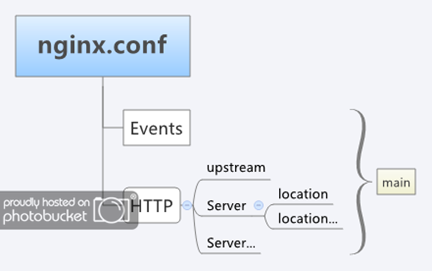
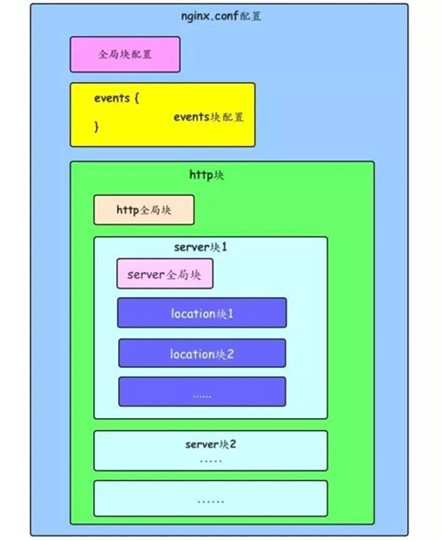
**一、Nginx的配置文件的整体结构**

Nginx是通过配置文件来做到各个功能的实现的。Nginx的配置文件的格式非常合乎逻辑，学习这种格式以及如何使用这种每个部分是基础，这将帮助我们有可能手工创建一个配置文件。

其整体结构如下：



思维导图



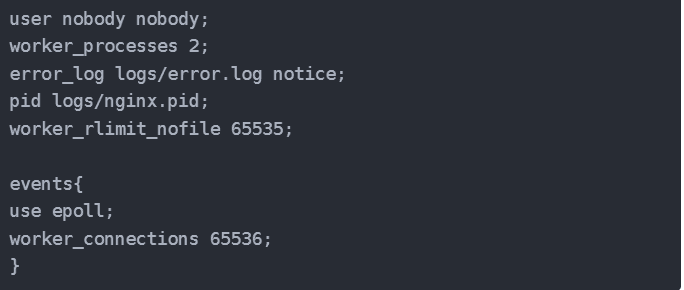
框架图

从图中我们可以看出主要包含以下几大部分内容：  
**main（全局设置）**：设置的指令将影响其他所有设置；  
**server（主机设置）**：指令主要用于指定主机和端口、  
**upstream（负载均衡服务器设置）**：指令主要用于负载均衡，设置一系列的后端服务器  
**location（URL匹配特定位置的设置）**：用于匹配网页位置。

这四者之间的关系式：server继承main，location继承server，upstream既不会继承其他设置也不会被继承。在这四个部分当中，每个部分都包含若干指令，这些指令主要包含Nginx的主模块指令、事件模块指令、HTTP核心模块指令，同时每个部分还可以使用其他HTTP模块指令，例如Http SSL模块、HttpGzip Static模块和Http Addition模块等。

**二、各部分详解**

**1. 全局块**



该部分配置主要影响Nginx全局，通常包括下面几个部分：

    a、配置运行Nginx服务器用户（组）  
     如：**user nobody nobody;**   user是个主模块指令，指定Nginx Worker进程运行以及用户组。  
    *指令格式：user user [group];    
     user：指定可以运行Nginx服务器的用户；group：可选项，可以运行Nginx服务器的用户组。  
     如果user指令不配置或者配置为user nobody nobody，默认由nobody账户运行。*

    b、worker process数      
      如：**worker\_processes 2;**    woker\_processes是个主模块指令，制定了Nginx要开启的进程数。每个Nginx进程平均耗费10M~12M内存。建议指定和CPU的数量一致即可。  
      Nginx服务器实现并发处理服务的关键。  
      *指令格式：worker\_processes number | auto;  
      number : Nginx 进程最多可以产生的worker process 数。  
      auto ： Nginx 进程将自动检测*  
      在按照上面的配置格式配置了之后，假如上面的数目是2，那么启动Nginx服务器后，在后台主机上查看Nginx的进程情况，可以看到应该是有2个Nginx进程。

1. [root@vm10-0-0-3 logs]# ps -ef|grep nginx
2. root     15624 13519  0 10:27 pts/1    00:00:00 grep --color=auto nginx
3. root     18465     1  0 Jun17 ?        00:00:00 nginx: master process ./nginx
4. root     19243 18465  0 Jun17 ?        00:00:00 nginx: worker process
5. root     19244 18465  0 Jun17 ?        00:00:00 nginx: worker process
6. root     19245 18465  0 Jun17 ?        00:00:00 nginx: worker process
7. root     19246 18465  0 Jun17 ?        00:00:00 nginx: worker process

    c、错误日志的存放路径  
      如：**error\_log logs/error.log  notice;**  error\_log 是个主模块指令，用来定义全局错误日志文件。日志输出级别有debug，info，notice，warn，error，erit可供选择，其中，debug输出日志最为详细，而crit输出日志最少。  
     *指定格式：error\_log file  | stderr;  
     file : 日志输出到某个文件file  
     stderr : 日志输出到标准错误输出 （日志输出级别）。*

    d、Nginx进程PID存放路径  
       如：**pid logs/nginx.pid;**  pid是个主模块指令，用来指定进程pid的存储文件位置。  
       Nginx进程是作为系统守护进程在进行，需要在某个文件中保存当前运行程序的主进程号，Nginx支持该保存文件路径的定义。  
       *指令格式：pid file;  
       file：指定存放路径和文件名称。*  
        如果不指定，则默认置于路径 logs/nginx.pid

[root@vm10-0-0-3 logs]# ll

total 204

-rw-r--r-- 1 root root 53873 Jun 18 09:58 access.log

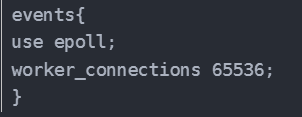
-rw-r--r-- 1 root root 138319 Jun 17 23:22 error.log

-rw-r--r-- 1 root root 6 Jun 17 22:25 nginx.pid

    e、worker\_rlimit\_nofile   
       如：**worker\_rlimit\_nofile 65535;** 用来绑定worker进程和CPU，Linux内核2.4 以上可用  
       指定进程可以打开的最多文件描述数目，理论值应该是最多打开文件数（ulimit -n ）与nginx进程数相除，但是Nginx分配请求并不是那么均匀，所以最好与ulimit -n 的值保持一致。  
       现在在Linux2.6 内核下开启文件打开数为65535，worker\_rlimit\_nofile 就相应应该填写65535.这是因为Nginx调度时请求到进程并不是那么均衡，所以假如填写10240，总并发量达到3-4万时就有进程超过10240了，这就会返回502。

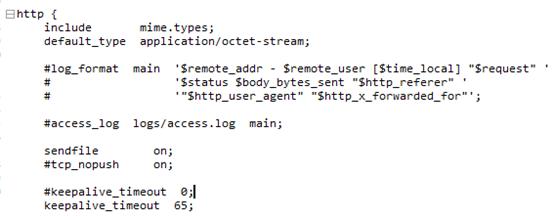
 events 事件指令是设定Nginx的工作模式及连接数上限

**2:events块**



 events事件指令是设定Nginx的工作模式及连接数上限。每个配置选项的含义解释如下：  
*a、use  
如：use epoll;   
use是事件模块指令，用来指定Nginx的工作模式。Nginx支持的工作模式有select、poll、kqueue、epoll、rtsig和/dev/poll 。其中select 和poll 都是标准的工作模式，kqueue和epoll是高效的工作模式，不同的是epoll用在Linux平台上，而kqueue用在BSD系统中。对于Linux系统，epoll工作模式是首选。*  
  
*b、worker\_connections  
如：worker\_connections65536;*  
*work\_connections也是个事件模块指令，用于定义Nginx每个进程的最大连接数，默认是1024。*

**3：http块**



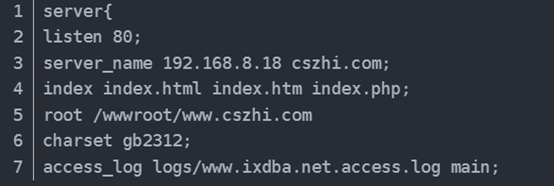
A、定义MIMI-Type  
如：include  mime.types;  
*指令格式:include file;*  
*该指令主要用于将其他的Nginx配置或第三方模块的配置引用到当前的主配文件中，减少主配置文件的复杂度。default\_type  application/octet-stream;  
default\_type*  
属于HTTP核心模块指令，这里设定默认类型为二进制流。也就是当文件类型未定义时使用这种方式，

B、自定义服务日志  
 log\_format 是Nginx的HttpLog模块指令，用于指定Nginx日志的输出日志。当然其中还有更为详细的配置，但是这里只做大略的研究学习，不过分深入细节。

C、允许sendfile方式传输文件  
*指令格式：sendfile  on;  
这个指令中的参数on是表示开启高效文件传输模式，默认是关闭状态（off），将tcp\_nopush和tcp\_nodelay两个指令设置为on用于防止网络阻塞；*

D、连接超时时间  
*指令格式：keepalive\_timeout number;  
如：keepalive\_timeout 65;  
  keepalive\_timeout  
 设置客户端连接保持活动的超时时间。在超过这个时间之后，服务器会关闭该连接；*

**4、server块**



server 块是对虚拟主机的配置，server标志定义虚拟主机开始，

A、listen用于指定虚拟主机的服务端口，  
B、server\_name 用来指定IP地址或域名，多个域名之间用空格分开。  
C、index用于设定访问的默认首页地址.。  
D、root指令用于指定虚拟主机的网页根目录，这个目录可以是相对路径，也可以是绝对路径。  
E、charset用于设置网页的默认编码格式。  
F、access\_log 用来指定虚拟主机的访问日志存放路径，最后的main 用于指定访问日志的输出格式。

**5.location块**

URL地址匹配是进行Nginx配置中最灵活的部分。 location支持正则表达式匹配，也支持条件判断匹配，用户可以通过location指令实现Nginx对动、静态网页进行过滤处理。使用location URL匹配配置还可以实现反向代理，用于实现PHP动态解析或者负载负载均衡。



a、location配置  
b、请求根目录配置  
c、更改location的URI  
d、网站默认首页配置