<http://www.nginx.cn/doc/>

关闭nginx用pkill -9 nginx

开启nginx用nginx

主模块

（1）error\_log

（2）include

（3）pid是port id

（4）user是指定执行nginx工作进程的用户

（5）worker\_processes是工作进程

（6）worker\_cpu\_affinity是几个工作进程对应几核cpu

事件模块

（1）accept\_mutex是惊群问题

（2）multi\_accept是告诉nginx收到一个新连接通知后接受尽可能多的连接

（3）worker\_connections是

nginx作为http服务器的时候：

max\_clients = worker\_processes \* worker\_connections

nginx作为反向代理服务器的时候：

max\_clients = worker\_processes \* worker\_connections/4

http核心模块

（1）alias是

location /a/ {

alias /b/;

}

若请求/a/a.html将返回/b/a.html

（2）client\_body\_in\_file\_only是请求体被存储到一个文件中，请求结束后，该文件不会被删除，一般调试时用

（3）client\_body\_buffer\_size是请求体使用的buffer的最大值

（4）client\_body\_temp\_path是存储请求体的文件的目录路径

（5）client\_body\_timeout是请求体的超时时间

（6）client\_header\_buffer\_size是请求头的buffer大小

（7）large\_client\_header\_buffers是当请求头过大，请求头的buffer大小

（8）client\_header\_timeout是请求头的超时时间

（9）client\_max\_body\_size是能接收的最大请求体的大小

（10）default\_type是设置文件使用的默认的MIME-type

（11）error\_page是发生错误的时候显示一个预定义的uri

（12）if\_modified\_since是定义如何比较修改时间和请求头中"If-Modified-Since"时间

（13）index是首页文件名称

（14）internal是指定某个location只能被内部的请求调用，外部的调用请求会返回404

（15）keepAliveTimeout是关闭连接之前，连接器等待另一个HTTP请求的毫秒数

（16）keepalive\_requests是在长连接期间可接受的请求数

（17）limit\_except是限制http方法

（18）limit\_rate是设置最高下载速度

（19）limit\_rate\_after是设置http请求传输多少字节后开始限速

（20）listen

listen 80是监听所有端口为80的ipv4的地址

listen [::]:80是监听所有端口为80的ipv6的地址

（21）location

匹配模式及顺序

location = /uri 　　　 精确匹配

location ^~ /uri 　　 前缀匹配，顺序是从长到短

location ~ pattern 区分大小写的正则匹配，顺序是从上到下

location ~\* pattern 　不区分大小写的正则匹配，顺序是从上到下

location /uri 　　　　 前缀匹配，顺序是从长到短

location / 通用匹配

（22）open\_file\_cache max=65535 inactive=30s

最多缓存多少个文件，缓存多少时间

open\_file\_cache\_min\_uses 1

在30S中没有使用到这个配置的次数的话就删除

open\_file\_cache\_valid 40s

多少时间检查一次，如果发现30s内没有用过一次的删除

（23）server\_name\_in\_redirect

当URL 指向一个目录并且在最后没有包含“/”时，Nginx 内部会自动的做一个 301 重定向，这时会有两种情况：

1、server\_name\_in\_redirect on（默认），URL重定向为：server\_name中的第一个域名 + 目录名 + /

2、server\_name\_in\_redirect off，URL重定向为：原URL中的域名 + 目录名 + /

（24）port\_in\_redirect是nginx在redirect的时候不要带上port

（25）recursive\_error\_pages是否允许递归使用error\_page

（26）root是

location /a/ {

root /b/;

}

若请求/a/a.html将返回/b/a/a.html

（27）satisfy

如果在一个字段中同时使用了Access模块和Auth Basic模块的指令，可以使用这个指令确定一种验证方式：

all - 必须同时匹配Access和Auth Basic中指令指定的权限。

any - 具有Access 或 Auth Basic指令任一权限即可通过匹配

（28）sendfile是设置为on表示启动高效传输文件的模式。若设置为off，会先在用户空间（Nginx进程空间）申请一个buffer，用read函数把数据从磁盘读到cache，再从cache读取到用户空间的buffer，再用write函数把数据从用户空间的buffer写入到内核的buffer，最后到tcp socket。若设置为on，可以让数据不用经过用户buffer。

（29）server是在http中，可以定义多个虚拟server

（30）server\_name

在server块中，可以通过server\_name来配置server的多域名，匹配的顺序如下：

1、完全匹配

2、后缀匹配

3、前缀匹配

4、正则匹配

5、listen指令里配置了default或default\_server的server

6、第一个匹配上listen的server

（31）server\_tokens是隐藏错误页面和http响应头的nginx版本号

（32）tcp\_nodelay

on数据立即发送

off数据每积攒0.2秒发送

（33）tcp\_nopush数据每积攒一定大小发送

（34）try\_files是按顺序检测文件的存在,并且返回第一个找到的文件

（35）types

1、MIME-type和Content-Type的关系：

当web服务器收到静态的资源文件请求时，依据请求文件的后缀名在服务器的MIME配置文件中找到对应的MIME Type，再根据MIME Type设置HTTP Response的Content-Type，然后浏览器根据Content-Type的值处理文件。

2、MIME Type是经过ietf组织协商，以 RFC 的形式作为建议的标准发布在网上的。如果是某个客户端自己定义的格式，一般只能以 application/x- 开头。

（36）变量https://www.cnblogs.com/qytang/p/5542264.html

nginx配置文件中内置变量以美元符$开始，也叫全局变量。

$arg\_PARAMETER是GET请求中变量名PARAMETER参数的值。

$args是GET请求中的参数。

$binary\_remote\_addr 二进制码形式的客户端地址。

$body\_bytes\_sent 传送页面的字节数

$content\_length 请求头中的Content-length字段。

$content\_type 请求头中的Content-Type字段。

$cookie\_COOKIE是cookie COOKIE的值。

$document\_root 当前请求在root指令中指定的值。

$document\_uri 与$uri相同。

$host 请求中的Host字段，如果请求中的Host字段不可用或者空，则为处理请求的server的server\_name指令的值。值为小写，不包含端口。

$http\_HEADER HTTP请求头中的内容，HEADER为HTTP请求中的内容转为小写，-变为\_，例如：$http\_user\_agent(Uaer-Agent的值)

$is\_args 如果$args设置，值为"?"，否则为""。

$limit\_rate 限制连接速率。

$query\_string 与$args相同。

$remote\_addr 客户端的IP地址。

$remote\_port 客户端的端口。

$remote\_user 已经经过Auth Basic Module验证的用户名。

$request\_filename 当前连接请求的文件路径，由root或alias指令与URI请求生成。

$request\_body 这个变量（0.7.58+）包含请求的主要信息

$request\_body\_file 客户端请求主体信息的临时文件名。

$request\_completion 如果请求成功，设为"OK"；如果请求未完成或者不是一系列请求中最后一部分则设为空。

$request\_method 客户端请求的动作，通常为GET或POST。

$request\_uri 包含一些客户端请求参数的原始URI，它无法修改

$scheme 所用的协议，比如http或者是https

$server\_addr 服务器地址，在完成一次系统调用后可以确定这个值，如果要绕开系统调用，则必须在listen中指定地址并且使用bind参数。

$server\_name 服务器名称。

$server\_port 请求到达服务器的端口号。

$server\_protocol 请求使用的协议，通常是HTTP/1.0或HTTP/1.1。

$uri 请求中的当前URI(不带请求参数，参数位于$args)，不同于浏览器传递的$request\_uri的值，它可以通过内部重定向，或者使用index指令进行修改。不包括协议和主机名

HTTP Proxy模块

HTTP Proxy模块转发请求到其他服务器，转发请求到其他服务器之前nginx将缓存它们

（1）proxy\_pass

反向代理指代理外网用户请求到内网指定web服务器

如果proxy\_pass后面的url后加/，不带匹配的路径。如果proxy\_pass后面的url后不加/，带匹配的路径

（2）proxy\_hide\_header通过反向代理，响应客户端时，隐藏后端服务版本

（3）proxy\_pass\_header与proxy\_hide\_header功能相反

（4）proxy\_redirect修改后端服务器响应头部中的location或refresh，与proxy\_pass配合使用

（5）proxy\_set\_header更改或添加客户端请求头部信息

Http Rewrite模块

（1）if

~和!~是否区分大小正则

~\*和!~\*是否不区分大小写正则

-f和!-f检查一个文件是否存在

-d和!-d检查一个目录是否存在

-e和!-e检查一个文件、目录、符号链接是否存在

-x和!-x检查一个文件是否可执行

（2）return停止处理并为客户端返回状态码，默认302状态码

return code [text] 文本将被放置在响应主体

return code URL URL将成为location头部值

（3）rewrite

执行顺序是

1处理在server中定义的模块指令

2为请求查找location

3处理在选中的location中定义的模块指令

循环至多重复10次，之后nginx返回错误500

重写后若以”http://”或”https://”开始，将停止继续处理，直接返回给客户端。

flag将停止继续处理，flag可以是如下参数：

last，完成该rewrite规则的执行后，停止处理后续rewrite指令集；重新查找新URI

break，完成该rewrite规则的执行后，停止处理后续rewrite指令集，不重新查找新URI。但当前location内非rewrite语句和location外非rewrite语句可以执行；

redirect，返回302临时重定向，地址栏会显示跳转后的地址

permanent，返回301永久重定向，地址栏会显示跳转后的地址

重写后的URL包含原URL的请求参数，即原URL的?后内容。若不想带原URL的参数，可以在重写后的URL后加一个问号

若使用}或;需用双引号或单引号包围

（4）set

定义一个变量并赋值，值可以是文本，变量或者文本变量混合体。