

Nginx 配置详情介绍

什么是 Nginx ?

Nginx 是一个高性能的 HTTP 和反向代理服务器，同时也是一个 IMAP/POP3/SMTP 代理服务器。

Nginx 与 Apache 相比：在性能上，占资源少，支持更多的并发连接；在功能上，Nginx 是优秀的代理服务器和负载均衡服务器；在安装配置上，Nginx 安装简单、配置灵活。

Nginx 的各种配置案例

Nginx 配置文件主要分为 4 部分：main（全局设置）、server（主机设置）、upstream（负载均衡服务器设置）和 location（URL 匹配特定位置的设置）。

分为全局变量和 http 模块，全局变量设置了 Nginx 服务的进程数和每个进程的最大连接数目。

一个基本的 Nginx 配置文件如下图：

```
1 #user  nobody;
2 worker_processes  1;
3
4 events {
5     #use  epoll;
6     worker_connections  1024;
7 }
8
9
10 http {
11     include      mime.types;
12     default_type  application/octet-stream;
13
14     sendfile      on;
15
16     keepalive_timeout  65;
17
18     server {
19         listen      80;
20         server_name  localhost;
21
22
23         location / {
24             root      html;
25             index      index.html index.htm;
26         }
27
28         error_page   500 502 503 504  /50x.html;
29         location = /50x.html {
30             root      html;
31         }
32     }
33 }
```

对 Nginx 的各种配置主要是针对 http 模块的添加修改。

1. 监听多个端口

通过 listen 方式， 添加多个监听端口

```
server {  
    listen      10.65.106.63:80;  
    listen      10.65.106.63:88;  
    server_name localhost;  
  
    location / {  
        root    html;  
        index   index.html index.htm;  
    }  
}
```

2. 多个 IP 映射到不同的站点目录

Nginx 内核将用户的请求映射到 location block， 设置 server 下的监听 IP 和站点目录， 就可以实现精准映射。

```
server {  
    listen      10.65.106.63:80;  
  
    location / {  
        root    /opt/nginx/html;  
        index   index.php index.html index.htm;  
    }  
  
    server {  
        listen 10.65.102.9:80;  
        server_name 10.65.102.9;  
        location / {  
            root /opt/nginx/html2;  
            index index.html;  
        }  
    }  
}
```

3. 设置反向代理

Nginx 反向代理的指令不需要新增额外的模块，默认自带 proxy_pass 指令，只需要修改配置文件就可以实现反向代理。



```
server {  
    listen 10.65.106.63:800;  
    server_name search;  
    location /{  
        proxy_pass http://74.125.31.106;  
    }  
}
```

4. 配置负载均衡

Nginx 的负载均衡策略可以划分为两大类：内置策略和扩展策略。内置策略包含加权轮询和 ip hash，在默认情况下这两种策略会编译进 Nginx 内核，只需在 Nginx 配置中指明参数即可。扩展策略有很多，如 fair、通用 hash、consistent hash 等，默认不编译进 Nginx 内核。



- 默认轮询

访问 888 端口，会依次访问 server 10.65.102.9 和 server 10.65.106.67

```
upstream aa.com{  
    server 10.65.102.9;  
    server 10.65.106.67;  
}  
server {  
    listen 888;  
    server_name aa.com;  
    location / {  
        proxy_pass http://aa.com;  
    }  
}
```

- 设置权重

很多情况，服务器的性能并不完全相同，Nginx 通过设置权重，将客户请求按照一定的概率转送到后台服务器上。性能高的服务器接收到的概率大，合理利用了服务器集群的性能。

```
upstream aa.com{
    server 10.65.102.9 weight = 10;
    server 10.65.106.67 weight = 1;
}
server {
    listen 888;
    server_name aa.com;
    location / {
        proxy_pass http://aa.com;
    }
}
```

- Ip_hash

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

```
upstream aa.com{
    ip_hash;
    server 10.65.102.9;
    server 10.65.106.67;
}
server {
    listen 888;
    server_name aa.com;
    location / {
        proxy_pass http://aa.com;
    }
}
```

5. Fast-CGI 配置

Nginx 可以处理静态页面，将动态脚本交给 php, ruby 等语言来解析，配置过程如下：

```
location ~ \.php$ {
    root            html;
    fastcgi_pass    127.0.0.1:9000;
    fastcgi_index   index.php;
    fastcgi_param   SCRIPT_FILENAME /scripts$fastcgi_script_name;
    include         fastcgi_params;
}
```

6. HTTPS 配置

https 需要在本地生成证书，按照如下方式配置：

```
server {
    listen      443;

    ssl
    ssl_certificate      /etc/ssl/cert.crt;
    ssl_certificate_key  /etc/ssl/cert_nopass.key;

    ssl_protocols  SSLv2 SSLv3 TLSv1;
    ssl_ciphers   HIGH:!aNULL:!MD5;
    ssl_prefer_server_ciphers   on;

    location / {
        root    html;
        index   index.html index.htm;
    }
}
```

7. 虚拟主机设置

很多主机厂商，不愿让用户通过域名来访问服务器，Nginx 通过监听 http 请求包中内容，来将用户分转到不同的主机上。

```
server{
    listen 80;
    server_name fish.net;
    location /{
        index index.html;
        root fish.net;
    }
}
Server {
    listen 80 default;
    server_name fish.com;
    location /{
        index index.html;
        root fish.com;
    }
}
```