# (VIP课程)高性能反向代理服务器-Nginx 第三次课

授课老师：Mic

授课时间：2017年12月10号20:00

# 课程内容安排

1. 课程回顾
2. nginx结合现有项目完成基本配置
3. Nginx的进程模型
4. Nginx配置https请求
5. Nginx+keepalived实战

# 课程笔记

Nginx的课程回顾

1. 反向代理和正向代理
2. 关于web请求到服务端的整个过程的解析
3. Nginx的安装和常用命令
4. Nginx虚拟主机配置； 基于域名、基于端口
5. location的匹配规则
6. Rewrite的使用rewrite regex replacement [flag]
7. nginx的本地缓存和Gzip的压缩策略
8. nginx的负载均衡和负载均衡的调度算法

upstream ip\_hash / 轮询（默认） /fair /url\_hash /weight

## Nginx的实践演练

gupao-protal 首页

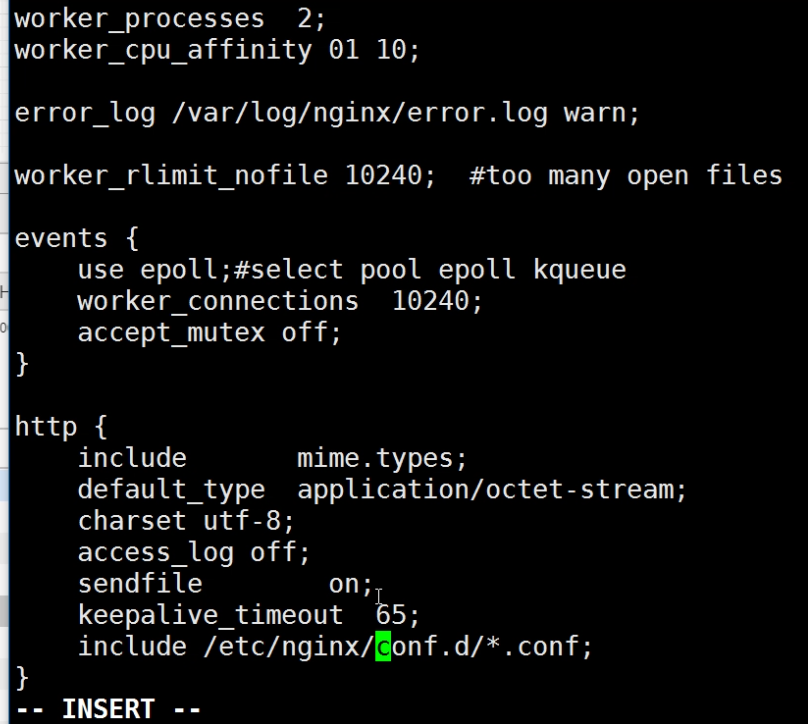
tomcat1 / tomcat2

nginx配置对应的文件 ; /etc/nginx/conf.d/\*.conf

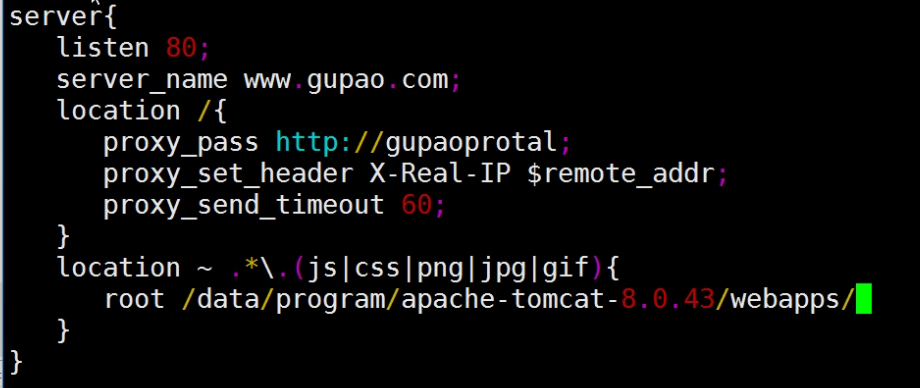
upstream.conf 用来配置负载均衡的服务

[www.gupao.com.conf](http://www.gupao.com.conf) 用来配置host信息

**配置信息请参考对应的文件，在git上**

****

**鲸群（鸡 一粒米 任务 线程池）**

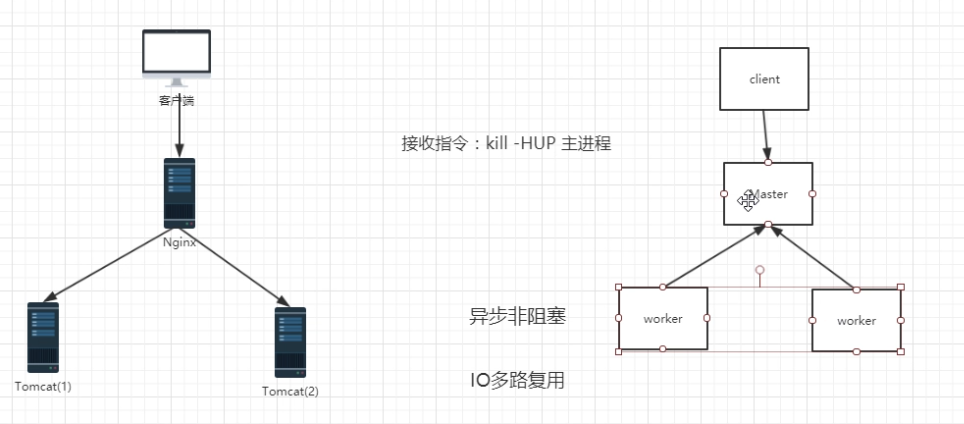
****

静态资源和和tomcat不在同一个服务器上 静态资源单独部署

## Nginx的进程模型

Nginix每个进程里面都是单个线程 利用了多路复用 ；nginx是多进程的

服务器快速响应客户端请求的工作方式：多进程，多线程，异步方式



### Master进程

充当整个进程组与用户的交互接口，同时对进程进行监护。它不需要处理网络事件，不负责业务的执行，只会通过管理work进程来实现重启服务、平滑升级、更换日志文件、配置文件实时生效等功能。

主要是用来管理worker进程

1. 接收来自外界的信号 （前面提到的 kill -HUP 信号等）

我们要控制nginx，只需要通过kill向master进程发送信号就行了。比如kill -HUP pid，则是告诉nginx，从容地重启nginx，我们一般用这个信号来重启nginx，或重新加载配置，因为是从容地重启，因此服务是不中断的。master进程在接收到HUP信号后是怎么做的呢？首先master进程在接到信号后，会先重新加载配置文件，然后再启动新的worker进程，并向所有老的worker进程发送信号，告诉他们可以光荣退休了。新的worker在启动后，就开始接收新的请求，而老的worker在收到来自master的信号后，就不再接收新的请求，并且在当前进程中的所有未处理完的请求处理完成后，再退出

1. 向各个worker进程发送信号
2. 监控worker进程的运行状态
3. 当worker进程退出后（异常情况下），会自动重新启动新的worker进程

### Work进程

主要是完成具体的任务逻辑。它的主要关注点是客户端和后端真实服务器之间的数据可读、可写等I/O交互事件

各个worker进程之间是对等且相互独立的，他们同等竞争来自客户端的请求，一个请求只可能在一个worker进程中处理，worker进程个数一般设置为cpu核数

master进程先建好需要listen的socket后，然后再fork出多个woker进程，这样每个work进程都可以去accept这个socket。当一个client连接到来时，所有accept的work进程都会受到通知，但只有一个进程可以accept成功，其它的则会accept失败

## Nginx配置https的请求

* https基于SSL/TLS这个协议；
* 非对称加密、对称加密、 hash算法
* crt的证书->返回给浏览器

### 创建证书

* 创建服务器私钥

openssl genrsa -des3 -out server.key 1024

* 创建签名请求的证书（csr）; csr核心内容是一个公钥

openssl req -new -key server.key -out server.csr

* 去除使用私钥是的口令验证

cp server.key server.key.org

openssl rsa -in server.key.org -out server.key

* 标记证书使用私钥和csr

openssl x509 -req -days 365 -in server.csr -signkey server.key -out server.crt

x509是一种证书格式

server.crt就是我们需要的证书

生成的证书要经过审核才可以，否则浏览器上显示未受信任

截图上面有个8080端口 www.gupao.com



tomcat也要配置

**nginx配置文件在git上的[www.gupao.com.conf](http://www.gupao.com.conf)文件**

### tomcat增加对https的支持

1.Connector 8080节点加入 redirectPort="443" proxyPort="443"

redirectPort ：当http请求有安全约束才会转到443端口使用ssl传输

## Nginx+keepalived

keepalived – >VRRP(虚拟路由器冗余协议)

VRRP全称 Virtual Router Redundancy Protocol，即 虚拟路由冗余协议。可以认为它是实现路由器高可用的容错协议，即将N台提供相同功能的路由器组成一个路由器组(Router Group)，这个组里面有一个master和多个backup，但在外界看来就像一台一样，构成虚拟路由器，拥有一个虚拟IP（vip，也就是路由器所在局域网内其他机器的默认路由），占有这个IP的master实际负责ARP相应和转发IP数据包，组中的其它路由器作为备份的角色处于待命状态。master会发组播消息，当backup在超时时间内收不到vrrp包时就认为master宕掉了，这时就需要根据VRRP的优先级来选举一个backup当master，保证路由器的高可用。

### 安装keepalived

1. Tar -zxvf keepalived.tar.gz
2. ./configure --prefix=/mic/data/program/keepalived --sysconf=/etc
3. 缺少依赖：yum install gcc ; yum install openssl-devel；yum -y install libnl libnl-devel
4. 编译安装 make && make install
5. cd到解压的包 /parker/data/program/keepalived-1.3.9
6. ln -s /mic/data/program/keepalived/sbin/keepalived /sbin --建立软链接
7. cp /mic/data/program/keepalived-1.3.9/keepalived/etc/init.d/keepalived /etc/init.d/
8. 添加到系统服务
   1. chkconfig --add keepalived
   2. chkconfig keepalived on
   3. Service keepalived start

### 基于keepalived+nginx的配置

**配置文件详见 keepalived.conf; gitlab**

# Q&A