



黑客派

# FastDFS 安装配置

作者: [StefanYang](#)

原文链接: <https://hacpai.com/article/1544166679491>

来源网站: 黑客派

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

<p>这篇文章介绍如何搭建 FastDFS 集群<br> 不明白什么是 FastDFS 的小伙伴可以先看看上一篇文章 <a href="https://link.hacpai.com/forward?goto=http%3A%2F%2Fwww.honeybee.top%2Farticles%2F2018%2F12%2F06%2F1544065761459.html" target="\_blank" rel="nofollow ugc">FastDFS 分布式文件系统详解</a></p>

<script async src="https://pagead2.googlesyndication.com/pagead/js/adsbygoogle.js"></script>

<!-- 黑客派PC帖子内嵌-展示 -->

<ins class="adsbygoogle" style="display:block" data-ad-client="ca-pub-5357405790190342" data-ad-slot="8316640078" data-ad-format="auto" data-full-width-responsive="true"></ins>

<script>

(adsbygoogle = window.adsbygoogle || []).push({});

</script>

<hr>

<h3 id="安装FastDFS">安装 FastDFS</h3>

<p>安装依赖包 (在安装 FastDFS 和 Nginx 之前, 需确保 gcc、gcc-c++、libstdc++-devel、make 等依赖库和工具已经安装) : <br> <code>yum -y install gcc gcc-c++ libstdc++-devel pcre-devel zlib-devel wget make</code></p>

<p><strong>安装 libfastcommon:</strong></p>

<p>安装 FastDFS 必须先安装 libfastcommon 类库, 否则会导致报错</p>

<ul>

<li><p>下载源包: <br> <code>cd /usr/local</code><br> <code>wget https://github.com/happyfish100/libfastcommon/archive/V1.0.7.tar.gz</code></p> </li>

<li><p>解压: <br> <code>tar -xvf V1.0.7.tar.gz</code></p> </li>

<li><p>编译安装: <br> <code>cd libfastcommon-1.0.7</code><br> <code>./make.sh</code><br> <code>./make.sh install</code></p> </li>

</ul>

<p><strong>安装 FastDFS:</strong></p>

<ul>

<li><p>下载 FastDFS:<br> <code>cd /usr/local</code><br> <code>wget https://github.com/happyfish100/fastdfs/archive/V5.05.tar.gz</code></p> </li>

<li><p>解压: <br> <code>tar -xvf V5.05.tar.gz</code></p> </li>

<li><p>编译、安装: <br> <code>cd fastdfs-5.05</code><br> <code>./make.sh</code><br> <code>./make.sh install</code></p> </li>

</ul>

<p>安装好之后, 在/usr/bin 目录下, 可以看 fdfs 开头的命令工具<br> 作者提供的示例配置文件在 etc/fdfs 目录下, tracker 需要 tracker.conf 配置文件, storage 需要 storage.conf 配置文件。</p>

<h3 id="配置Tracker">配置 Tracker</h3>

<p>将 tracker.conf.sample 文件复制为 tracker.conf, 然后修改 tracker.conf 文件<br> <code>cd /etc/fdfs</code><br> <code>cp tracker.conf.sample tracker.conf</code><br> <code>vim tracker.conf</code></p>

<p>这里我修改了两个配置: </p>

<pre><code class="highlight-chroma">##Tracker 数据和日志目录地址

<code>base\_path=/data/fastdfs

<code>##HTTP 服务端口

<code>http.server\_port=8081

</code></pre>

<p>还有一些配置说明: </p>

<pre><code class="highlight-chroma">##配置文件是否不生效, false 为生效<br>disabled=false

##提供服务的端口

```
port=22122
```

```
##Tracker 数据和日志目录地址(这个目录需要手动创建)
```

```
base_path=/data/fastdfs
```

```
##HTTP 服务端口
```

```
http.server_port=8081
```

```
</code></pre>
```

<p>手动创建下文件目录: <br> <code>mkdir /data/fastdfs</code> </p>

<p>测试下启动 Tracer:<br> <code>/usr/bin/fdfs\_trackerd /etc/fdfs/tracker.conf start</code>  
<br> 确认是否启动成功, 查看 22122 端口是否开始监听: <br> <code>netstat -unltp|grep fdfs</code> </p>

<p> </p>

<p>Tracker 服务启动成功后, 也会在 base\_path 下创建 data、logs 两个目录。 <br>  </p>

### ``` <script async src="https://pagead2.googlesyndication.com/pagead/js/adsbygoogle.js"></script> ``` ``` <!-- 黑客派PC帖子内嵌-展示 --> ``` ``` <ins class="adsbygoogle" style="display:block" data-ad-client="ca-pub-5357405790190342" data-ad-slot="8316640078" data-ad-format="auto" data-full-width-responsive="true"></ins> ``` ``` <script> ``` ``` (adsbygoogle = window.adsbygoogle || []).push({}); ``` ``` </script> ``` <p>同样的复制示例配置文件 storage.conf.sample 后修改: <br> <code>cd /etc/fdfs</code> <br> <code>cp storage.conf.sample storage.conf</code> <br> <code>vim storage.conf</code> </p> <p>这里修改了如下配置: </p> ``` <pre><code class="highlight-chroma">##Storage 数据和日志目录地址(这个目录需要手动创建) base_path=/data/fastdfs/storage ``` ``` ##文件存储目录 ``` ``` store_path0=/data/fastdfs/storage/file ``` ``` ##Tracker服务器IP和端口 ``` ``` tracker_server=`你的Tracker服务器IP`:23000 ``` ``` ##HTTP 服务端口 ``` ``` http.server_port=8082 ``` ``` </code></pre> ``` <p>其它的一些配置说明: </p> ``` <pre><code class="highlight-chroma">##配置文件是否不生效, false 为生效 disabled=false ``` ``` ##指定此 storage server 所在 组(卷) ``` ``` group_name=group1 ``` ``` ##storage server 服务端口 ``` ``` port=23000 ``` 原文链接: [FastDFS 安装配置](#)

##心跳间隔时间, 单位为秒 (这里是指主动向 tracker server 发送心跳)

```
heart_beat_interval=30
```

##Storage 数据和日志目录地址(这个目录需要手动创建)

```
base_path=/data/fastdfs/storage
```

##存储路径个数, 需要和store\_path个数匹配

```
store_path_count=1
```

##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...

```
store_path0=/data/fastdfs/storage/file
```

##FastDFS 存储文件时, 采用了两级目录。这里配置存放文件的目录个数。

##如果本参数只为 N (如: 256), 那么 storage server 在初次运行时, 会在 store\_path 下自动建 N \* N 个存放文件的子目录

```
subdir_count_per_path=256
```

##tracker\_server 的列表, 会主动连接 tracker\_server

##有多个 tracker server 时, 每个 tracker server 写一行

```
tracker_server= [tracker_server ip]:23000
```

```
</code></pre>
```

<p>手动创建下文件目录: <br> <code>mkdir /data/fastdfs/storage</code></p>

<p>测试启动 Storage:<br> <strong>这边要确保服务器端口已经开放</strong><br> <code>/usr/bin/fdfs\_storaged /etc/fdfs/storage.conf start</code></p>

```
<blockquote>
```

<p>这边一开始没有启动成功, 报错解决: <br> 查看日志: <code>cat /data/fastdfs/storage/logs/storaged.log</code> <br>  <br> 原因: 缺少目录/data/fastdfs/storage/file/data <br> <code>mkdir /data/fastdfs/storage/file/data</code></p>

```
</blockquote>
```

<p>确认是否启动成功, 查看 23000 端口是否开始监听: <br> <code>netstat -unltp | grep fdfs</code></p>

```
<script async src="https://pagead2.googlesyndication.com/pagead/js/adsbygoogle.js"></script>
```

```
<!-- 黑客派PC帖子内嵌-展示 -->
```

```
<ins class="adsbygoogle" style="display:block" data-ad-client="ca-pub-5357405790190342" data-ad-slot="8316640078" data-ad-format="auto" data-full-width-responsive="true"></ins>
```

```
</script>
```

```
(adsbygoogle = window.adsbygoogle || []).push({};
```

```
</script>
```

<p>启动 Storage 前确保 Tracker 是启动的。初次启动成功, 会在 base\_path 配置的路径下创建 da、logs 两个目录。如果看到 23000 端口正常被监听后, 这时候说明 Storage 服务启动成功啦</p>

<p>还可以验证下 Storage 是否登记到了 Tracker 服务器<br> 运行 fdfs\_monitor 查看 storage 服务器是否已经登记到 tracker 服务器: <code>fdfs\_monitor /etc/fdfs/storage.conf</code> <br> 如果出现 <code>ip\_addr = Active</code> 行, 则表明 storage 服务器已经登记到 tracker 服务器</p>

<h3 id="安装Nginx并配置fastdfs-nginx-module模块">安装 Nginx 并配置 fastdfs-nginx-module

模块

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进行文件复制，有同步延迟的问题。

假设 Tracker 服务器将文件上传到了 192.168.51.128，上传成功后文件 ID 已经返回给客户端。时 FastDFS 存储集群机制会将这个文件同步到同组存储 192.168.51.129，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 192.168.51.129 上取文件，就会出现文件无法访问的错误。而 fastdfs-nginx-module 可以重定向文件链接到源服务器取文件，避免客户端由于复制延迟导致的文件无法访问错误。

下载安装包并解压：

```
cd /usr/local
wget http://nginx.org/download/nginx-1.14.2.tar.gz
tar -zxvf nginx-1.14.2.tar.gz
```

下载fastdfs-nginx-module\_v1.16.tar.gz上传至服务器目录 /usr/local

下载地址: [https://pan.baidu.com/s/1hs3qp84#list/path=%2Fnot found render function for node \[type=NodeHTMLEntity, Tokens=&\]parentPath=%2FIT%E6%96%87%E6%A1%A3%E8%B5%84%E6%96%99](https://pan.baidu.com/s/1hs3qp84#list/path=%2Fnot%20found%20render%20function%20for%20node%20[type%3DNodeHTMLEntity%2C%20Tokens%3D%26]parentPath=%2FIT%E6%96%87%E6%A1%A3%E8%B5%84%E6%96%99)

上传好后解压:

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

安装之前需要先配置下软连接:

```
ln -sv /usr/include/fastcommon /usr/local/include/fastcommon
```

```
ln -sv /usr/include/fastdfs /usr/local/include/fastdfs
```

```
ln -sv /usr/lib64/libfastcommon.so /usr/local/lib/libfastcommon.so
```

在安装 Nginx 之前，先添加 fastdfs-nginx-module-master 模块：

```
cd /usr/local/nginx-1.14.2
./configure --prefix=/usr/local/nginx --add-module=/usr/local/fastdfs-nginx-module/src
```

```
<script async src="https://pagead2.googlesyndication.com/pagead/js/adsbygoogle.js"></script>
```

```
<!-- 黑客派PC帖子内嵌-展示 -->
```

```
<ins class="adsbygoogle" style="display:block" data-ad-client="ca-pub-5357405790190342" data-ad-slot="8316640078" data-ad-format="auto" data-full-width-responsive="true"></ins>
```

```
<script>
```

```
(adsbygoogle = window.adsbygoogle || []).push({});
```


```
</script>
```

成功后的输出结果:  alt="imagepng" data-src="http://file.honeybee.top//file/2018/12/d92154853c4f421a80616258602df03\_image.png"/>

编译: `make`

安装: `make install`

查看 Nginx 版本信息: `/usr/local/nginx/sbin/nginx -V`

编译的时候有个报错解决:  alt="imagepng" data-src="http://file.honeybee.top//file/2018/12/8ae5fce81b5249f69e9d88d29223064\_image.png"/> 修改以下 fastdfs-nginx-module 的配置文件: `vi /usr/local/fastdfs-nginx-module-master/src/config` 修改项: `CORE_INCS="$CORE_INCS /usr/include/fastdfs /usr/include/fastcommon/"` 重新添加

块:   
 `cd /usr/local/nginx-1.14.2`   
 `./configure --prefix=/usr/local/nginx --add-module=/usr/local/fastdfs-nginx-module/src`   
 重新编译:   
 `make`   
 编译成功。

</blockquote>

<h4 id="配置">配置</h4>

<p><strong>配置 fastdfs-nginx-module</strong></p>

<p>复制 fastdfs-nginx-module 源码中的配置文件到/etc/fdfs 目录:   
 `cp /usr/local/fastdfs-nginx-module/src/mod_fastdfs.conf /etc/fdfs/`   
 修改这个配置文件:   
 `vim /etc/fdfs/mod_fastdfs.conf` </p>

```
<pre><code class="highlight-chroma">##保存日志目录
base_path=/data/fastdfs/storage
##tracker服务连接信息
tracker_server=[tracker_server_ip]:22122
##storage服务器的端口号
storage_server_port=23000
##当前服务器的group名
group_name=group1
##文件url是否有group名
url_have_group_name=true
##存储路径个数,需要和store_path个数匹配
store_path_count=1
##存储路径,必须和storage.conf中配置的store_path0路径一致
store_path0=/data/fastdfs/storage/file
##设置组的个数
group_count=0
```

</code></pre>

<p>复制 FastDFS 的部分配置文件到/etc/fdfs 目录:   
 `cp /usr/local/fastdfs-5.05/conf/http.conf /usr/local/fastdfs-5.05/conf/mime.types /etc/fdfs` </p>

<p><strong>配置 nginx</strong></p>

<p>编辑 nginx 的配置文件:   
 `vim /usr/local/nginx/conf/nginx.conf` </p>

```
<pre><code class="highlight-chroma">##将server段中的listen端口号修改为8080
listen 8080;
```

##在server段中添加:

```
location ~/group[0-9]/M00 {
root /data/fastdfs/data;
ngx_fastdfs_module;
}
```

</code></pre>

<p>启动 nginx: `/usr/local/nginx/sbin/nginx`   
 确认 nginx 是否启动成功:   
 `netstat -unltp | grep nginx`   
 这个时候打开浏览器,访问 8080 端口的页面,可看到 nginx 的欢迎页。 </p>

```
<script async src="https://pagead2.googlesyndication.com/pagead/js/adsbygoogle.js"></script>
```

```
<!-- 黑客派PC帖子内嵌-展示 -->
```

```
<ins class="adsbygoogle" style="display:block" data-ad-client="ca-pub-5357405790190342" data-ad-slot="8316640078" data-ad-format="auto" data-full-width-responsive="true"></ins>
```

```
<script>
  (adsbygoogle = window.adsbygoogle || []).push({});
</script>
<h3 id="测试上传">测试上传</h3>
<p><strong>需要配置下 client.conf 文件</strong></p>
<p><code>cp /etc/fdfs/client.conf.sample /etc/fdfs/client.conf</code></p>
<p>修改以下参数: </p>
<pre><code class="highlight-chroma">##日志存放路径
base_path=/data/fastdfs
```

tracker\_server=[tracker服务器ip]:22122

http.tracker\_server\_port=8080

</code></pre>

<p>测试上传一个文件: <br> <code>/usr/bin/fdfs\_upload\_file /etc/fdfs/client.conf /data/7418.pg</code><br> 返回: <br> </p>

<p>使用浏览器访问: <br> <code>http://[tracker服务器ip]:8080/group1/M00/00/00/rBPhW1KFbWASYfCAACUpBwVaA219.jpg</code></p>

<p><p>

<p>有看到上传的图片, 说明搭建成功啦~~! </p>

<hr>

<p><strong>参考: </strong></p>

<p><a href="https://link.hacpai.com/forward?goto=https%3A%2F%2Fgithub.com%2Fhappyfish100%2F" target="\_blank" rel="nofollow ugc">官方网站</a><br> <a href="https://link.hacpai.com/forward?goto=https%3A%2F%2Fgithub.com%2Fhappyfish100%2Ffastdfs%2Fwiki%2F" target="\_blank" rel="nofollow ugc">配置文档</a><br> <a href="https://link.hacpai.com/forward?goto=https%3A%2F%2Fwww.oschina.net%2Fquestion%2Ftag%2Ffastdfs" target="\_blank" rel="nofollow ugc">参考资料</a><br> <a href="https://link.hacpai.com/forward?goto=http%3A%2F%2Fjuejin.im%2Fentry%2F5b7c1fe0f265da43275d2568" target="\_blank" rel="nofollow ugc">架构之路搭建 FastDFS 分布式文件系统</a></p>