## Nginx是一款什么软件

Nginx 是⼀个⾼性能的HTTP和反向代理web服务器，核⼼特点是占有内存少，并发能⼒强。它通常的用途有：

1. Http服务器（Web服务器）

性能⾮常⾼，⾮常注重效率，能够经受⾼负载的考验。 ⽀持50000个并发连接数，不仅如此，CPU和内存的占⽤也⾮常的低，10000个没有活动的连 接才占⽤2.5M的内存。

1. 反向代理服务器

正向代理指的是在浏览器中配置代理服务器的相关信息，通过代理服务器访问⽬标⽹站，代理服务器收 到⽬标⽹站的响应之后，会把响应信息返回给我们⾃⼰的浏览器客户端 。

反向代理 指的是浏览器客户端发送请求到反向代理服务器（⽐如Nginx），由反向代理服务器选择原始 服务器提供服务获取结果响应，最终再返回给客户端浏览器。

1. 负载均衡服务器

负载均衡，当⼀个请求到来的时候，Nginx反向代理服务器根据请求去找到⼀个 原始服务器来处理当前请求，那么这叫做反向代理。那么，如果⽬标服务器有多台 ，找哪⼀个⽬标服务器来处理当前请求呢，这样⼀个寻找确定的过程就叫做负载均衡。 负载均衡就是为了解决⾼负载的问题。

1. 动静分离

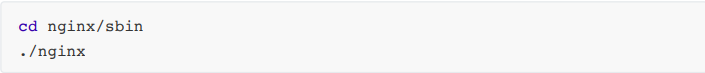
静态界面和动态资源都在服务器的同一个地方。Tomcat的静态文件处理性能不高，通过Nginx 将动态资源的请求交给tomcat处理，nginx 自己处理静态资源。提升服务性能。

## Nginx的安装 过程

1. 上传nginx安装包到linux服务器，nginx安装包(.tar⽂件)下载地址：<http://nginx.org> 。系统中使用的Nginx软件版本是nginx-1.18.0。
2. 安装Nginx依赖，pcre、openssl、gcc、zlib（推荐使⽤yum源⾃动安装）

执行命令 yum -y install gcc zlib zlib-devel pcre-devel openssl openssl-devel

1. 解压Nginx软件包 执行命令 tar -xvf nginx-1.17.8.tar 。
2. 进⼊解压之后的⽬录 nginx-1.18.0
3. 命令⾏执⾏ ./configure
4. 命令⾏执⾏ make
5. 命令⾏执⾏ make install
6. 完毕之后在/usr/local/下会产⽣⼀个nginx⽬录。进⼊sbin⽬录（/usr/local/nginx/sbin）中，执⾏启动nginx命令。



1. 访问服务器的80端⼝（nginx默认监听80端⼝）查看Nginx是否启动成功。



## Nginx常用命令

./nginx 启动nginx

./nginx -s stop 终⽌nginx（当然也可以找到nginx进程号，然后使⽤kill -9 杀掉nginx进程）

./nginx -s reload (重新加载nginx.conf配置⽂件)

## Nginx应⽤场景之反向代理

需求：在子服务器中，子服务器目前由素材服务和内容服务两部分组成。以Alpha环境为例，

素材服务部署在39.107.254.51机器上的8086端口，它暴露给外界的接口服务有一个固定的前缀（http://39.107.254.51:8086/material\_moudel/）。内容服务器部署在39.107.254.51机器上的8080端，它暴露给外界的接口服务有一个固定的前缀口http:// 39.107.254.51:8080/sserver/。 为了使服务前缀不混乱，使用Nginx做反向代理，客户端在请求子服务器的请求时，统一使用 http:// 39.107.254.51:7980 前缀，如果地址是以<http://39.107.254.51:7980/sserver/material_moudel/>开头，就由nginx 将该地址重定位到转到素材服务（http://39.107.254.51:8086/material\_moudel/），否则，判断是否以<http://39.107.254.51:7980/sserver>开头，如果是，就由nginx重定向到内容服务（http:// 39.107.254.51:8080/sserver/）。

如何实现： 首先需要修改Nginx的配置文件，将监听端口设置成7980，同时设置反向代理路由。然后重新启动nginx服务。

1. 找到文件（/usr/local/nginx/conf/nginx.conf），进入编辑是 vim [文件名]
2. 并将文件中的配置项设置为下图 ：



推出编辑 是先按Esc键，然后输入一个:[冒号] 若是保存编辑并退出 输入wq ,若直接退出为q!

2.设置成功后，重新启动nginx。先进入sbin目录后,启动命令为

./nginx -s reload

## Nginx应⽤场景之动静分离

动静分离就是讲动态资源和静态资源的请求处理分配到不同的服务器上，⽐较经典的组合就是 Nginx+Tomcat架构（Nginx处理静态资源请求，Tomcat处理动态资源请求），Nginx反向代理⽬标服务器Tomcat，这本身就是 Tomcat在处理动态资源请求了。 所以，我们只需要配置静态资源访问即可。具体配置如下图。

将静态资源映射到服务器目录使用了 命令 alias

