# 前提

学习NgInx服务器要会：原理、反向代理操作，负载均衡的操作

NgInx服务器：Nginx是一款高性能的http 服务器/反向代理服务器及电子邮件（IMAP/POP3）代理服务器

反向代理：两个域名指向同一台nginx服务器，用户访问不同的域名显示不同的网页内容

其实就是通过nginx服务器默认端口80指向另一个服务器(tomcat)的操作

负载均衡：如果一个服务由多条服务器提供，需要把负载分配到不同的服务器处理

通俗就是：集群的服务器，让它平均的访问操作

# 反向代理(重要)步骤

正向代理：用户角度：指的的是用户本身电脑连接到有网的然后返还结果给你

反向代理: 服务器角度：代理用户访问的地址，让此地址拿来访问哪一个服务器提供服务

两个域名指向同一台nginx服务器，用户访问不同的域名显示不同的网页内容

第一步：安装两个tomcat，分别运行在8080和8081端口。

第二步：启动两个tomcat。

第三步：反向代理服务器的配置

用root用户打开：/usr/local/nginx/conf/nginx.conf

upstream tomcat1 {

server 192.168.25.148:8080; #指定tomcat服务器的ip地址和端口号

}

server {

listen 80; #nginx的端口号

server\_name www.wwl1.com; #自己配置的域名

#charset koi8-r;

#access\_log logs/host.access.log main;

location / {

proxy\_pass http://tomcat1; #指向 上面红色配好的地址和端口号

index index.html index.htm; # 要访问的网址

}

}

upstream tomcat2 {

server 192.168.25.148:8081;

}

server {

listen 80;

server\_name www.wwl2.com;

#charset koi8-r;

#access\_log logs/host.access.log main;

location / {

proxy\_pass http://tomcat2;

index index.html index.htm;

}

}

2、刷新配置文件：进入 /usr/local/nginx/sbin 目录

命令： ./nginx -s reload

第四步 修改window的hosts文件：（C:\Windows\System32\drivers\etc）

192.168.0.104 www.wwl2.com

192.168.0.104 www.wwl1.com

第五步 访问操作

http://www.wwl2.com/ 就会访问tomcat2的主页 index.jsp

http://www.wwl1.com/ 就会访问tomcat1的主页 index.jsp

这5步就是反向代理访问不同的tomcat服务的操作

# 负载均衡(重要)

根据上面的步骤

如果一个服务由多条服务器提供，需要把负载分配到不同的服务器处理，需要负载均衡。

下面的配置就是在访问tomcat2时（平均访问）：访问8081服务器后访问8082服务器

upstream tomcat2 {

server 192.168.25.148:8081;

server 192.168.25.148:8082;

}

可以根据服务器的实际情况调整服务器权重。权重越高分配的请求越多，权重越低，请求越少。默认是都是1，

upstream tomcat2 {

server 192.168.25.148:8081;

server 192.168.25.148:8082 weight=2;#这写法就是让此服务器不是1比1形式访问

}

# Nginx的高可用

了解：老师说一般都是运维操作的

就是 解决了tomcat服务器的压力和挂掉的问题，谁来解决我（Nginx）挂掉呢，我一旦挂掉就都不行了

nginx作为负载均衡器，所有请求都到了nginx，可见nginx处于非常重点的位置，如果nginx服务器宕机后端web服务将无法提供服务，影响严重。

为了屏蔽负载均衡服务器的宕机，需要建立一个备份机。主服务器和备份机上都运行高可用（High Availability）监控程序，通过传送诸如“I am alive”这样的信息来监控对方的运行状况。当备份机不能在一定的时间内收到这样的信息时，它就接管主服务器的服务IP并继续提供负载均衡服务；当备份管理器又从主管理器收到“I am alive”这样的信息时，它就释放服务IP地址，这样的主服务器就开始再次提供负载均衡服务。

使用到的技术 ：keepalived+nginx实现主备

什么是keepalived 通俗就是：两台装了此软件备用机连续访问主用机”你死了没有”死了就自己顶上，主用机复活了自动又抢回nginx的服务

keepalived是集群管理中保证集群高可用的一个服务软件，用来防止单点故障。

Keepalived的作用是检测web服务器的状态，如果有一台web服务器死机，或工作出现故障，Keepalived将检测到，并将有故障的web服务器从系统中剔除，当web服务器工作正常后Keepalived自动将web服务器加入到服务器群中，这些工作全部自动完成，不需要人工干涉，需要人工做的只是修复故障的web服务器。

高可用环境

两台nginx，一主一备：192.168.101.3和192.168.101.4

两台tomcat服务器：192.168.101.5、192.168.101.6

# nginx

什么是nginx

Nginx是一款高性能的http 服务器/反向代理服务器及电子邮件（IMAP/POP3）代理服务器。由俄罗斯的程序设计师Igor Sysoev所开发，官方测试nginx能够支支撑5万并发链接，并且cpu、内存等资源消耗却非常低，运行非常稳定。

应用场景

1、http服务器：Nginx是一个http服务可以独立提供http服务。可以做网页静态服务器。

2、虚拟主机：可以实现在一台服务器虚拟出多个网站。例如个人网站使用的虚拟主机。

3、反向代理,负载均衡：当网站的访问量达到一定程度后，单台服务器不能满足用户的请求时，需要用多台服务器集群可以使用nginx做反向代理。并且多台服务器可以平均分担负载，不会因为某台服务器负载高宕机而某台服务器闲置的情况。

## nginx安装

下载nginx：

官方网站：

http://nginx.org/

使用的版本是1.8.0版本。

Nginx只提供的源码。因为每个linux很多版本安装方式都不一样

要求的安装环境

1、需要安装gcc的环境。命令 yum install gcc-c++，ubuntu：apt-get install gcc

2、第三方的开发包PCRE、zlib、openssl

PCRE(Perl Compatible Regular Expressions)是一个Perl库，包括 perl 兼容的正则表达式库。nginx的http模块使用pcre来解析正则表达式，所以需要在linux上安装pcre库。

命令 yum install -y pcre pcre-devel

ubuntu系统命令:sudo apt-get install libpcre3 libpcre3-dev

注：pcre-devel是使用pcre开发的一个二次开发库。nginx也需要此库。

zlib库提供了很多种压缩和解压缩的方式，nginx使用zlib对http包的内容进行gzip，所以需要在linux上安装zlib库。

命令 yum install -y zlib zlib-devel

ubuntu系统命令: sudo apt-get install zlib1g-dev

OpenSSL 是一个强大的安全套接字层密码库，囊括主要的密码算法、常用的密钥和证书封装管理功能及SSL协议，并提供丰富的应用程序供测试或其它目的使用。

nginx不仅支持http协议，还支持https（即在ssl协议上传输http），所以需要在linux安装openssl库。

yum install -y openssl openssl-devel

ubuntu系统命令：apt-get install openssl

Nginx安装步骤

第一步：把nginx的源码包上传到linux系统

第二步：解压缩

[root@localhost ~]# tar zxf nginx-1.8.0.tar.gz

第三步：解压完进入根目录执行./configure命令创建一makeFile文件。\代表换行

./configure \

--prefix=/usr/local/nginx \

--pid-path=/var/run/nginx/nginx.pid \

--lock-path=/var/lock/nginx.lock \

--error-log-path=/var/log/nginx/error.log \

--http-log-path=/var/log/nginx/access.log \

--with-http\_gzip\_static\_module \

--http-client-body-temp-path=/var/temp/nginx/client \

--http-proxy-temp-path=/var/temp/nginx/proxy \

--http-fastcgi-temp-path=/var/temp/nginx/fastcgi \

--http-uwsgi-temp-path=/var/temp/nginx/uwsgi \

--http-scgi-temp-path=/var/temp/nginx/scgi

注意：启动nginx之前，上边将临时文件目录指定为/var/temp/nginx，需要在/var下创建temp及nginx目录

[root@localhost sbin]# mkdir /var/temp/nginx/client -p

第四步：执行命令：make

第五步：执行命令：make install

## nginx操作

启动nginx

进入 /usr/local/nginx/sbin 目录

[root@localhost sbin]# ./nginx

关闭nginx：

[root@localhost sbin]# ./nginx -s stop

推荐使用：

[root@localhost sbin]# ./nginx -s quit

重启nginx：

1、先关闭后启动。

2、刷新配置文件：

[root@localhost sbin]# ./nginx -s reload

访问nginx

默认是80端口。

注意：linux是否关闭防火墙。

直接在另一台主机浏览器中访问：http://192.168.0.104/ 即可



## 配置虚拟主机

要明白的就是nginx就是一台服务器：启动多个网站。

如何区分不同的网站：

1、域名不同，2、端口不同

演示通过端口区分不同虚拟机

第一步 Nginx的配置文件：进行配置（就跟配置tomcat差不多，配置端口）

路径文件：/usr/local/nginx/conf/nginx.conf

# '$status $body\_bytes\_sent "$http\_referer" '

# '"$http\_user\_agent" "$http\_x\_forwarded\_for"';

#access\_log logs/access.log main;

sendfile on;

#tcp\_nopush on;

#keepalive\_timeout 0;

keepalive\_timeout 65;

#gzip on;

#从这里开始配置------配置第一个端口号80、要访问的目录、缺省文件名

server {

listen 80; #这个是端口号，tomcat不能同时配置两个端口号

server\_name localhost; #这个是ip地址

#charset koi8-r;

#access\_log logs/host.access.log main;

location / {

root html;

index index.html index.htm;

}

}

#配置第二个端口号81、要访问的目录、缺省文件名

server {

listen 81;

server\_name localhost;

#charset koi8-r;

#access\_log logs/host.access.log main;

location / {

root html18; #要访问的静态资源目录

index index.html index.htm; #和tomcat一样默认要进入的资源文件名

}

}

}

第二步 重新加载配置文件：sbin/nginx -s reload

第三步 为了区分在nginx目录中再新建一个html文件夹命名为html18

访问操作

http://192.168.0.104/ 访问的是默认配置的80端口进入/usr/local/nginx目录下的html文件夹的index.html

http://192.168.0.104:81/ 你上面配置的第二个端口默认进入的是html18文件夹的index.html

这三步就是配置不同的虚拟机端口操作

演示通过域名区分虚拟主机

域名就是网站

Dns服务器：把域名解析为ip地址。保存的就是域名和ip的映射关系。

一级域名：Taobao.com二级域名 www.baidu.com三级域名Aaa.image.baidu.com

一个域名对应一个ip地址，一个ip地址可以被多个域名绑定。

第一步 修改window的hosts文件：（C:\Windows\System32\drivers\etc）

可以配置域名和ip的映射关系，如果hosts文件中配置了域名和ip的对应关系，不需要走dns服务器。

第二步 Nginx的配置和上面端口配置的差不多，不同点：端口号不变还是80

server {

listen 80; #配置两个虚拟机端口号不变

server\_name www.taobao.com; #这里配置域名一

#charset koi8-r;

#access\_log logs/host.access.log main;

location / {

root html-taobao; #/usr/local/nginx目录下的文件夹名字

index index.html index.htm;

}

}

server {

listen 80;

server\_name www.baidu.com; #这里配置域名二

#charset koi8-r;

#access\_log logs/host.access.log main;

location / {

root html-baidu;

index index.html index.htm;

}

}

}