Ngnix(万物皆文件)

1. Ngnix在架构体系中的位置与作用

外网(考虑网络安全,跨域等)

浏览器

网路cdn

Ngnix负载

商品服务器

订单服务器

服务A

服务B

主

从

从

ES

1. 网关

---面向客户的总路口

1. 虚拟主机

--为不同域名/ip/端口提供服务

1. 路由

--使用反相代理，整合后面的服务为一个完整的业务

1. 静态服务器

--mvvm模式中，用来发布html/css/jd/ing

1. 负载集群

--使用upstream，负载多个tomcat

1. Ngnix的模块化设计
2. 核心模块

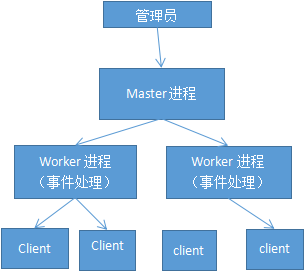
--正常运行必不可少的模块，提供错误日志记录，配置文件解析，事件驱动机制，进程管理等核心功能

1. 标准http模块

--提供http协议解析相关的功能，如端口配置，网页的编码设置，http响应头设置等

1. 可选的http模块
2. 邮件服务模块
3. 第三方模块
4. Ngnix的多进程模式(效率高，一个线程，独享操作资源)

进程=操作系统资源+线程(共享操作系统的资源，多线程，会造成资源抢占，造成线性安全问题，需要数据的原子性)



Master主要管理worker进程，包括接来自外界的信号，向各个worker进程发送信号，监控worker进程的运行状态，当worker进程退出后(异常情况下)，会自动重新启动新的worker进程。

发送信号的方式

Kill -QUIT 进程号 安全停止

Kill -TERM 进程号 立即停止

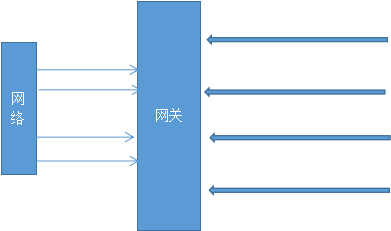
停止nginx

./nginx -s stop 停止

./nginx -s quit 退出

./ngnix -s reload 重新加载nginx.conf

1. Epoll--Nginx的异步非IO



Poll(以前的模式):被惊醒的线程轮询所有的http连接，定位触发事件的那个连接

1. poll模式:线程被惊醒后(省略轮询的工作linux内核把这个做了)，直接到map中定位触发事件的那个连接
2. 正向代理和方向代理

正向代理(代理的是需求方)

1. Nginx安装（源码安装，yum安装）

源码安装方便扩展

(1)解压tar包

(2)./configure --prefix=/usr/local/nginx --with-http\_stub\_status\_module --with-http\_ssl\_module

--prefix 安装指定的目录

--with-http\_stub\_status\_module 启用监控模块

--with-http\_ssl\_module 启用网络安全https模块

(3)make 编译

(4)make install编译安装

目录结构

conf ，html， logs，sbin

添加nginx的环境变量/etc/profile

source 环境变量生效

7，Ngnix.conf

main(全局配置)

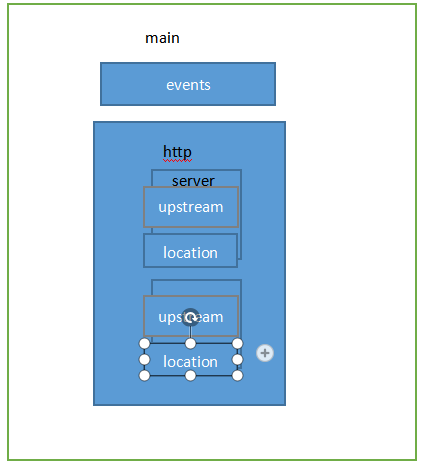
events:设置ngnix的工作模式及连接数的上限

http:服务器的相关属性

server:虚拟主机的配置

upstream:上游服务器的设置，主要为方向代理，负载均衡相关配置

location: url匹配特定位置后的设置



域名(ip)+端口=虚拟主机

nginx -t 查看配置文件的正确性

本地(host文件)

重要的配置:

8，server

日志:

error日志不可编辑: error\_log logs/error.log error;error\_log logs/error.log info

access日志可以编辑:access\_log access.log main(格式,可以自定义)，根据业务需要的不一样定义多个日志格式，不同的区域块(server,location)也可以定义不一样的日志格式

9，日志切割

编写自动分割的Nginx日志脚本，

nginx,不会自动创建日志文件，需要发送USR1信号，USR1信号是重新打开日志文件

USR1信号:自定义信号 --linux定义这个信号的解释权，归接收方自己来定义(nginx定义是重新打开日志文件)

(1,location语法，rewrite语法，2，rua脚本，做一些事情(redis

缓存值，)，直接返回给客户端，3，负载/动静分离/https/跨域)

虚拟主机--------ip/域名+端口

默认主机---指定那个默认的

1. 路由-Location

语法规则：

location[=|~|~\*|^~]/uri/{........}

最常用：

location=/

Location ^~/static

匹配顺序:精准匹配，普通匹配，正则匹配

匹配流程：  
1，有=精准匹配:直接走这个流程，返回

1. 普通匹配(非正则匹配):得到的location命中，最长的被命中(不满，先不返回)
2. 得到的普通匹配里最长的那个location，分情况，非正则的，直接走执行流程，返回
3. 最长的匹配不是非正则，走正则匹配，匹配正则的第一条

方向代理:

1. Rewrite的使用(内部重定向，外部重定向Rewrite[] flag)

Regex 正则表达式:匹配path成功，则往后执行

Replacement把路径跟换成Replacement字符串

flag 指的是重定向的方式:break/last内部重定向；redirect/permanent外部重定向(临时重定向302,永久重定向301)

Break:停止rewrite指令命令，替换path路径，流程不变往下走

Last:停止rewrite指令命令，替换path路径，重新location

没有flag：顺序执行

Nginx的11个阶段

Rewrite,find\_config,rewrite,access,content

代理proxy pass=ip:port,把整个path路径转发给tomcat

Root,整个path路径，拿去找文件

Proxy pass=ip:port/,只会匹配剩余的路径path，转发给tomcat

Alias，只会把剩余的路径，拿去找文件

12复杂均衡

Upstream [别名]{

#ip\_hash;

Server 192.168.0.128:8383 weight=3;

Server 192.168.244.233:8383 weight-=1;

}

Ip\_hash场景，简单的session登陆，防止用户重新登陆

Openresty

Lua脚本(ngnix可以直接对接redis，高并发性能好的工具，一般业务tomcat的性能多行程下比较差,可以直接跳过)