FastDFS分布式文件系统集群安装与配置

目录

[FastDFS分布式文件系统集群安装与配置 1](#_Toc495411707)

[**一、** **FastDFS集群规划** 3](#_Toc495411708)

[**二、** **FastDFS集群架构图** 3](#_Toc495411709)

[**三、** **安装集群节点** 4](#_Toc495411710)

[A. 安装所需的依赖包 4](#_Toc495411711)

[B. 安装libfatscommon 4](#_Toc495411712)

[C. 安装FastDFS 4](#_Toc495411713)

[**四、** **配置跟踪节点(192.168.1.200 192.168.1.201)** 5](#_Toc495411714)

[说明：每个节点执行相同的操作 5](#_Toc495411715)

[A. 复制tracker样例配置文件并重命名 5](#_Toc495411716)

[B. 修改trackerconf配置文件 5](#_Toc495411717)

[C. 创建base\_path指定的目录 5](#_Toc495411718)

[D. 防火墙中打开tracker服务器端口 默认为 22122 5](#_Toc495411719)

[E. 启动tracker服务器 5](#_Toc495411720)

[F. 停止tracker服务器 6](#_Toc495411721)

[G. 设置tracker服务开机启动 6](#_Toc495411722)

[**五、** **配置存储节点** 6](#_Toc495411723)

[A. 复制storage样例配置文件并重命名 6](#_Toc495411724)

[B. 编辑配置文件 6](#_Toc495411725)

[C. 创建基础数据目录 6](#_Toc495411726)

[D. 防火墙中打开storage服务器端口 默认为 23000 6](#_Toc495411727)

[E. 启动storage服务器 6](#_Toc495411728)

[F. 停止storage服务器 7](#_Toc495411729)

[G. 设置storage服务开机启动 7](#_Toc495411730)

[**六、** **文件上传测试** 7](#_Toc495411731)

[A. 修改tracker服务器clientconf配置文件 7](#_Toc495411732)

[B. 执行文件上传命令 7](#_Toc495411733)

[**七、** **储节点安装Nginx和fastdfs-nginx-module模块** 7](#_Toc495411734)

[A. fastdfs-nginx-module作用说明 7](#_Toc495411735)

[B. 安装nginx和fastdfs-nginx-module模块 8](#_Toc495411736)

[C. 复制 fastdfs-nginx-module 源码中的配置文件到etcfdfs 目录并修改 8](#_Toc495411737)

[D. 复制FastDFS源文件目录中HTTP相关的配置文件到etcfdfs目录 9](#_Toc495411738)

[E. 创建数据存放目录的软链接 9](#_Toc495411739)

[F. 配置fastdfs-nginx-module（inx简洁版样例） 9](#_Toc495411740)

[G. 防火墙中打开Nginx的 8888 端口 10](#_Toc495411741)

[H. 启动Nginx 10](#_Toc495411742)

[I. 通过浏览器访问测试时上传的文件 10](#_Toc495411743)

[**八、** **跟踪节点安装Nginx和ngx\_cache\_purge模块** 11](#_Toc495411744)

[A. 安装Nginx所需的依赖包 11](#_Toc495411745)

[B. 安装nginx和ngx\_cache\_purge模块 11](#_Toc495411746)

[C. 配置Nginx设置tracker负载均衡以及缓存 11](#_Toc495411747)

[D. 防火墙打开Nginx 8000 端口 13](#_Toc495411748)

[E. 启动Nginx 14](#_Toc495411749)

[F. 文件访问测试 14](#_Toc495411750)

[**九、** **配置Tracker服务器高可用反向代理与负载均衡** 14](#_Toc495411751)

[A. 安装keepalived与Nginx 14](#_Toc495411752)

[B. 配置Keeyalived Nginx高可用 14](#_Toc495411753)

[C. 配置nginx对tracker节点的负载均衡 15](#_Toc495411754)

[D. 重启1921681206 和 1921681207 中的Nginx 16](#_Toc495411755)

[E. 通过虚拟IP访问文件测试 16](#_Toc495411756)

1. **FastDFS集群规划**

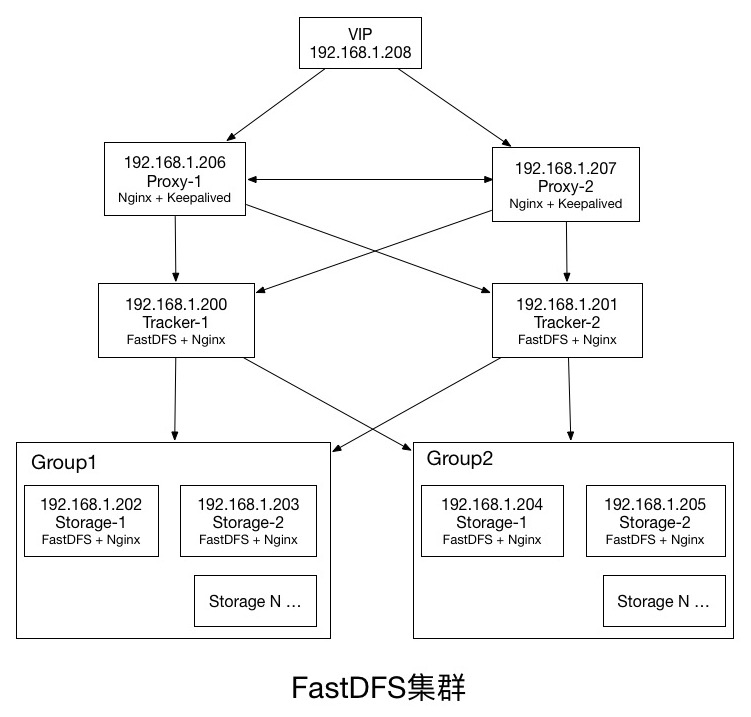
跟踪服务器负载均衡节点1：192.168.1.206 dfs-nginx-proxy-1   
跟踪服务器负载均衡节点2：192.168.1.207 dfs-nginx-proxy-2   
跟踪服务器1：192.168.1.200 dfs-tracker-1   
跟踪服务器2：192.168.1.201 dfs-tracker-2   
存储服务器1：192.168.1.202 dfs-storage-group1-1   
存储服务器2：192.168.1.203 dfs-storage-group1-2   
存储服务器3：192.168.1.204 dfs-storage-group2-1   
存储服务器3：192.168.1.205 dfs-storage-group2-2   
HA虚拟IP：192.168.1.208   
HA软件：Keepalived   
操作系统：CentOS 7   
用户：root   
数据目录：/fastdfs

**安装包：**   
fastdfs-master-V5.05.zip：FastDFS源码   
libfastcommon-master.zip：（从 FastDFS 和 FastDHT 中提取出来的公共 C 函数库）   
fastdfs-nginx-module-master.zip：storage节点http服务nginx模块   
nginx-1.10.0.tar.gz：Nginx安装包   
ngx\_cache\_purge-2.3.tar.gz：Nginx图片缓存清除模块   
**获取安装包的方式：**   
1> 从这里下载打包好的所有安装包：<http://download.csdn.net/detail/xyang81/9667493>   
2> 从作者github官网挨个下载fastdfs源码及其依赖库：<https://github.com/happyfish100> 和 Nginx缓存清除模块：<https://github.com/FRiCKLE/ngx_cache_purge>

开始前，先将所有安装包下载到各个节点的/usr/local/src目录中。

1> 本文称节点IP最后一段就代表某个节点，如：192.168.1.206，文中提到206节点，就代表192.168.1.206。   
2> 本文称tracker或跟踪服务器是同一个意思   
3> 本文称storage或存储服务器是同一个意思

1. **FastDFS集群架构图**



1. **安装集群节点**
   1. 安装所需的依赖包
2. shell**>** yum install make cmake gcc gcc-c++
   1. 安装libfatscommon
      * 1. shell**>** cd **/**usr**/**local**/**src
        2. shell**>** unzip libfastcommon-master.zip
        3. shell**>** cd libfastcommon-master
        4. ## 编译、安装
        5. shell**>** **./**make.sh
        6. shell**>** **./**make.sh install
   2. 安装FastDFS
3. shell**>** cd **/**usr**/**local**/**src
4. shell**>** unzip fastdfs-master-V5.05.zip
5. shell**>** cd fastdfs-master
6. ## 编译、安装
7. shell**>** **./**make.sh
8. shell**>** **./**make.sh install
9. [**配置跟踪节点(192.168.1.200 192.168.1.201**](http://blog.csdn.net/xyang81/article/details/52928230#t6)**)**

说明：每个节点执行相同的操作

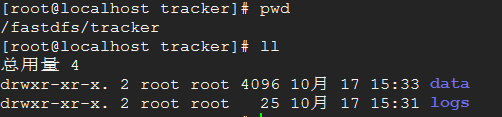
* 1. 复制tracker样例配置文件并重命名

1. shell**>** cp **/**etc**/**fdfs**/**tracker.conf.sample **/**etc**/**fdfs**/**tracker.conf
   1. 修改trackerconf配置文件
2. shell**>** vim **/**etc**/**fdfs**/**tracker.conf
3. # 修改的内容如下：
4. disabled**=**false # 启用配置文件
5. port**=**22122 # tracker服务器端口（默认22122）
6. base\_path**=/**fastdfs**/**tracker # 存储日志和数据的根目录
   1. 创建base\_path指定的目录
7. shell**>** mkdir -p **/**fastdfs**/**tracker
   1. 防火墙中打开tracker服务器端口 默认为 22122
8. shell**>** vi **/**etc**/**sysconfig**/**iptables

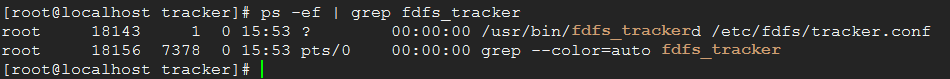
添加如下端口行：   
-A INPUT -m state --state NEW -m tcp -p tcp --dport 22122 -j ACCEPT   
重启防火墙：

1. shell**>** service iptables restart
   1. 启动tracker服务器
2. shell> /etc/init.d/fdfs\_trackerd start

初次启动，会在/fastdfs/tracker目录下生成logs、data两个目录：



检查FastDFS Tracker Server是否启动成功：**ps** **-**ef **|** **grep** fdfs\_trackerd



* 1. 停止tracker服务器

1. shell**>** **/**etc**/**init.d**/**fdfs\_trackerd stop
   1. 设置tracker服务开机启动
2. shell**>** chkconfig fdfs\_trakcerd on
3. **配置存储节点**

**group1：** 192.168.1.202，192.168.1.203

**group2：** 192.168.1.204，192.168.1.205

说明：每个节点执行相同的操作

* 1. 复制storage样例配置文件并重命名

1. shell**>** cp **/**etc**/**fdfs**/**storage.conf.sample **/**etc**/**fdfs**/**storage.conf
   1. 编辑配置文件
2. shell**>** vi **/**etc**/**fdfs**/**storage.conf
3. # 修改的内容如下:
4. disabled**=**false # 启用配置文件
5. port**=**23000 # storage服务端口
6. group\_name**=**group1 # 组名（第一组为group1，第二组为group2，依次类推...）
7. base\_path**=/**fastdfs**/**storage # 数据和日志文件存储根目录
8. store\_path0**=/**fastdfs**/**storage # 第一个存储目录，第二个存储目录起名为：store\_path1=xxx，其它存储目录名依次类推...
9. store\_path\_count**=**1 # 存储路径个数，需要和store\_path个数匹配
10. tracker\_server**=**192**.**168**.**0**.**200**:**22122 # tracker服务器IP和端口
11. tracker\_server**=**192**.**168**.**0**.**201**:**22122 # tracker服务器IP和端口
12. http.server\_port**=**8888 # http访问文件的端口

其它参数保留默认配置

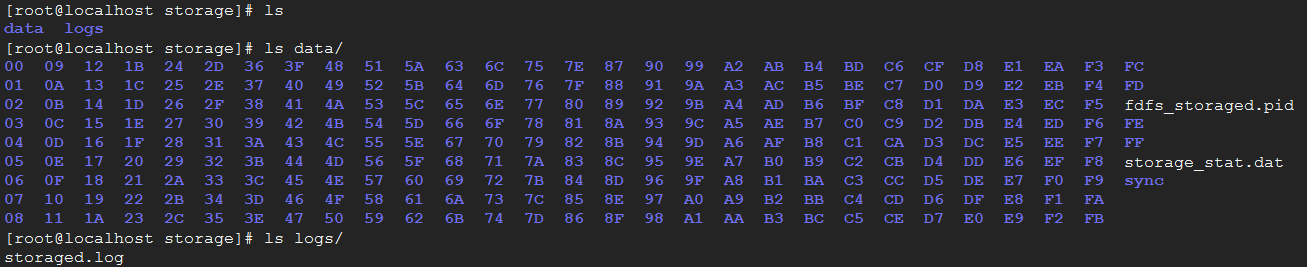
* 1. 创建基础数据目录

1. shell**>** mkdir -p **/**fastdfs**/**storage
   1. 防火墙中打开storage服务器端口 默认为 23000
2. shell**>** vi **/**etc**/**sysconfig**/**iptables

添加如下端口行：   
-A INPUT -m state --state NEW -m tcp -p tcp --dport 22122 -j ACCEPT   
重启防火墙：

1. shell**>** service iptables restart
   1. 启动storage服务器
2. shell**>** **/**etc**/**init.d**/**fdfs\_storaged start

初次启动，会在/fastdfs/storage目录下生成logs、data两个目录



检查FastDFS Tracker Server是否启动成功：**ps** **-**ef **|** **grep** fdfs\_storaged

11111

各节点启动后，使用**tail** -f **/**fastdfs**/**storage**/**logs**/**storaged.log命令监听存储节点的日志，可以看到存储节点链接到跟踪服务器，并提示哪一个为leader跟踪服务器，同时也能看到同一组中其它节点加入进来的日志信息。

所有存储节点都启动之后，可以在任一存储节点上使用如下命令查看集群的状态信息：

1. shell**>** **/**usr**/**bin**/**fdfs\_monitor **/**etc**/**fdfs**/**storage.conf
   1. 停止storage服务器
2. shell**>** **/**etc**/**init.d**/**fdfs\_storaged stop
   1. 设置storage服务开机启动
3. shell**>** chkconfig fdfs\_storaged on
4. **文件上传测试**
   1. 修改tracker服务器clientconf配置文件
5. shell**>** cp **/**etc**/**fdfs**/**client.conf.sample **/**etc**/**fdfs**/**client.conf
6. shell**>** vi **/**etc**/**fdfs**/**client.conf
7. base\_path**=/**fastdfs**/**tracker
8. tracker\_server**=**192**.**168**.**1**.**200**:**22122
9. tracker\_server**=**192**.**168**.**1**.**201**:**22122
   1. 执行文件上传命令
10. shell**>** **/**usr**/**bin**/**fdfs\_upload\_file **/**etc**/**fdfs**/**client.conf **/**usr**/**local**/**src**/**FastDFS\_v5.05.tar.gz

返回以下ID号，说明文件上传成功：

group1/M00/00/00/wKgBh1Xtr9-AeTfWAAVFOL7FJU4.tar.gz

group2/M00/00/00/wKgBiVXtsDmAe3kjAAVFOL7FJU4.tar.gz

（从返回的ID号中也可以看出，同一个文件分别存储在两个组内group1和group2，但也有可能在同一组中，具体策略是由FastDFS根据服务器的存储情况来分配的）

1. **储节点安装Nginx和fastdfs-nginx-module模块**

说明：每个节点执行相同的操作

* 1. fastdfs-nginx-module作用说明

FastDFS 通过 Tracker 服务器，将文件放在 Storage 服务器存储，但是同组存储服务器之间需要进入文件复制流程，有同步延迟的问题。假设 Tracker 服务器将文件上传到了 192.168.1.202，上传成功后文件 ID已经返回给客户端。此时 FastDFS 存储集群机制会将这个文件同步到同组存储 192.168.1.203，在文件还没有复制完成的情况下，客户端如果用这个文件 ID 在 192.168.1.203上取文件，就会出现文件无法访问的错误。而 fastdfs-nginx-module 可以重定向文件连接到源服务器（192.168.1.202）上取文件，避免客户端由于复制延迟导致的文件无法访问错误。

* 1. 安装nginx和fastdfs-nginx-module模块

1. ## 安装nginx所需的依赖包
2. shell**>** yum install gcc gcc-c++ make automake autoconf libtool pcre pcre-devel zlib zlib-devel openssl openssl-devel
3. ## 编译安装nginx（添加fastdfs-nginx-module模块）
4. shell**>** cd **/**usr**/**local**/**src
5. shell**>** tar **-**zxvf nginx-1.10.0.tar.gz
6. shell**>** unzip fastdfs-nginx-module-master.zip
7. shell**>** cd nginx-1.10.0
8. shell**>** **./**configure **--**prefix**=/**opt**/**nginx **--**sbin-path**=/**usr**/**bin**/**nginx **--**add-module**=/**usr**/**local**/**src**/**fastdfs-nginx-module**/**src
9. shell**>** make **&&** **make** install
   1. 复制 fastdfs-nginx-module 源码中的配置文件到etcfdfs 目录并修改
10. shell**>** cp **/**usr**/**local**/**src**/**fastdfs-nginx-module**/**src**/**mod\_fastdfs.conf **/**etc**/**fdfs**/**
11. shell**>** vi **/**etc**/**fdfs**/**mod\_fastdfs.conf
    * 1. 第一组存储服务器的mod\_fastdfsconf配置
12. connect\_timeout**=**10
13. base\_path**=**/tmp
14. tracker\_server**=**192.168.1.200:22122
15. tracker\_server**=**192.168.1.201:22122
16. storage\_server\_port**=**23000
17. group\_name**=**group1 # 第一组storage的组名
18. url\_have\_group\_name**=**true
19. store\_path0**=**/fastdfs/storage
20. group\_count**=**2
21. **[group1]**
22. group\_name**=**group1
23. storage\_server\_port**=**23000
24. store\_path\_count**=**1
25. store\_path0**=**/fastdfs/storage
26. **[group2]**
27. group\_name**=**group2
28. storage\_server\_port**=**23000
29. store\_path\_count**=**1
30. store\_path0**=**/fastdfs/storage
    * 1. 第二组存储服务器的mod\_fastdfsconf配置
31. connect\_timeout**=**10
32. base\_path**=**/tmp
33. tracker\_server**=**192.168.1.200:22122
34. tracker\_server**=**192.168.1.201:22122
35. storage\_server\_port**=**23000
36. group\_name**=**group2 # 第二组storage的组名
37. url\_have\_group\_name**=**true
38. store\_path0**=**/fastdfs/storage
39. group\_count**=**2
40. **[group1]**
41. group\_name**=**group1
42. storage\_server\_port**=**23000
43. store\_path\_count**=**1
44. store\_path0**=**/fastdfs/storage
45. **[group2]**
46. group\_name**=**group2
47. storage\_server\_port**=**23000
48. store\_path\_count**=**1
49. store\_path0**=**/fastdfs/storage
    1. 复制FastDFS源文件目录中HTTP相关的配置文件到etcfdfs目录
50. shell**>** cd **/**usr**/**local**/**src**/**FastDFS**/**conf
51. shell**>** cp http.conf mime.types **/**etc**/**fdfs**/**
    1. 创建数据存放目录的软链接
52. shell**>** ln -s **/**fastdfs**/**storage**/**data**/** **/**fastdfs**/**storage**/**data**/**M00
    1. 配置fastdfs-nginx-module（inx简洁版样例）
53. shell**>** vi **/**opt**/**nginx**/**conf**/**nginx.conf
54. user nobody**;**
55. worker\_processes 1**;**
56. events **{**
57. worker\_cnnections 1024**;**
58. **}**
59. http **{**
60. include mime.types**;**
61. default\_type application**/**octet-stream**;**
62. sendfile on**;**
63. keepalive\_timeout 65**;**
64. server **{**
65. listen 8888**;**
66. server\_name localhost**;**
67. # FastDFS 文件访问配置(fastdfs-nginx-module模块)
68. location **~/**group**([**0**-**9**])/**M00 **{**
69. ngx\_fastdfs\_module**;**
70. **}**
71. error\_page 500 502 503 504 **/**50x.html**;**
72. location **=** **/**50x.html **{**
73. root html**;**
74. **}**
75. **}**
76. **}**

**注意：**

a、8888 端口值要与/etc/fdfs/storage.conf 中的 http.server\_port=8888 相对应，因为 http.server\_port 默认为 8888，如果想改成 80，则要对应修改过来。

b、Storage 对应有多个 group 的情况下，访问路径带 group 名，如：<http://xxxx/group1/M00/00/00/xxx>, 对应的 Nginx 配置为：

1. location **~/**group**([**0**-**9**])/**M00 **{**
2. ngx\_fastdfs\_module**;**
3. **}**

C、如下载时如发现老报 404，将nginx.conf第一行user nobody;修改为user root;后重新启动。

* 1. 防火墙中打开Nginx的 8888 端口

1. shell**>** vi **/**etc**/**sysconfig**/**iptables
2. ## 添加
3. -A INPUT **-**m state **--**state NEW **-**m tcp -p tcp **--**dport 8888 **-**j ACCEPT
4. ## 重启防火墙
5. shell**>** service iptables restart
   1. 启动Nginx
6. shell**>** **/**opt**/**nginx**/**sbin**/**nginx
7. ngx\_http\_fastdfs\_set pid**=**xxx astdfs-nginx-module进程ID

重启 Nginx 的命令为：/usr/local/nginx/sbin/nginx -s reload

设置Nginx开机启动：

1. shell**>** vi **/**etc**/**rc.local
2. # 加入
3. **/**opt**/**nginx**/**sbin**/**nginx
4. shell**>** chmod **+**x **/**etc**/**rc.local # centos7
   1. 通过浏览器访问测试时上传的文件

<http://192.168.1.202:8888/group1/M00/00/00/wKgBh1Xtr9-AeTfWAAVFOL7FJU4.tar.gz>   
<http://192.168.1.204:8888/group2/M00/00/00/wKgBiVXtsDmAe3kjAAVFOL7FJU4.tar.gz>

1. **跟踪节点安装Nginx和ngx\_cache\_purge模块**

说明：每个节点执行相同的操作。

tracker节点：192.168.1.200，192.168.1.201   
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。

* 1. 安装Nginx所需的依赖包

1. shell**>** yum install gcc gcc-c++ make automake autoconf libtool pcre pcre-devel zlib zlib-devel openssl openssl-devel
   1. 安装nginx和ngx\_cache\_purge模块
2. shell**>** cd **/**usr**/**local**/**src
3. shell**>** tar **-**zxvf nginx-1.10.0.tar.gz
4. shell**>** tar **-**zxvf ngx\_cache\_purge-2.3.tar.gz
5. shell**>** cd nginx-1.10.0
6. shell**>** **./**configure **--**prefix**=/**opt**/**nginx **--**sbin-path**=/**usr**/**bin**/**nginx **--**add-module**=/**usr**/**local**/**src**/**ngx\_cache\_purge-2.3
7. shell**>** make **&&** **make** install
   1. 配置Nginx设置tracker负载均衡以及缓存
8. shell**>** vi **/**opt**/**nginx**/**conf**/**nginx.conf
9. user nobody**;**
10. worker\_processes 1**;**
11. events **{**
12. worker\_connections 1024**;**
13. use epoll**;**
14. **}**
15. http **{**
16. include mime.types**;**
17. default\_type application**/**octet-stream**;**
18. #log\_format main '$remote\_addr - $remote\_user [$time\_local] "$request" '
19. # '$status $body\_bytes\_sent "$http\_referer" '
20. # '"$http\_user\_agent" "$http\_x\_forwarded\_for"';
21. #access\_log logs/access.log main;
22. sendfile on**;**
23. tcp\_nopush on**;**
24. keepalive\_timeout 65**;**
25. #gzip on;
26. #设置缓存
27. server\_names\_hash\_bucket\_size 128**;**
28. client\_header\_buffer\_size 32k**;**
29. large\_client\_header\_buffers 4 32k**;**
30. client\_max\_body\_size 300m**;**
31. proxy\_redirect off**;**
32. proxy\_set\_header Host **$http\_host;**
33. proxy\_set\_header X-Real-IP **$remote\_addr;**
34. proxy\_set\_header X-Forwarded-For **$proxy\_add\_x\_forwarded\_for;** proxy\_connect\_timeout 90**;**
35. proxy\_send\_timeout 90**;**
36. proxy\_read\_timeout 90**;**
37. proxy\_buffer\_size 16k**;**
38. proxy\_buffers 4 64k**;**
39. proxy\_busy\_buffers\_size 128k**;**
40. proxy\_temp\_file\_write\_size 128k**;** #设置缓存存储路径、存储方式、分配内存大小、磁盘最大空间、缓存期限
41. proxy\_temp\_path **/**fastdfs**/**cache**/**nginx**/**proxy\_cache**/**tmp**;**
42. #设置 group1 的服务器
43. upstream fdfs\_group1 **{**
44. server 192**.**168**.**1**.**202**:**8888 weight**=**1 max\_fails**=**2 fail\_timeout**=**30s**;**
45. server 192**.**168**.**1**.**203**:**8888 weight**=**1 max\_fails**=**2 fail\_timeout**=**30s**;**
46. **}**
47. #设置 group2 的服务器
48. upstream fdfs\_group2 **{**
49. server 192**.**168**.**1**.**204**:**8888 weight**=**1 max\_fails**=**2 fail\_timeout**=**30s**;**
50. server 192**.**168**.**1**.**205**:**8888 weight**=**1 max\_fails**=**2 fail\_timeout**=**30s**;**
51. **}**
52. server **{**
53. listen 8000**;**
54. server\_name localhost**;**
55. #charset koi8-r;
56. #access\_log logs/host.access.log main;
57. #设置 group 的负载均衡参数
58. location **/**group1**/**M00 **{**
59. proxy\_next\_upstream http\_502 http\_504 error timeout invalid\_header**;**
60. proxy\_cache http-cache**;**
61. proxy\_cache\_valid 200 304 12h**;**
62. proxy\_cache\_key **$uri$is\_args$args;**
63. proxy\_pass http://fdfs\_group1;
64. expires 30d**;**
65. **}**
66. location **/**group2**/**M00 **{**
67. proxy\_next\_upstream http\_502 http\_504 error timeout invalid\_header**;** proxy\_cache http-cache**;**
68. proxy\_cache\_valid 200 304 12h**;**
69. proxy\_cache\_key **$uri$is\_args$args;**
70. proxy\_pass http://fdfs\_group2;
71. expires 30d**;**
72. **}**
73. #设置清除缓存的访问权限
74. location **~/**purge**(/.\*)** **{**
75. allow 127**.**0**.**0**.**1**;**
76. allow 192**.**168**.**1**.**0**/**24**;**
77. deny all**;**
78. proxy\_cache\_purge http-cache **$uri$is\_args$args;**
79. **}**
80. #error\_page 404 /404.html;
81. # redirect server error pages to the static page /50x.html
82. error\_page 500 502 503 504 **/**50x.html**;**
83. location **=** **/**50x.html **{**
84. root html**;**
85. **}**
86. **}**

按以上 nginx 配置文件的要求，创建对应的缓存目录：

1. shell**>** mkdir -p **/**fastdfs**/**cache**/**nginx**/**proxy\_cache
2. shell**>** mkdir -p **/**fastdfs**/**cache**/**nginx**/**proxy\_cache**/**tmp
   1. 防火墙打开Nginx 8000 端口
3. shell**>** vi **/**etc**/**sysconfig**/**iptables
4. ## 添加如下配置
5. -A INPUT **-**m state **--**state NEW **-**m tcp -p tcp **--**dport 8000 **-**j ACCEPT
6. shell**>** service iptables restart # 重新启动防火墙
   1. 启动Nginx
7. shell**>** **/**opt**/**nginx**/**sbin**/**nginx

设置开机启动：

1. shell**>** vi **/**etc**/**rc.local
2. ## 加入以下配置
3. **/**opt**/**nginx**/**sbin**/**nginx
4. shell**>** chmod **+**x **/**etc**/**rc.local #centos7
   1. 文件访问测试

**前面直接通过访问Storage节点中的Nginx访问文件：**

<http://192.168.1.202:8888/group1/M00/00/00/wKgBh1Xtr9-AeTfWAAVFOL7FJU4.tar.gz>

<http://192.168.1.204:8888/group2/M00/00/00/wKgBiVXtsDmAe3kjAAVFOL7FJU4.tar.gz>

**现在可以通过Tracker中的Nginx来进行访问：**   
(1)、通过 Tracker1 中的 Nginx 来访问   
<http://192.168.1.200:8000/group1/M00/00/00/wKgBh1Xtr9-AeTfWAAVFOL7FJU4.tar.gz>

<http://192.168.1.200:8000/group2/M00/00/00/wKgBiVXtsDmAe3kjAAVFOL7FJU4.tar.gz>

(2)、通过 Tracker2 中的 Nginx 来访问   
<http://192.168.1.201:8000/group1/M00/00/00/wKgBh1Xtr9-AeTfWAAVFOL7FJU4.tar.gz>

<http://192.168.1.201:8000/group2/M00/00/00/wKgBiVXtsDmAe3kjAAVFOL7FJU4.tar.gz>

由上面的文件访问效果可以看到，每一个Tracker中的Nginx都单独对后端的Storage组做了负载均衡，但整套FastDFS集群，如果想对外提供统一的文件访问地址，还需要对两个Tracker中的Nginx进行HA集群

1. **配置Tracker服务器高可用反向代理与负载均衡**

使用Keepalived + Nginx组成的高可用负载均衡集群，做两个Tracker节点中Nginx的负载均衡。

* 1. 安装keepalived与Nginx

分别在192.168.1.206和192.168.1.207两个节点安装Keepalived与Nginx。   
keepalived安装与配置：<http://blog.csdn.net/xyang81/article/details/52554398>   
Nginx的安装与配置：<http://blog.csdn.net/xyang81/article/details/51476293>

* 1. 配置Keeyalived Nginx高可用

请参考[《Keepalived+Nginx实现高可用（HA）》](http://blog.csdn.net/xyang81/article/details/52556886)

**注意：将VIP的IP地址修改为192.168.1.208**

* 1. 配置nginx对tracker节点的负载均衡

2个节点的Nginx配置相同，如下所示：

1. shell**>** vi **/**opt**/**nginx**/**conf**/**nginx.conf
2. user root**;**
3. worker\_processes 1**;**
4. events **{**
5. worker\_connections 1024**;**
6. use epool**;**
7. **}**
8. http **{**
9. include mime.types**;**
10. default\_type application**/**octet-stream**;**
11. sendfile on**;**
12. keepalive\_timeout 65**;**
13. ## FastDFS Tracker Proxy
14. upstream fastdfs\_tracker **{**
15. server 192**.**168**.**1**.**200**:**8000 weight**=**1 max\_fails**=**2 fail\_timeout**=**30s**;**
16. server 192**.**168**.**1**.**201**:**8000 weight**=**1 max\_fails**=**2 fail\_timeout**=**30s**;**
17. **}**
18. server **{**
19. listen 80**;**
20. server\_name localhost**;**
21. location **/** **{**
22. root html**;**
23. index index.html index.htm**;**
24. **}**
25. error\_page 500 502 503 504 **/**50x.html**;**
26. location **=** **/**50x.html **{**
27. root html**;**
28. **}**
29. ## FastDFS Proxy
30. location **/**dfs **{**
31. root html**;**
32. index index.html index.htm**;**
33. proxy\_pass http**://**fastdfs\_tracker**/;**
34. proxy\_set\_header Host **$http\_host;**
35. proxy\_set\_header Cookie **$http\_cookie;**
36. proxy\_set\_header X-Real-IP **$remote\_addr;**
37. proxy\_set\_header X-Forwarded-For **$proxy\_add\_x\_forwarded\_for;**
38. proxy\_set\_header X-Forwarded-Proto **$scheme;**
39. client\_max\_body\_size 300m**;**
40. **}**
41. **}**
42. **}**
    1. 重启1921681206 和 1921681207 中的Nginx
43. shell**>** **/**opt**/**nginx**/**sbin**/**nginx -s reload
    1. 通过虚拟IP访问文件测试

现在可以通过 Keepalived+Nginx 组成的高可用负载集群的 VIP(192.168.1.208)来访问 FastDFS 集群中的文件了：

<http://192.168.1.208/dfs/group1/M00/00/00/wKgBh1Xtr9-AeTfWAAVFOL7FJU4.tar.gz>

<http://192.168.1.208/dfs/group2/M00/00/00/wKgBiVXtsDmAe3kjAAVFOL7FJU4.tar.gz>

注意：千万不要使用 kill -9 命令强杀 FastDFS 进程,否则可能会导致 binlog 数据丢失。