=/fastdfs/storage
store_path0=/fastdfs/storage
tracker_server=192.168.148.131:22122 #这个IP地址是你服务器的IP,如果是云服务器则是你的公网IP,不要盲目复制。
```

保存退出并创建/fastdfs/storage目录

```
:wq!
[root@localhost fdfs]# mkdir /fastdfs/storage
```

启动storage

```
/etc/init.d/fdfs_storaged start #直接在你的终端上输入
```

查看tracker和storage是否启动成功

```
ps -ef | grep fdfs
```

```
[root@localhost ~]# ps -ef | grep fdfs
root      1221      1  0 07:47 ?        00:00:00 /usr/bin/fdfs_trackerd /etc/fdfs/tracker.conf
root      1245      1 99 07:47 ?        00:00:01 /usr/bin/fdfs_storaged /etc/fdfs/storage.conf
root      1247     1185  0 07:47 pts/0    00:00:00 grep --color=auto fdfs
[root@localhost ~]#
```

出现如上的显示才证明tracker和storage都正常启动了。注意 这个地方要吐槽一下啦，要是你输入ps -ef | grep fdfs出现的不是这三行，而是两行或其他，就证明肯定有一个服务没有启动成功，一般就是你的配置有错误；那么你有下面两个选择来解决这一错误情况：

1. 执行/etc/init.d/fdfs_storaged stop和/etc/init.d/fdfs_trackerd stop关闭服务，然后先启动tracker服务再执行storage服务，一般就可以解决
2. 如果上面的办法还是不行，就重新安装FastDFS（从新直接编译方法即可）；其次还要重新删除tracker.conf和storage.conf重新配置；最后再重新启动服务（尽量重启一下服务器）

11. 测试文件上传

进入/etc/fdfs目录并复制一份client.conf.sample并命名为client.conf

```
[root@localhost fdfs]# cp client.conf.sample client.conf
[root@localhost fdfs]# vi client.conf
```

编辑下列两项内容

```
base_path=/fastdfs/tracker
tracker_server=192.168.148.131:22122 #这个IP地址是你服务器的IP地址，云服务器的公网IP，不要盲目复制
```

```
# network timeout in seconds
# default value is 30s
network_timeout=60

# the base path to store log files
base_path=/fastdfs/tracker

# tracker_server can occur more than once, and tracker_server format is
# "host:port", host can be hostname or ip address
tracker_server=192.168.148.131:22122
```

https://blog.csdn.net/qq_37215474

下面使用fdfs_upload_file脚本进行文件上传操作。先创建测试文件，输入命令vim test.txt回车后点击i键进入编辑模式，输入Hello Coding!，点击ESC键，输入:wq!保存并退出。

```
[root@localhost ~]# vim test.txt
Hello WangKai FastDFS is succeeded!
/usr/bin/fdfs_upload_file /etc/fdfs/client.conf ~/test.txt #直接在终端输入
```

如果出现类似下面的返回地址证明上传成功

```
"test.txt" [New] 1L, 16C written
[root@localhost ~]# /usr/bin/fdfs upload file /etc/fdfs/client.conf test.txt
group1/M00/00/00/wKiUg1uFPWaACPbMAAAAEADPLa8183.txt
[root@localhost ~]#
```

13. FastDFS和Nginx结合（因为通过浏览器访问要用）

解压nginx-1.6.2.tar.gz到/usr/local目录下；解压fastdfs-nginx-module_v1.16.tar.gz到/usr/local目录下，编译和安装

```
[root@localhost ~]# cd ~/software
[root@localhost software]# tar -zxvf nginx-1.6.2.tar.gz -C /usr/local/
[root@localhost software]# tar -zxvf fastdfs-nginx-module_v1.16.tar.gz -C /usr/local/fast/
```

进入/usr/local/fast/fastdfs-nginx-module/src目录下，修改其中的config文件，把其中第四行的usr/local/include都改为/usr/include

```
[root@localhost software]# cd /usr/local/fast/fastdfs-nginx-module/src
[root@localhost src]# vi config
ngx_addon_name=ngx_http_fastdfs_module
HTTP_MODULES="$HTTP_MODULES ngx_http_fastdfs_module"
NGX_ADDON_SRCS="$NGX_ADDON_SRCS $ngx_addon_dir/ngx_http_fastdfs_module.c" 去掉这行中的两个local
CORE_INCS="$CORE_INCS /usr/include/fastdfs /usr/include/fastcommon/"
CORE_LIBS="$CORE_LIBS -L/usr/local/lib -lfastcommon -lfdfsclient"
CFLAGS="$CFLAGS -D_FILE_OFFSET_BITS=64 -DFDFS_OUTPUT_CHUNK_SIZE='256*1024' -DFDFS_MOD_CONF_FILENAME='\"/etc/fdfs/mod_fastdfs.conf\"'"
```

14.编译Nginx

```
[root@localhost src]# cd /usr/local/ && ll
[root@localhost local]# cd nginx-1.6.2 && ./configure --prefix=/usr/local/nginx --add-module=../fast/fastdfs-nginx-module/src/
```

正常情况下，会显示如下信息证明编译成功，因为这里需要为nginx添加fastdfs-nginx-module模块，和平常安装的nginx是不同的，如果不显示如下页面就重新进行编译

```
Configuration summary
+ using system PCRE library
+ OpenSSL library is not used
+ md5: using system crypto library
+ sha1: using system crypto library
+ using system zlib library

nginx path prefix: "/usr/local/nginx"
nginx binary file: "/usr/local/nginx/sbin/nginx"
nginx configuration prefix: "/usr/local/nginx/conf"
nginx configuration file: "/usr/local/nginx/conf/nginx.conf"
nginx pid file: "/usr/local/nginx/logs/nginx.pid"
nginx error log file: "/usr/local/nginx/logs/error.log"
nginx http access log file: "/usr/local/nginx/logs/access.log"
nginx http client request body temporary files: "client_body_temp"
nginx http proxy temporary files: "proxy_temp"
nginx http fastcgi temporary files: "fastcgi_temp"
nginx http uwsgi temporary files: "uwsgi_temp"
nginx http scgi temporary files: "scgi_temp"

[root@localhost nginx-1.6.2]#
```

https://blog.csdn.net/qq_37215474

安装，并copy /usr/local/fast/fastdfs-nginx-module/src/mod_fastdfs.conf 到/etc/fdfs

```
[root@localhost nginx-1.6.2]# make && make install
[root@localhost nginx-1.6.2]# cd /usr/local/fast/fastdfs-nginx-module/src/
[root@localhost src]# cp /usr/local/fast/fastdfs-nginx module/src/mod_fastdfs.conf /etc/fdfs/
[root@localhost src]# cd /etc/fdfs && ll
[root@localhost fdfs]# vim mod_fastdfs.conf
```

修改如下部分

```
connect_timeout=10
tracker_server=192.168.148.131:22122 #IP地址还是根据自己的服务器IP地址改
url_have_group_name = true
store_path0=/fastdfs/storage
```

赋值FastDFS里的2个文件，到/etc/fdfs目录中，创建一个软连接，在/fastdfs/storage文件储存目录下创建软连接，将其连接到实际存放数据的目录

```
[root@localhost fdfs]# cd /usr/local/fast/FastDFS/conf/ && ll
[root@localhost conf]# cp http.conf mime.types /etc/fdfs/
[root@localhost conf]# ln -s /fastdfs/storage/data/ /fastdfs/storage/data/M00
```

修改nginx的配置文件nginx-conf

```
[root@localhost conf]# cd /usr/local/nginx/conf && ll
[root@localhost conf]# vim nginx.conf
```

修改如下部分

```
location ~/group([0-9])/M00 {  
    ngx_fastdfs_module;  
}
```

```
server {  
    listen      80;  
    server_name localhost;  
  
    #charset koi8-r;  
  
    #access_log logs/host.access.log main;  
  
    location ~/group([0-9])/M00 {  
        # root    html;  
        # index  index.html index.htm;  
        ngx_fastdfs_module;  
    }  
  
    #error_page 404                /404.html;  
  
    # redirect server error pages to the static page /50x.html  
    #  
    error_page 500 502 503 504 /50x.html;  
    location = /50x.html {  
        root    html;  
    }  
  
"nginx.conf" 118L, 2706C written  
[root@localhost conf]#
```

https://blog.csdn.net/qq_37215474

启动nginx

```
[root@localhost conf]# /usr/local/nginx/sbin/nginx
```

```
"nginx.conf" 118L, 2706C written
[root@localhost conf]# /usr/local/nginx/sbin/nginx
ngx_http_fastdfs_set pid=5648
[root@localhost conf]#
```

如果出现以上显示证明nginx启动成功，否者的话启动失败，要重新编译安装nginx，一般都是fastdfs-nginx-module模块注入到nginx失败。

以上我们已经完成了全部的配置，测试各个端口的启动情况，输入netstat -ntlp，可以看到如下信息

```
[root@localhost conf]# netstat -ntlp
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:22122           0.0.0.0:*               LISTEN      1221/fdfs_trackerd
tcp        0      0 0.0.0.0:80              0.0.0.0:*               LISTEN      5649/nginx: master
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      910/sshd
tcp        0      0 0.0.0.0:23000           0.0.0.0:*               LISTEN      1249/fdfs_storaged
tcp        0      0 127.0.0.1:25            0.0.0.0:*               LISTEN      1065/master
tcp6       0      0 :::22                   :::*                    LISTEN      910/sshd
tcp6       0      0 :::1:25                 :::*                    LISTEN      1065/master
[root@localhost conf]#
```

https://blog.csdn.net/qq_37215474

15. 通过http上传访问文件

```
[root@localhost conf]# /usr/bin/fdfs_upload_file /etc/fdfs/client.conf ~/test.txt
```

```
[root@localhost conf]# /usr/bin/fdfs_upload_file /etc/fdfs/client.conf ~/test.txt
group1/M00/00/00/wKiUg1uFSb6AULecAAAAEADPLa8389.txt
[root@localhost conf]#
```



https://blog.csdn.net/qq_37215474

16. 设置开机自启动

```
[root@localhost ~]# vi /etc/rc.d/rc.local
```

向其中添加需要开机自启动服务的脚本

```
/etc/init.d/fdfs_trackerd start
/etc/init.d/fdfs_storaged start
/usr/local/nginx/sbin/nginx
```

```
[root@localhost ~]# vi /etc/rc.d/rc.local
#!/bin/bash
# THIS FILE IS ADDED FOR COMPATIBILITY PURPOSES
#
# It is highly advisable to create own systemd services or udev rules
# to run scripts during boot instead of using this file.
#
# In contrast to previous versions due to parallel execution during boot
# this script will NOT be run after all other services.
#
# Please note that you must run 'chmod +x /etc/rc.d/rc.local' to ensure
# that this script will be executed during boot.

touch /var/lock/subsys/local

/etc/init.d/fdfs_trackerd start
/etc/init.d/fdfs_storaged start
/usr/local/nginx/sbin/nginx
~
```

https://blog.csdn.net/qq_37215474

其次，我们还要给rc.local文件赋予权限才能实现开机自启动

```
[root@localhost ~]# chmod +x /etc/rc.d/rc.local

~/
"/etc/rc.d/rc.local" 17L, 566C written
[root@localhost ~]# chmod +x /etc/rc.d/rc.local
[root@localhost ~]# █
```

输入命令reboot重启服务器，再输入netstat -ntlp查看当前启动的服务，可以看到storage, tracker, nginx三个服务都自启动了。

```
[root@localhost ~]# reboot

Last login: Tue Aug 28 09:30:55 2018 from 192.168.148.1
[root@localhost ~]# netstat -ntlp
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:22122           0.0.0.0:*               LISTEN     938/fdfs_trackerd
tcp        0      0 0.0.0.0:80             0.0.0.0:*               LISTEN     961/nginx: master p
tcp        0      0 0.0.0.0:22             0.0.0.0:*               LISTEN     905/sshd
tcp        0      0 0.0.0.0:23000          0.0.0.0:*               LISTEN     960/fdfs_storaged
tcp        0      0 127.0.0.1:25           0.0.0.0:*               LISTEN     1096/master
tcp6       0      0 :::22                  :::*                    LISTEN     905/sshd
tcp6       0      0 :::1:25                :::*                    LISTEN     1096/master
[root@localhost ~]# █
```

https://blog.csdn.net/qq_37215474

到这里FastDFS就全部完成。