**Nginx**

1. **特点**

主流的HTTP反向代理服务器（其企业版提供了基于TCP层的反向代理插件），对于构建大型分布式web应用，具有举足轻重的作用，具有内存少，并发能力强的特点，Nginx 可以作为静态页面的 web 服务器，同时还支持 CGI 协议的动态语言，比如 perl、php等。但是不支持 java。Java 程序只能通过与 tomcat 配合完成。Nginx 专为性能优化而开发，性能是其最重要的考量,实现上非常注重效率 ，能经受高负载的考验,有报告表明能支持高达 50,000 个并发连接数。。简单来说，Nginx有2个主要的功能：动/静态资源分离、负载均衡。

为了避免资源竞争，应该避免nginx和web server部署在同一个节点上，因为web server通常为CPU和内存高耗型，这会大大降低nginx的代理能力。

1. **安装**
2. **常用命令**

进入 nginx 目录中

cd /usr/local/nginx/sbin

1 、查看 nginx 版本号

./nginx -v

2 、启动 nginx

./nginx

3 、停止 nginx

./nginx -s stop

4 、重新加载 nginx

./nginx -s reload

1. **配置文件**

/usr/local/nginx/conf/nginx.conf

#user administrator administrators; #配置用户或者组，默认为nobody nobody。

#worker\_processes 2; #允许生成的进程数，默认为1

#pid /nginx/pid/nginx.pid; #指定nginx进程运行文件存放地址

error\_log log/error.log debug; #制定日志路径，级别。这个设置可以放入全局块，http块，server块，级别以此为：debug|info|notice|warn|error|crit|alert|emerg

events {

accept\_mutex on; #设置网路连接序列化，防止惊群现象发生，默认为on

multi\_accept on; #设置一个进程是否同时接受多个网络连接，默认为off

#use epoll; #事件驱动模型，select|poll|kqueue|epoll|resig|/dev/poll|eventport

worker\_connections 1024; #最大连接数，默认为512

}

http {

include mime.types; #文件扩展名与文件类型映射表

default\_type application/octet-stream; #默认文件类型，默认为text/plain

#access\_log off; #取消服务日志

log\_format myFormat '$remote\_addr–$remote\_user [$time\_local] $request $status $body\_bytes\_sent $http\_referer $http\_user\_agent $http\_x\_forwarded\_for'; #自定义格式

access\_log log/access.log myFormat; #combined为日志格式的默认值

sendfile on; #允许sendfile方式传输文件，默认为off，可以在http块，server块，location块。

sendfile\_max\_chunk 100k; #每个进程每次调用传输数量不能大于设定的值，默认为0，即不设上限。

keepalive\_timeout 65; #连接超时时间，默认为75s，可以在http，server，location块。

upstream mysvr {

server 127.0.0.1:7878;

server 192.168.10.121:3333 backup; #热备

}

error\_page 404 https://www.baidu.com; #错误页

server {

keepalive\_requests 120; #单连接请求上限次数。

listen 4545; #监听端口

server\_name 127.0.0.1; #监听地址

location ~\*^.+$ { #请求的url过滤，正则匹配，~为区分大小写，~\*为不区分大小写。

#root path; #根目录

#index vv.txt; #设置默认页

proxy\_pass http://mysvr; #请求转向mysvr 定义的服务器列表

deny 127.0.0.1; #拒绝的ip

allow 172.18.5.54; #允许的ip

}

}

}

配置文件中的内容

包含三部分内容

**1、全局块**：配置影响nginx全局的指令。一般有运行nginx服务器的用户组，nginx进程pid存放路径，日志存放路径，配置文件引入，允许生成worker process数等。

**2、events块** ：配置影响nginx服务器或与用户的网络连接。有每个进程的最大连接数，选取哪种事件驱动模型处理连接请求，是否允许同时接受多个网路连接，开启多个网络连接序列化等。

**3、http块**

还包含两部分：

http全局块：

server

1. **反向代理**

反向代理，其实客户端对代理是无感知的，因为客户端不需要任何配置就可以访问，我们只需要将请求发送到反向代理服务器，由反向代理服务器去选择目标服务器获取数据后，在返回给客户端，此时反向代理服务器和目标服务器对外就是一个服务器，暴露的是代理服务器地址，隐藏了真实服务器 IP 地址。

1. **正向代理**

Nginx 不仅可以做反向代理，实现负载均衡。还能用作正向代理来进行上网等功能。正向代理：如果把局域网外的 Internet 想象成一个巨大的资源库，则局域网中的客户端要访问 Internet，则需要通过代理服务器来访问，这种代理服务就称为正向代理。

1. **配置方式**
2. **负载均衡**

对于动态资源而言，如果有多个proxy server，那么nginx将会根据一定的算法选择合适的Server，并转发请求，最终将客户端request相对均衡的分发给多个Server。

1. **配置方式**

upstream mysvr {

server 192.168.10.121:3333;

server 192.168.10.122:3333;

}

server {

....

location ~\*^.+$ {

proxy\_pass http://mysvr; #请求转向mysvr 定义的服务器列表

}

}

1. **分配服务器策略**

1、轮询（默认）

每个请求按时间顺序逐一分配到不同的后端服务器，如果后端服务器down掉，能自动剔除。

2、weight

weight 代表权重默认为1, 权重越高被分配的客户端越多

3、ip\_hash

每个请求按访问 ip 的 的 hash 结果分配，这样每个访客固定访问一个后端服务器

4、fair （第三方）

按后端服务器的响应时间来分配请求，响应时间短的优先分配。

1. **动静分离**

Nginx支持正则表达式以区分静态资源或者动态资源，其中动态资源可以进一步转发给后端的proxy server，而静态资源则可以在Nginx层面使用本地缓存策略或者重定向（类CDN）到其他Nginx上。

1. **配置方式**
2. **分类**

1、把静态文件独立成单独的域名，放在独立的服务器上，也是目前主流推崇的方案。

2、动态跟静态文件混合在一起发布，通过Nginx来分开。

1. **高可用**
2. **Nginx配置主从模式**
3. **Nginx配置双主模式**