Nginx实例

1. **概念**

Nginx——engine X，是一款自由的、开源的、高性能HTTP服务器和反向代理服务器；也是一个IMAP、POP3、SMTP代理服务器；也就是说Nginx本身就可以托管网站（类似于Tomcat一样），进行Http服务处理，也可以作为反向代理服务器使用。

Nginx 解决了服务器的C10K（就是在一秒之内连接客户端的数目为10k即1万）问题。它的设计不像传统的服务器那样使用线程处理请求，而是一个更加高级的机制—事件驱动机制，是一种异步事件驱动结构。

虽然目前Nginx的份额在市场上只占很少部分，但是其高性能和低消耗内存的结构，使得其越来越普遍，典型的一个应用就是我们可以使用Nginx作为反向代理进行网站的负载均衡器。例如：Wordpress、GitHub等知名的网站都使用到了Nginx。

1. **正向代理和反向代理**

首先，代理服务器一般指局域网内部的机器通过代理服务器发送请求到互联网上的服务器，代理服务器一般作用在客户端。例如：GoAgent翻墙软件。我们的客户端在进行翻墙操作的时候，我们使用的正是正向代理，通过正向代理的方式，在我们的客户端运行一个软件，将我们的HTTP请求转发到其他不同的服务器端，实现请求的分发。

反向代理服务器作用在服务器端，它在服务器端接收客户端的请求，然后将请求分发给具体的服务器进行处理，然后再将服务器的相应结果反馈给客户端。Nginx就是一个反向代理服务器软件。

1. **Nginx的特点**
2. 跨平台：可以在大多数Unix like 系统编译运行。而且也有Windows的移植版本。
3. 配置异常简单：非常的简单，易上手。
4. 非阻塞、高并发连接：数据复制时，磁盘I/O的第一阶段是非阻塞的。官方测试能支持5万并发连接，实际生产中能跑2~3万并发连接数（得益于Nginx采用了最新的epoll事件处理模型（消息队列）。
5. Nginx代理和后端Web服务器间无需长连接；
6. Nginx接收用户请求是异步的，即先将用户请求全部接收下来，再一次性发送到后端Web服务器，极大减轻后端Web服务器的压力。
7. 发送响应报文时，是边接收来自后端Web服务器的数据，边发送给客户端。
8. 网络依赖性低，理论上只要能够ping通就可以实施负载均衡，而且可以有效区分内网、外网流量。
9. 支持内置服务器检测。Nginx能够根据应用服务器处理页面返回的状态码、超时信息等检测服务器是否出现故障，并及时返回错误的请求重新提交到其它节点上。
10. 采用Master/worker多进程工作模式。
11. 此外还有内存消耗小、成本低廉（比F5硬件负载均衡器廉价太多）、节省带宽、稳定性高等特点。
12. **实例**
13. 环境准备：
14. 一台Nginx服务器A，IP为123.56.219.140
15. 一台部署了Web服务应用的Centos服务器B，IP为114.115.202.41，tomcat服务端口8080，tomcat版本8.5.23
16. 一台部署了同样Web服务应用的Centos服务器C，IP为114.115.159.46，tomcat服务端口8081，tomcat版本8.0.45
17. Nginx安装

直接进行yum安装

yum install nginx.x86\_64

1. 配置Nginx反向代理

在http中加入

|  |
| --- |
| upstream local\_tomcat {  server 114.115.202.41:8080 weight=1;  server 114.115.159.46:8081 weight=5;  } |

我们在server外添加了一个upstream，而直接在proxy\_pass里面直接用http://+upstream的名称来使用。

upstream中的server元素必须要注意，不能加http://，但proxy\_pass中必须加。

另外我们需要做到消除单点故障，并且希望一个服务器访问的机会比另外一个大，这个可以在server最后加上一个weight=数字来指定，数字越大，表明请求到的机会越大。

在http的server部分中加入（修改）

|  |
| --- |
| listen 8080 default\_server;  server\_name \_;  root /usr/share/nginx/html;  # index index.html index.htm;  # Load configuration files for the default server block.  include /etc/nginx/default.d/\*.conf;  location / {  proxy\_pass <http://local_tomcat>;  } |

listen：表示当前的代理服务器监听的端口，默认的是监听80端口。注意，如果我们配置了多个server，这个listen要配置不一样，不然就不能确定转到哪里去了。

server\_name：表示监听到之后需要转到哪里去，这时我们直接转到本地，这时是直接到nginx文件夹内。

location：表示匹配的路径，这时配置了/表示所有请求都被匹配到这里

root：里面配置了root这时表示当匹配这个请求的路径时，将会在这个文件夹内寻找相应的文件，这里对我们之后的静态文件伺服很有用。

index：当没有指定主页时，默认会选择这个指定的文件，它可以有多个，并按顺序来加载，如果第一个不存在，则找第二个，依此类推。

下面的error\_page是代表错误的页面，这里我们暂时不用，先不管它。

修改后重新加载

nginx -s reload

或者重启服务

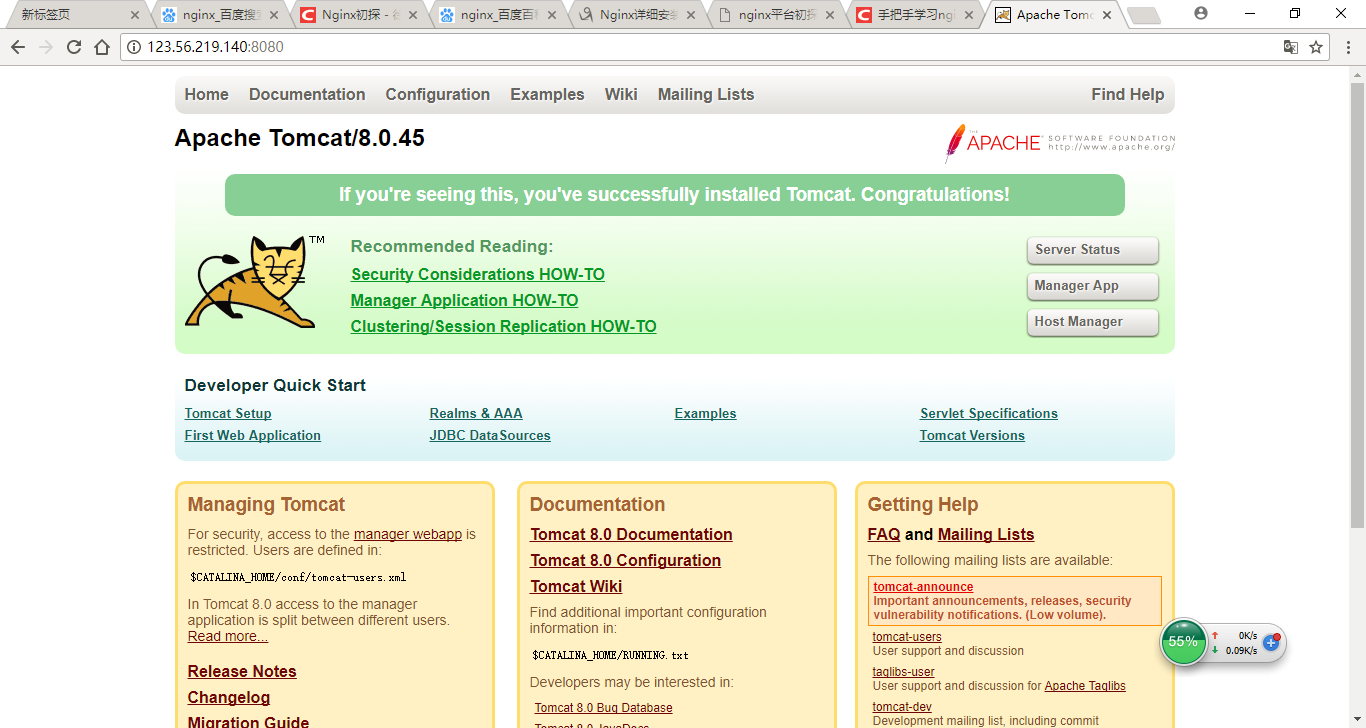
systemctl restart nginx.service

如果不想直接加载，而只是想看看自己的配置文件有没有问题，可以直接输入:

nginx -t

1. 进行验证

第一次访问



第N次访问

