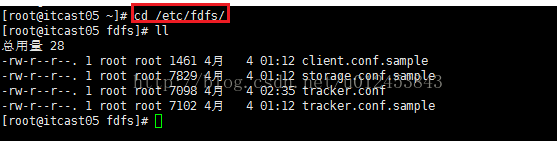
  上一篇博客，我们一起搭建了storage和tracker集群，FastDFS集群还未搭建完毕，不过由于篇幅过长，因此需要分成两篇来写。

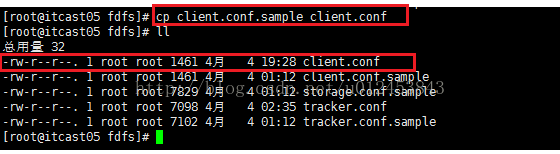
一、tracker和storage集群上传图片测试

        由于目前还没有搭建完集群，因此我们暂且在tracker的一台设备上（我们就选择192.168.156.5这台设备）使用client来上传图片进行测试。

        1.进入到/etc/fdfs/目录下，我们可以看到client.conf.sample这么一个配置文件，如下图所示。

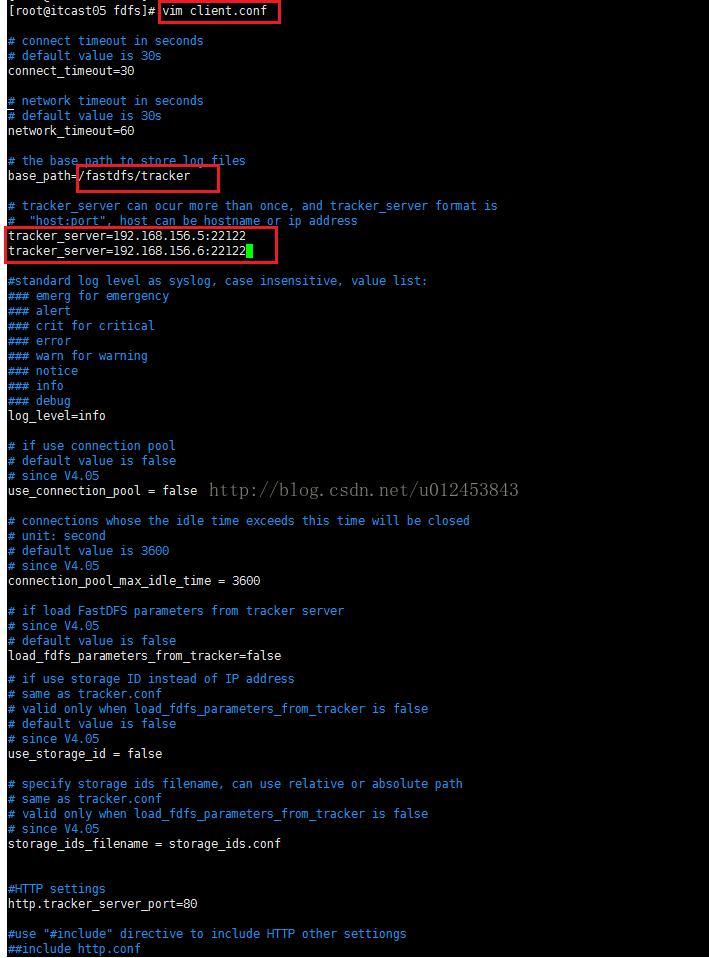


2.我们使用命令：cp client.conf.sample client.conf复制一份该文件并命名为client.conf，如下图所示。

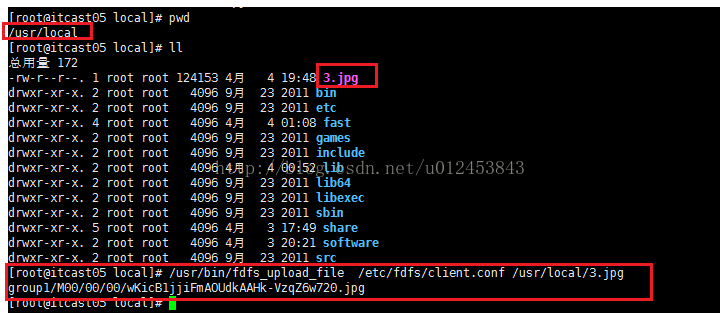


  3.修改client.conf配置文件，使用命令：vim client.conf，需要修改的配置有base\_path=/fastdfs/tracker、tracker\_server=192.168.156.5:22122和

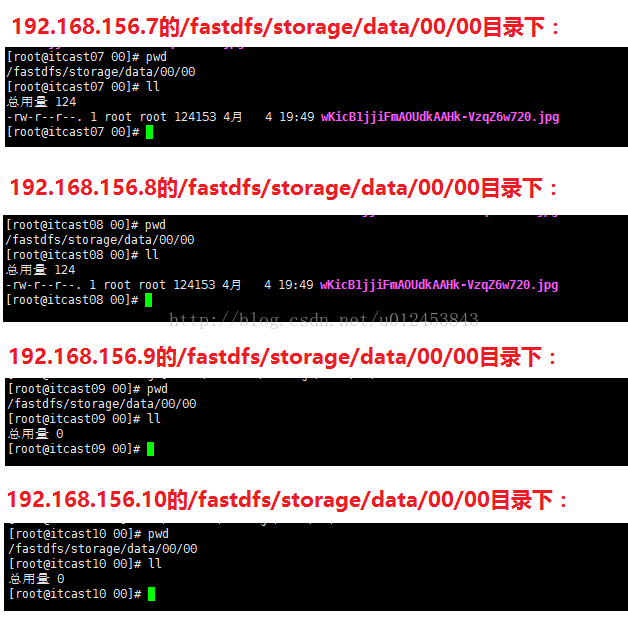
tracker\_server=192.168.156.6:22122，如下图所示。其中/fastdfs/tracker这个目录我们已经创建过了，因此这时不需要再创建了。



 4.下面我们来上传一张图片，我把/usr/local目录下一张3.jpg图片上传，使用的命令：/usr/bin/fdfs\_upload\_file  /etc/fdfs/client.conf /usr/local/3.jpg，可以看到这条命令由3部分组成，第一部分是/usr/bin/fdfs\_upload\_file，意思是指定要进行上传文件操作，第二部分是/etc/fdfs/client.conf，意思是指定上传操作使用的配置文件，这个配置文件就是我们上面刚配置过的client.conf文件，第三部分是/usr/local/3.jpg，意思是指定要上传哪个目录下的哪个文件。按回车执行上传命令后，会返回一个串：group1/M00/00/00/wKicB1jjiFmAOUdkAAHk-VzqZ6w720.jpg，其中group1表示这张图片被保存在了哪个组当中，M00代表磁盘目录，如果电脑只有一个磁盘那就只有M00， 如果有多个磁盘，那就M01、M02...等等。00/00代表磁盘上的两级目录，每级目录下是从00到FF共256个文件夹，两级就是256\*256个。wKicB1jjiFmAOUdkAAHk-VzqZ6w720.jpg表示被存储到storage上的3.jpg被重命名的名字，这样做的目的是为了防止图片名字重复。



 我们到两组group所在的四台设备的/fastdfs/storage/data/00/00目录下查看一下是否有我们刚才上传的图片，发现192.168.156.7和192.168.156.8两台设备上有该图片，而192.168.156.9和192.168.156.10两台设备上没有该图片。这是由于192.168.156.7和192.168.156.8两台设备属于group1，而192.168.156.9和192.168.156.10属于group2，返回的图片信息中明确说明了存储在了group1下面，因此可group2下面是没有该图片的。



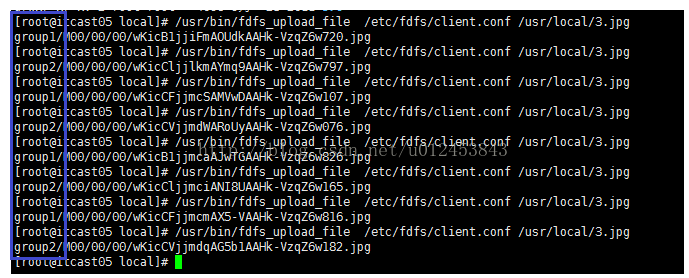
我们在搭建集群的时候，配置的策略是轮询策略，那么我们现在再上传一次该图片，看是否会存储到group2下面。如下图所示，发现这次返回的路径信息中显示存储到了group2下面。

https://img-blog.csdn.net/20170404115344681?watermark/2/text/aHR0cDovL2Jsb2cuY3Nkbi5uZXQvdTAxMjQ1Mzg0Mw==/font/5a6L5L2T/fontsize/400/fill/I0JBQkFCMA==/dissolve/70/gravity/Center

下面再到四台设备下的/fastdfs/storage/data/00/00目录下查看下图片信息，发现192.168.156.7和192.168.156.8这两台设备的图片还是刚才那一张，并没有新增图片。而192.168.156.9和192.168.156.10这两台设备上新增了我们刚才上传的图片，这说明第二次上传确实是存储到了group2上面。



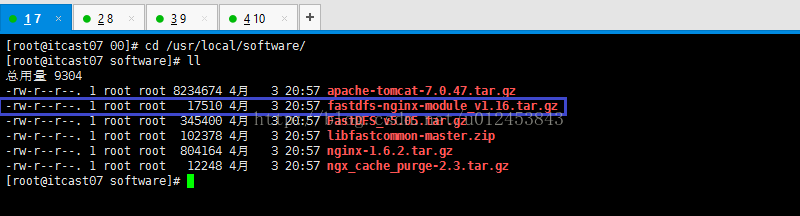
大家可以多次上传图片，就可以看到存储确实是一次group1，一次group2，轮询进行存储的，如下图所示。



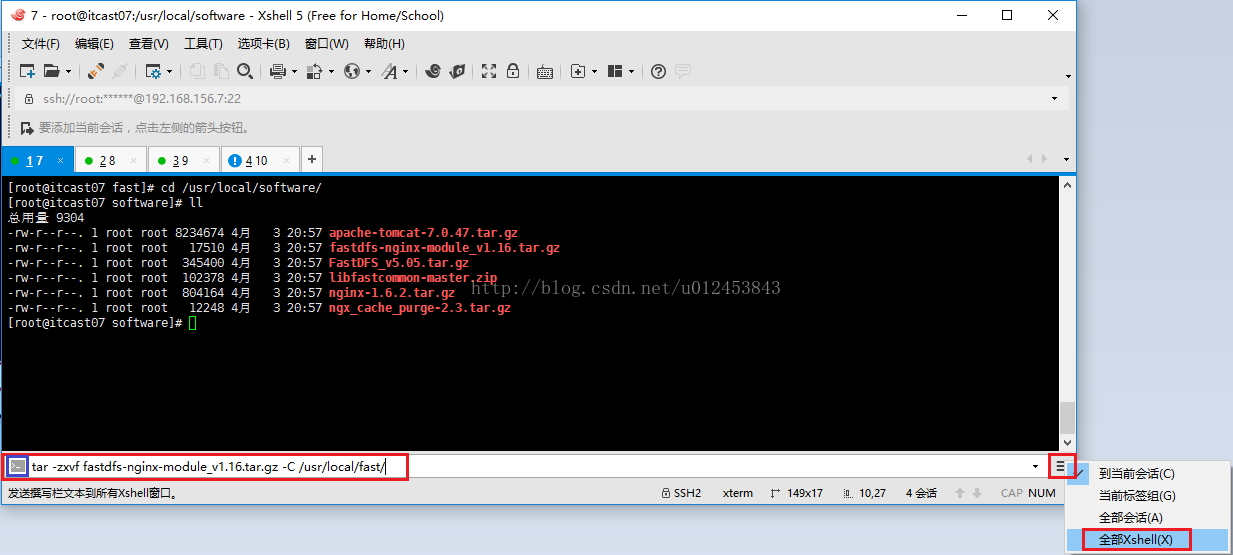
这里说一个情况，那就是同处一个组的某台设备如果发生了故障，那么这时上传的文件将只能存放到其它同组设备上，等故障设备修复后，同组的其它设备会自动将数据同步到该故障设备上，这个不用我们干预。

二、配置Ngnix

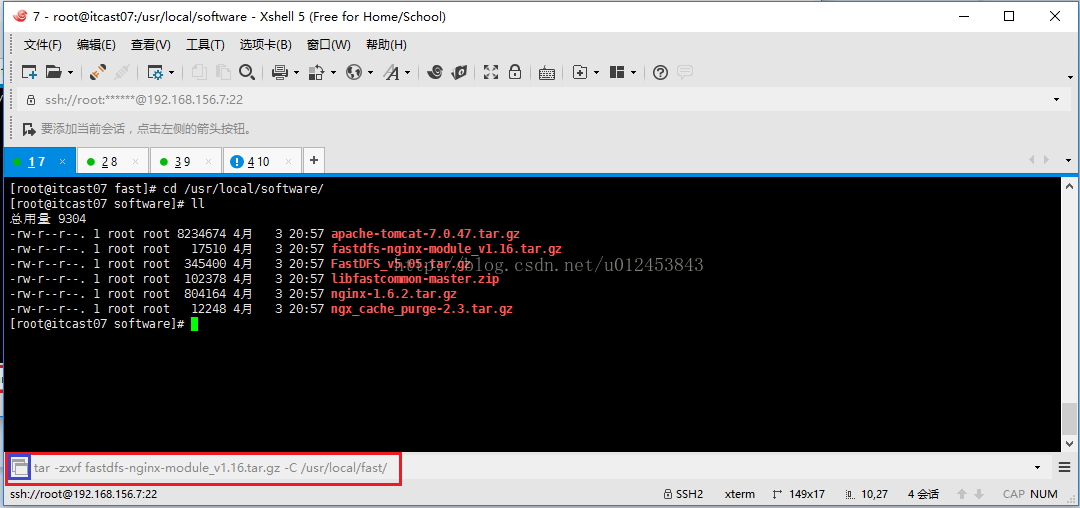
        1.到目前为止，我们还是不能够使用http方式来上传或下载文件的，因此我们需要配置下nginx来达到这个目的。我们先关闭除两个组所在的四台之外的其它所有设备，然后把剩下的这四台设备切换到/usr/local/software目录下，如下图所示。



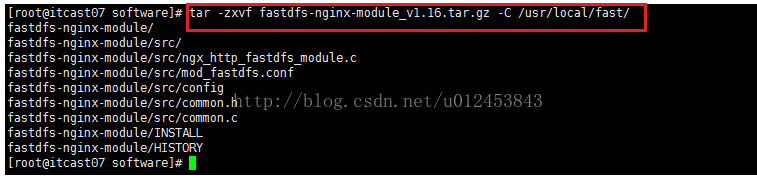
 2.这四台设备都需要执行同一个操作，那就是解压fastdfs-nginx-module\_v1.16.tar.gz，我们把命令：tar -zxvf fastdfs-nginx-module\_v1.16.tar.gz -C /usr/local/fast/复制到下面那个输入框中，默认情况下，在这个输入框中输入命令并按回车，命令执行的范围是当前会话，为了同时在四台设备上执行同样的命令，我们可以切换到"全部Xshell(X)"。如下图所示。



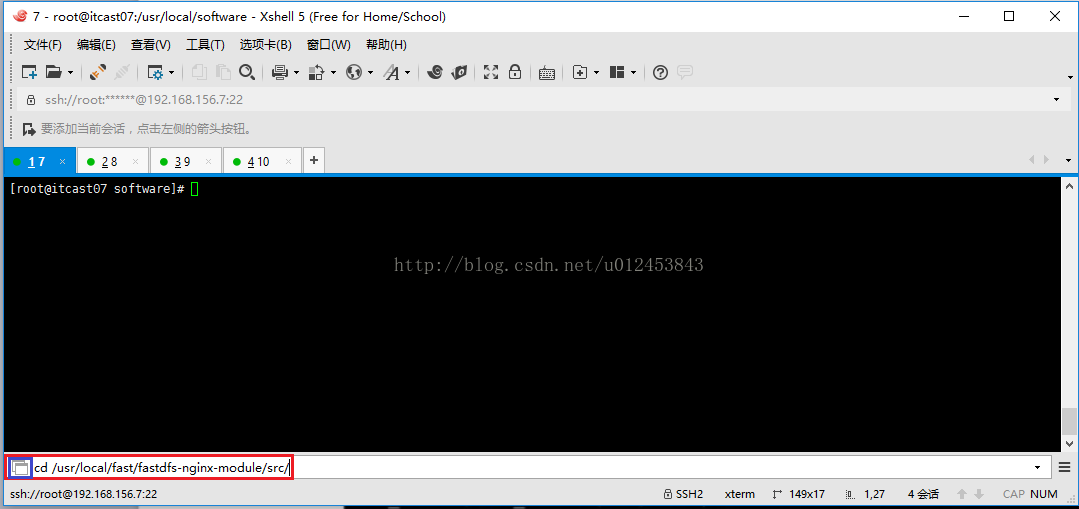
切换到"全部Xshell(X)"后，如下图所示，可以看到输入框左边的那个图标切换成了多个窗口的图标。这时我们把鼠标放到输入框中，按回车便可同时在四台设备上执行相同的命令。



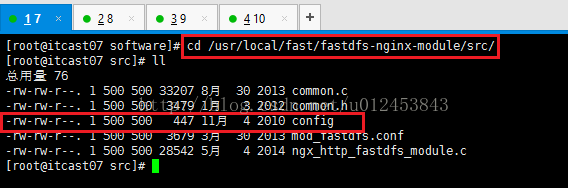
解压信息如下图所示，其它设备我就不一一截图了。



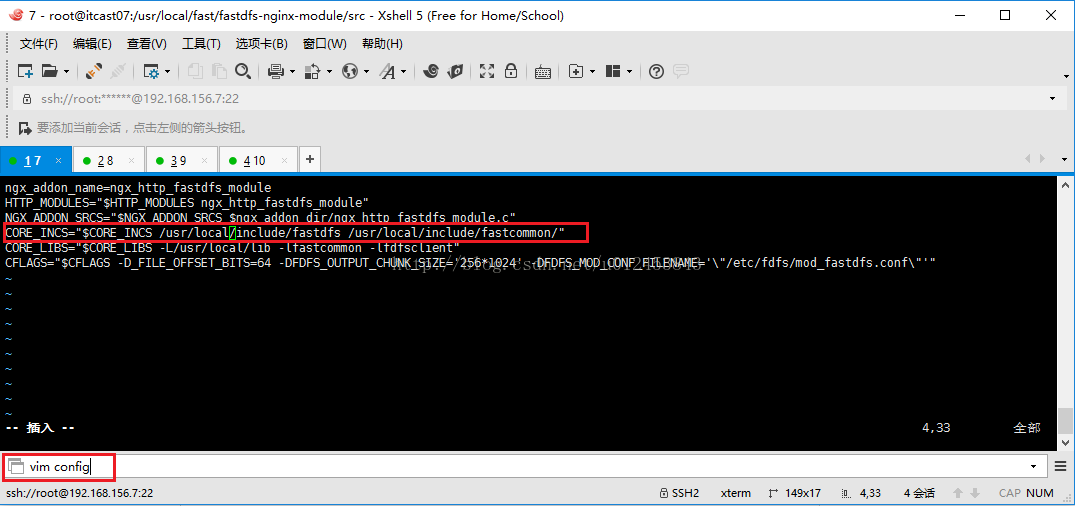
3.四台设备都需要进入到/usr/local/fast/fastdfs-nginx-module/src/这个目录下，因此我们把cd /usr/local/fast/fastdfs-nginx-module/src/命令放到下面的输入框中并按回车。



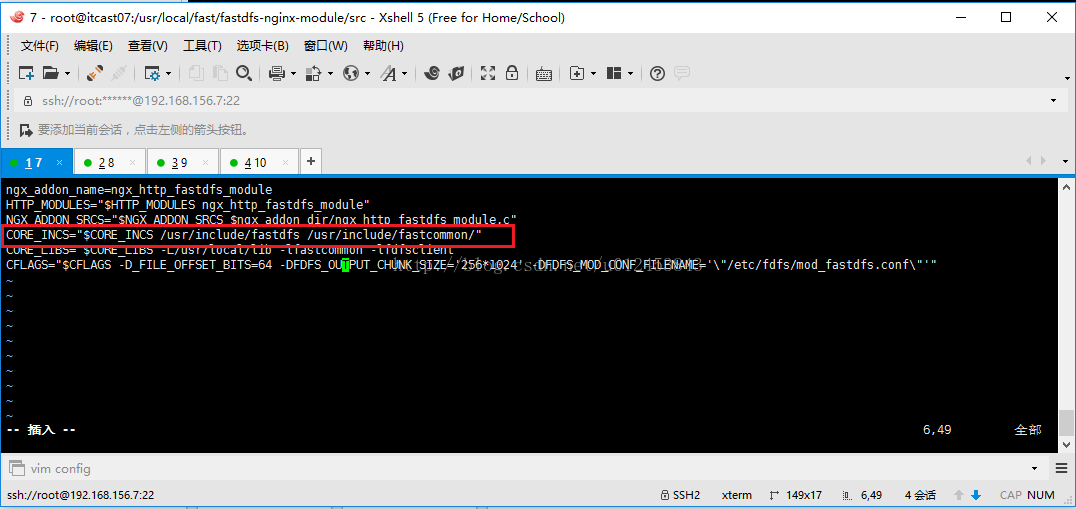
我们可以看到在/usr/local/fast/fastdfs-nginx-module/src/这个目录下有个config配置文件，我们需要修改下该文件。



     我们在下面的全局输入框中输入"vim config"同时打开四台设备的该配置文件，如下图所示，该配置文件的第四行我们需要做下修改，这是因为我们前面为了查看方便而把东西放到了/usr/local/include下，但是实际执行make install安装时却安装在了/usr/include下面，也就是我们多了一层local目录，因此我们需要把该行的两个local目录去掉。



去掉local目录后如下图所示。同样的，我们把其它三台设备的这行配置也都去掉local目录。



   4.安装nginx依赖包

        把下面四条命令执行一遍。

        yum install pcre

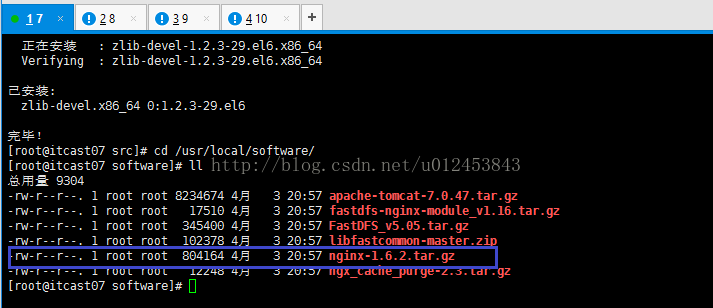
        yum install pcre-devel

        yum install zlib

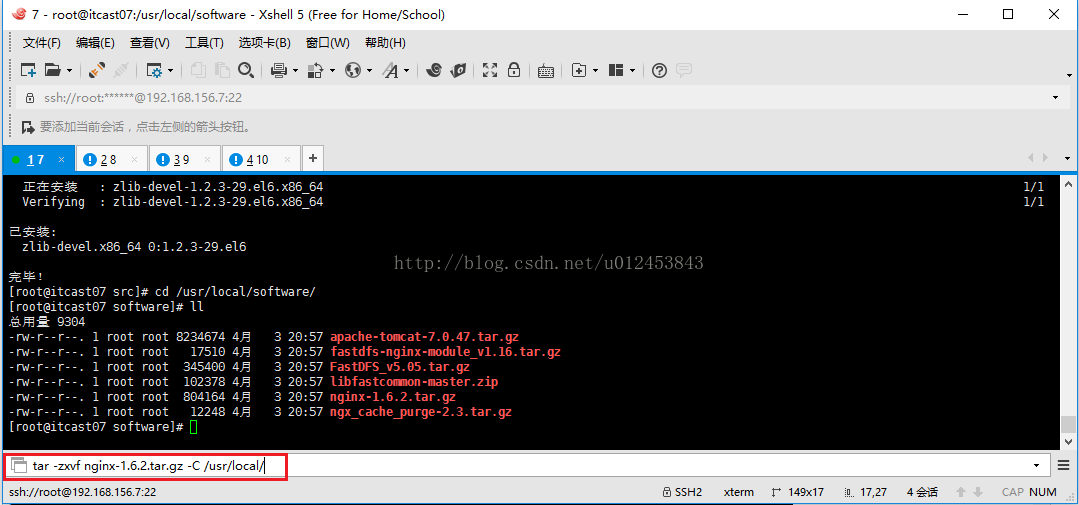
        yum install zlib-devel

      5.安装nginx

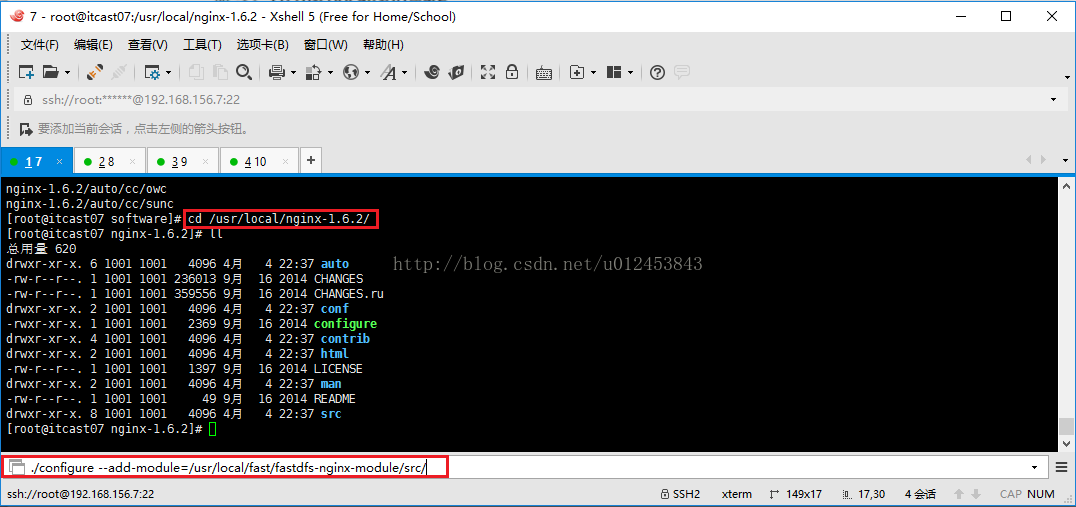
       四台设备都进入到/usr/local/software目录下，该目录下nginx-1.6.2.tar.gz是nginx安装包。



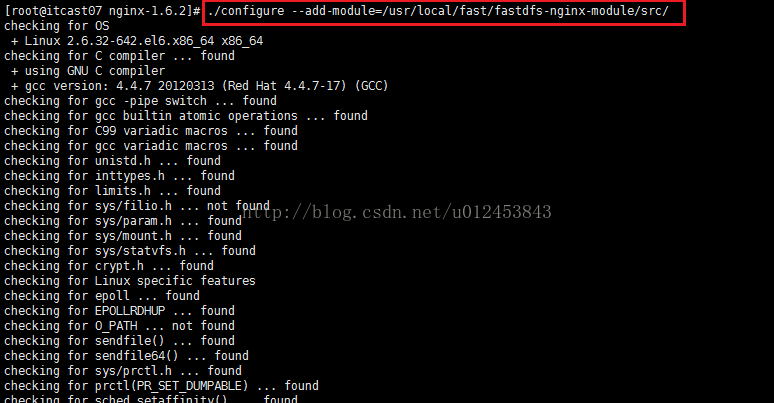
把解压命令：tar -zxvf nginx-1.6.2.tar.gz -C /usr/local/输入到下面的输入框中，回车即可在这四台设备上同时执行解压操作。



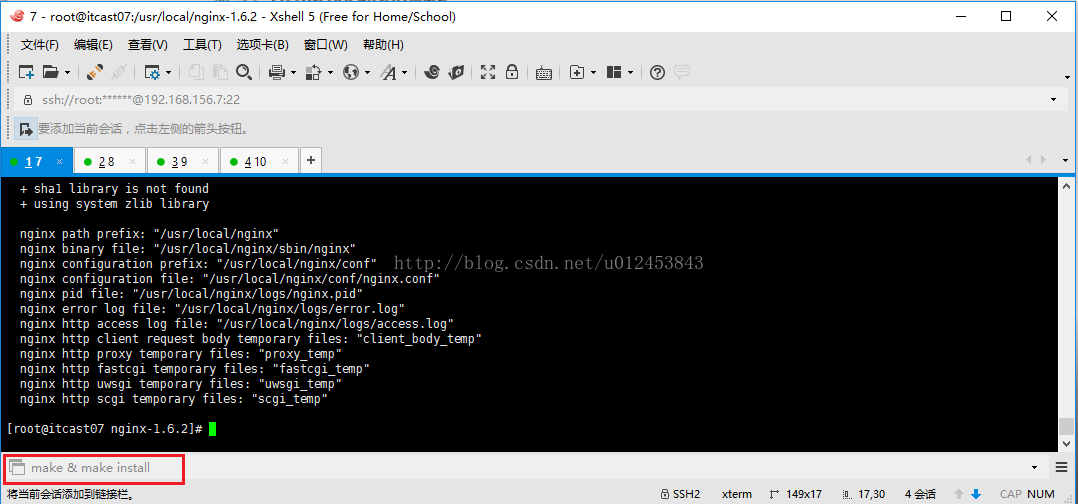
  安装完之后，我们都进入到/usr/local/nginx-1.6.2/目录下，然后在下面的输入框中输入命令：./configure --add-module=/usr/local/fast/fastdfs-nginx-module/src/并按回车，在四台设备上都加入下模块并进行检查。



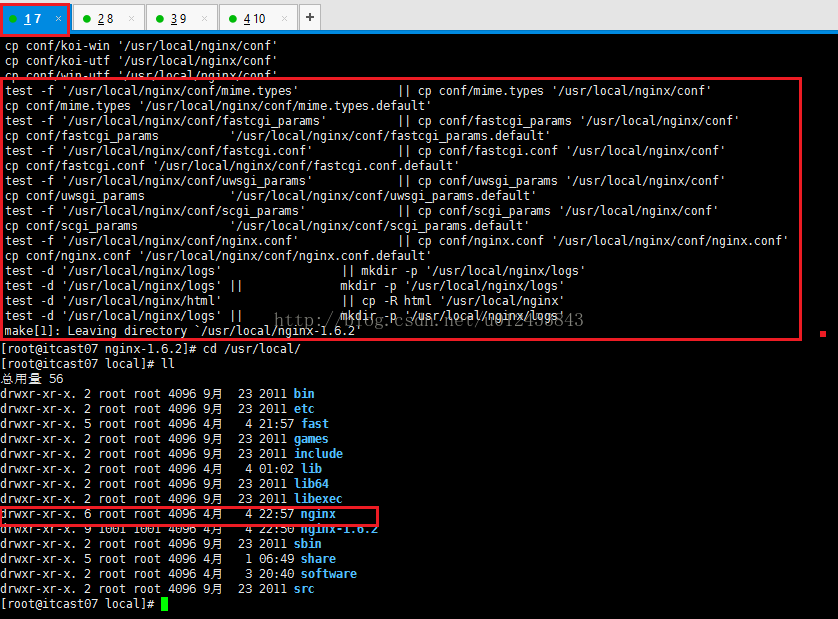
检查信息如下图所示。



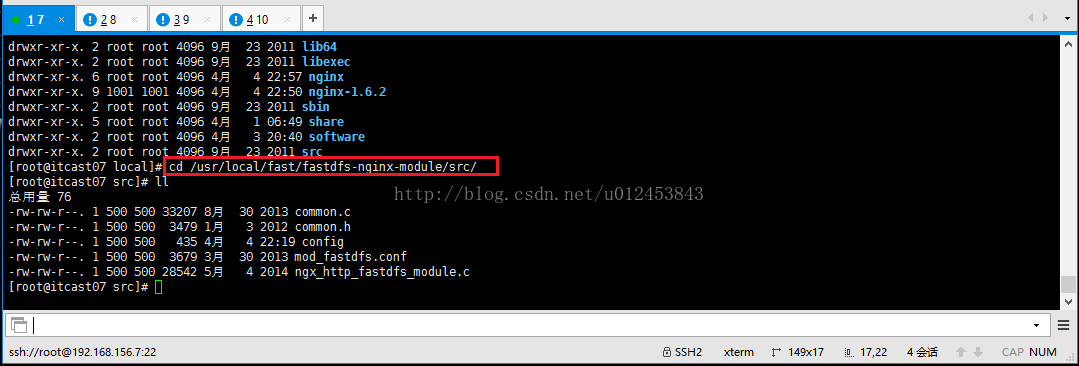
检查完之后，我们在下面输入框中输入命令make && make install，回车，四台设备同时执行编译安装操作。



安装信息中没有出现错误信息，说明我们的安装成功，安装完之后，我们到/usr/local/目录下，可以看到多了一个nginx的文件夹，如下图所示。



下面我们到/usr/local/fast/fastdfs-nginx-module/src/目录下，我们还是在输入框中输入cd /usr/local/fast/fastdfs-nginx-module/src/并回车，四台设备都进入到该目录下。



  我们将cp mod\_fastdfs.conf /etc/fdfs/复制到/etc/fdfs/目录下，我们在下面输入框中输入命令cp mod\_fastdfs.conf /etc/fdfs/并按回车，四台设备同时进行复制操作。

        复制完之后，我们修改/etc/fdfs/目录下的mod\_fastdfs.conf，我们使用命令vim /etc/fdfs/mod\_fastdfs.conf来编辑192.168.156.7上的该文件。

        需要修改的地方如下：

connect\_timeout=10

tracker\_server=192.168.156.5:22122

tracker\_server=192.168.156.6:22122

storage\_server\_port=23000//默认就是2300，不用做修改

url\_have\_group\_name=true

store\_path0=/fastdfs/storage

group\_name=group1

group\_count=2

[group1]

group\_name=group1

storage\_server\_port=23000

store\_path\_count=1

store\_path0=/fastdfs/storage

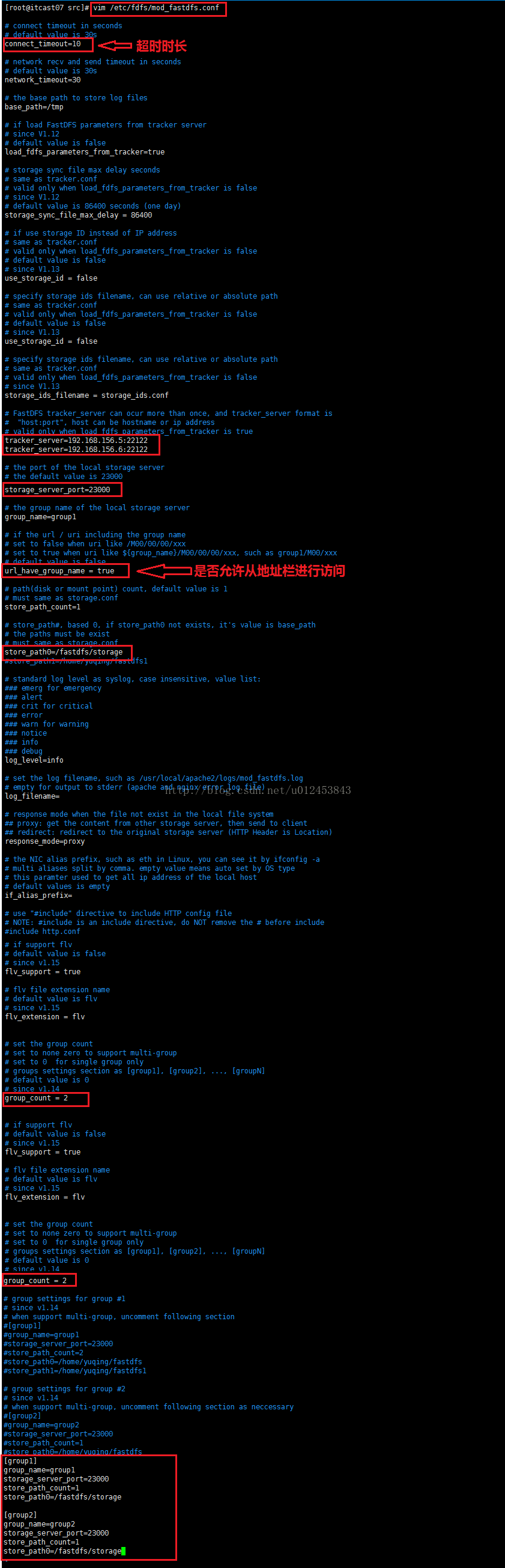
[group2]

group\_name=group2

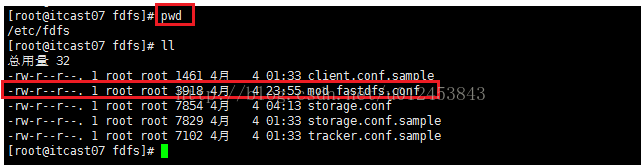
storage\_server\_port=23000

store\_path\_count=1

store\_path0=/fastdfs/storage

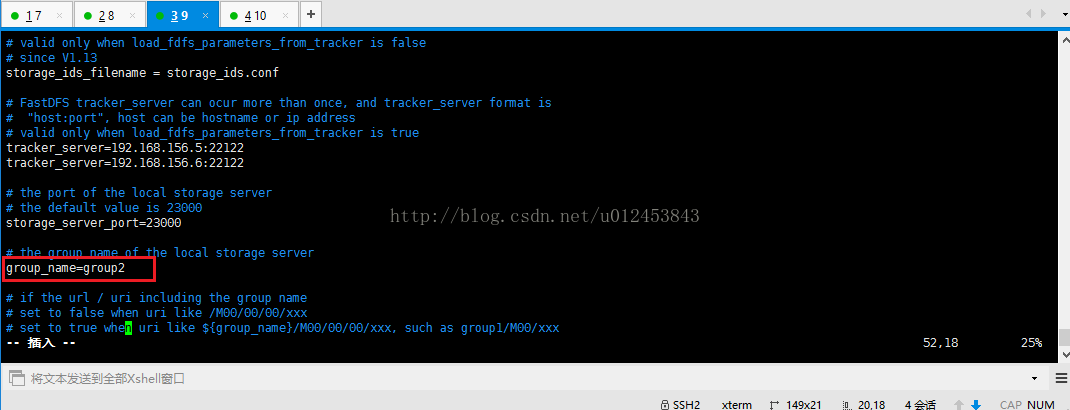


我们把192.168.156.7上的这个配置文件复制到其它三台设备上，先到/etc/fdfs/目录下，如下图所示。

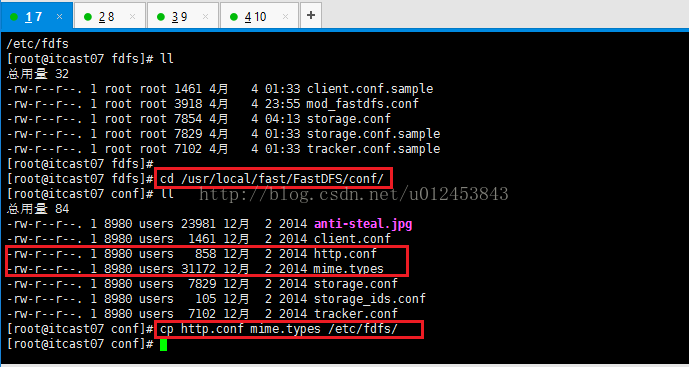


使用命令：scp  mod\_fastdfs.conf 192.168.156.8:/etc/fdfs/、scp  mod\_fastdfs.conf 192.168.156.9:/etc/fdfs/、scp  mod\_fastdfs.conf 192.168.156.10:/etc/fdfs/进行复制。

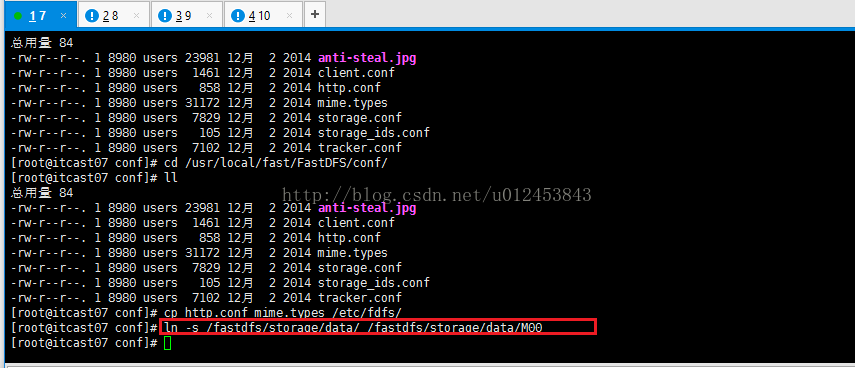
       复制完之后，我们到8、9、10这三台设备上看下配置文件是否就是我们刚才复制过去的文件。由于192.168.156.8这台设备与192.168.156.7这台设备同属group1，因此192.168.156.8这台设备不用做修改。我们只需把192.168.156.9和192.168.156.10这两台设备的这个配置文件的group名称改为group2即可。192.168.156.9的修改如下图所示，192.168.156.10与之一样。



接下来，我们需要把/usr/local/fast/FastDFS/conf/目录下的http.conf和mime.types两个文件复制到/etc/fdfs/目录下，由于这四台设备的这步操作都一样，因此我们在下面的输入框中输入cd /usr/local/fast/FastDFS/conf/并回车，都进入到该目录下，然后输入命令cp http.conf mime.types /etc/fdfs/并回车，在四台设备同时进行复制操作。



复制完之后，我们给四个节点都创建一下软链接，由于步骤都一样，因此在下面输入框中输入创建软链接命令：ln -s /fastdfs/storage/data/ /fastdfs/storage/data/M00并同时执行就可以了。如下图所示。

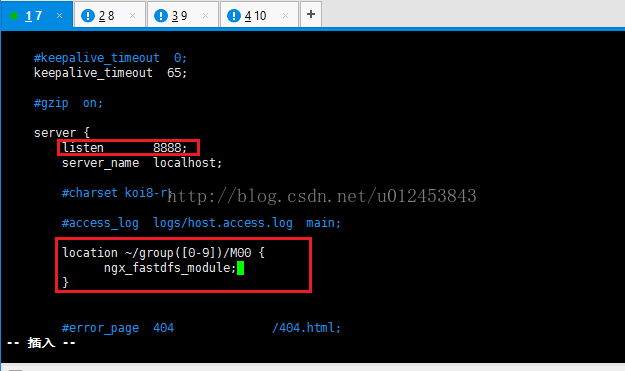


创建完软链接之后，我们来配置下nginx，由于四个节点操作一样，我们都进入到/usr/local/nginx/conf/这个目录下，并使用命令vim nginx.conf来编辑该文件，我们需要修改的地方有listen端口，把它由80改成8888，至于原因，是因为我们在上篇博客搭建storage的时候使用的端口是8888，因此这里也需要使用8888。另一个修改的地方是location，修改代码如下，路径采用正则表达式来匹配，匹配group0到group9下的M00前缀。由于fastdfs与nginx的模块结合，因此需要在location当中添加该模块。

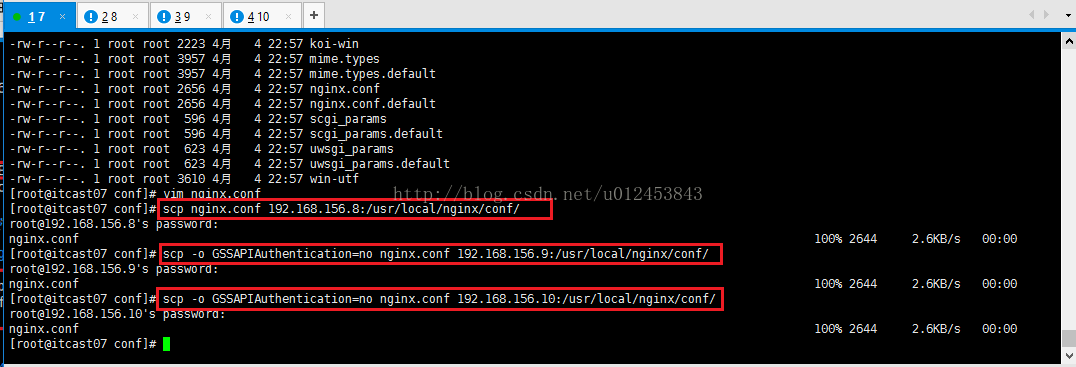
location ~/group([0-9])/M00 {

ngx\_fastdfs\_module;

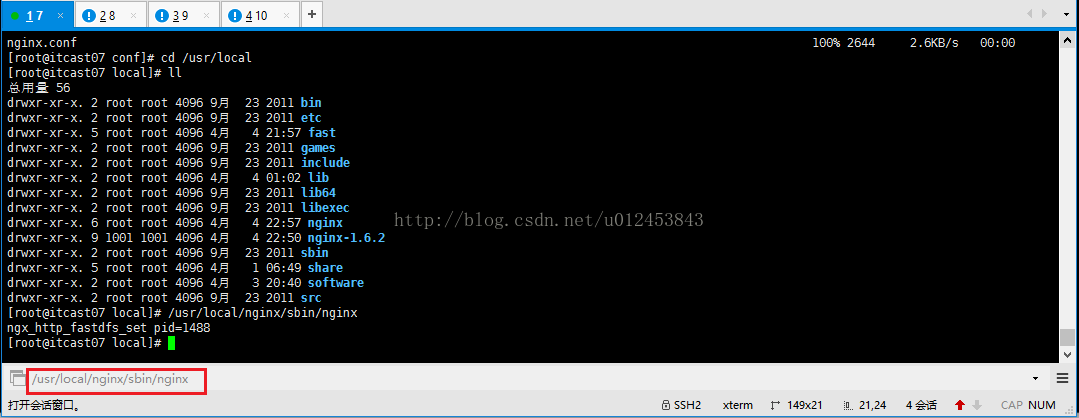
}



 修改完192.168.156.7之后，我们把这个配置文件再复制到其它三台设备上。在用scp复制文件到其它设备的时候，反应很慢，加上-o GSSAPIAuthentication=no会快点，但还是不够快，解决办法是我们可以在/etc/hosts文件当中配置所有要通信的设备的IP和名称的映射，这样scp就很快了。



6.四台设备都启动nginx，我们在下面的输入框中输入/usr/local/nginx/sbin/nginx并按回车，四台设备同时启动nginx。



7.启动完nginx之后，我们现在便可以通过http的方式访问上传到FastDFS上的文件了，比如我们现在再把那个3.jpg文件上传一次并拿回显地址去访问，如下所示（注意：上传是在192.168.156.5上进行的）

[root@itcast05 local]# /usr/bin/fdfs\_upload\_file /etc/fdfs/client.conf /usr/local/3.jpg

group1/M00/00/00/wKicCFjj1xqAcN8EAAHk-VzqZ6w619.jpg

[root@itcast05 local]#

我们在地址栏输入：http://192.168.156.7:8888/group1/M00/00/00/wKicCFjj1xqAcN8EAAHk-VzqZ6w619.jpg，这时我们访问不到图片，如下图所示。



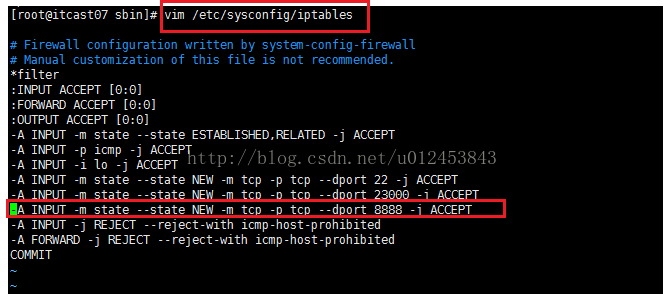
原因是我们的虚拟机的防火墙把端口8888给阻拦了，禁止外界访问，解决方法有两个：

       1.关闭虚拟机的防火墙，并禁止开启自启动，大家可以参考：http://blog.csdn.net/u012453843/article/details/52411019这篇博客进行防火墙的关闭操作。

       2.不关闭防火墙，只是让外界可以访问8888端口，这在实际环境中更实用，因此，我们采用方法2。

方法2的操作方法如下：

       使用命令：vim /etc/sysconfig/iptables打开编辑界面，如下图所示，我们添加的一行内容是：-A INPUT -m state --state NEW -m tcp -p tcp --dport 8888 -j ACCEPT



编辑完之后，保存退出，之后需要重新启动防火墙，重启的命令：service iptables restart，如下图所示。



    这样192.168.156.7这台设备的防火墙我们便配置好了，其它三台设备我们也都配置一下防火墙。

  配置过防火墙之后，我们再访问http://192.168.156.7:8888/group1/M00/00/00/wKicCFjj1xqAcN8EAAHk-VzqZ6w619.jpg，这时我们便可以看到图片了，如下图所示：

