# 第八单元 Nginx安装与配置

# 【授课重点】

1. 了解Nginx负载均衡的概念
2. 掌握Nginx的应用场景
3. 掌握Nginx的编译安装
4. 了解Nginx的目录结构
5. 掌握Nginx配置文件
6. 掌握Nginx配置参数
7. 掌握服务启动/停止/重启命令
8. Nginx虚拟主机配置

# 【考核要求】

1. 掌握Nginx负载均衡
2. 掌握Nginx的应用场景
3. 掌握Nginx的编译安装
4. 掌握Nginx的目录结构
5. 掌握Nginx配置文件
6. 掌握Nginx虚拟主机配置

# 【教学内容】

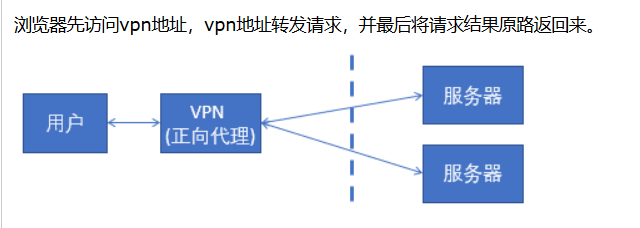
## 8.1 课程导入

### 代理的概念

任务交给中间人协助处理，服务方提供者与被服务方不接触。

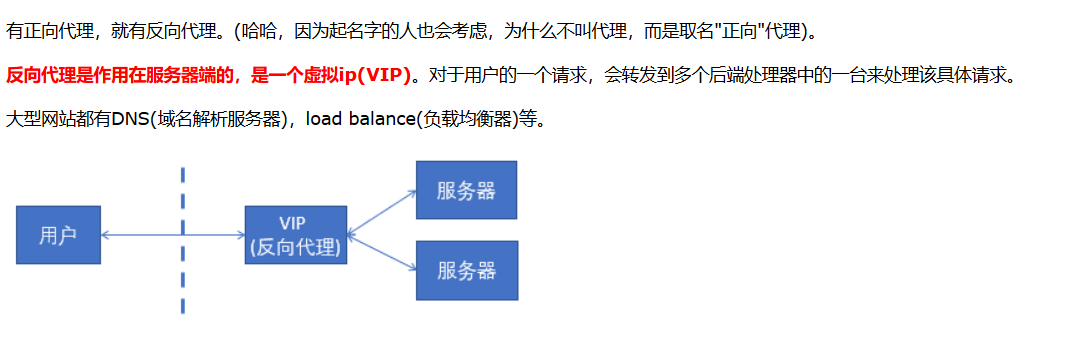
### 正向代理

接受服务方指定代理人的方式，称之正向代理。



### 反向代理

由服务方指定代理者的方式，称之为反向代理。



### 反向代理优点

负载均衡

安全，有效的防止攻击。

缓存的作用：访问频度非常大的内容

静态内容的存取。

## 8.2 什么是Nginx

Nginx 是一款高性能的 http 服务器/反向代理服务器及电子邮件（IMAP/POP3）代理服务器。由俄罗斯的程序设计师 Igor Sysoev 所开发，官方测试 nginx 能够支支撑 5 万并发链接，并且 cpu、内存等资源消耗却非常低，运行非常稳定。



### 8.2.1 Nginx 应用场景

1、http 服务器。Nginx 是一个 http 服务可以独立提供 http 服务。可以做网页静态服务器。

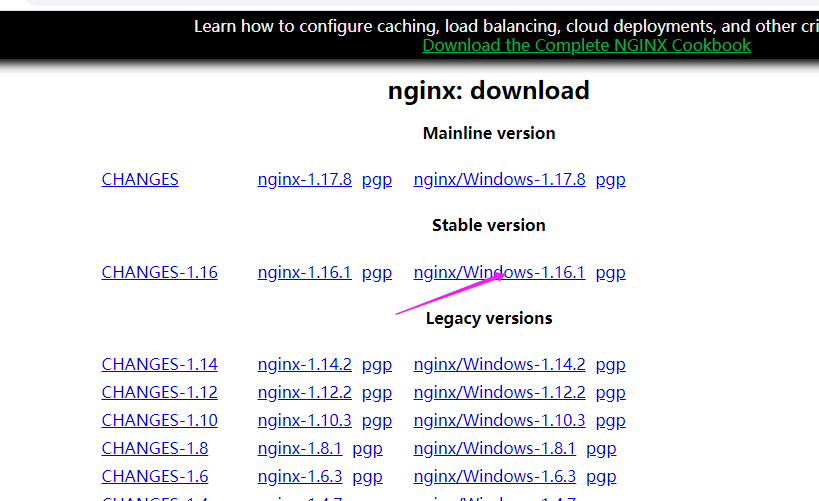
2、虚拟主机。可以实现在一台服务器虚拟出多个网站。例如个人网站使用的虚拟主机。

3、反向代理，负载均衡。当网站的访问量达到一定程度后，单台服务器不能满足用户的请求时，需要用多台服务器集群可以使用 nginx 做反向代理。并且多台服务器可以平均分担负载，不会因为某台服务器负载高宕机而某台服务器闲置的情况

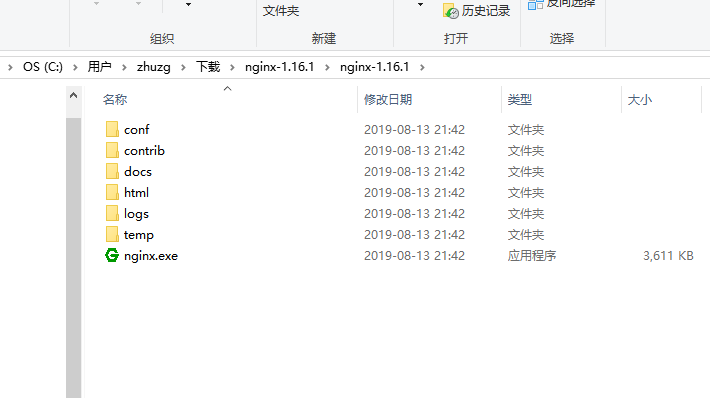
# Nginx的在Windows下的安装

1、找到官网



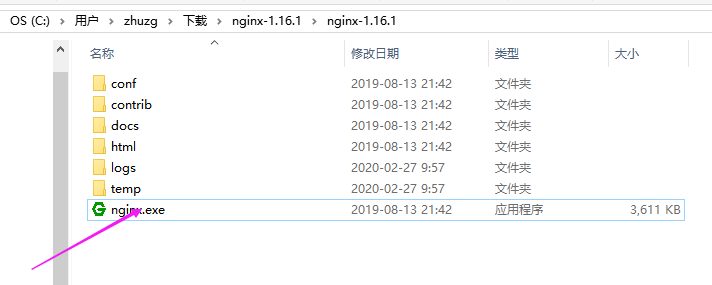
2、

1. 解压即可

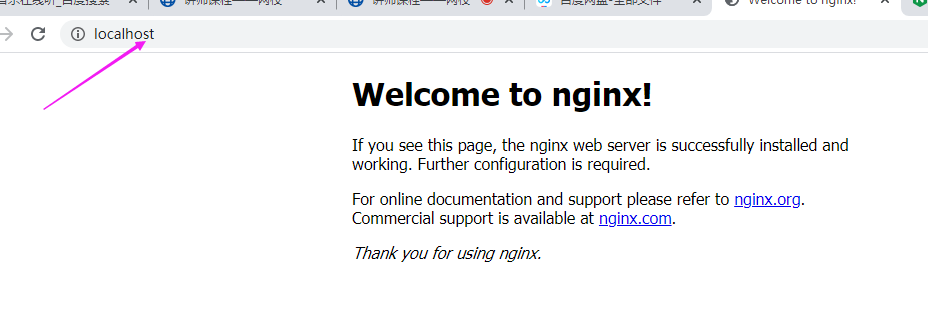
* 

4、使用

双击



5、 出现这个页码 就是安装成功了



## 8.3 Nginx的安装

### 8.3.0 nginx 安装前置条件

nginx是C语言开发，建议在linux上运行，本教程使用Centos6.5作为安装环境。

1. gcc 安装nginx需要先将官网下载的源码进行编译，编译依赖gcc环境，如果没有gcc环境，需要安装gcc：

* yum install gcc-c++

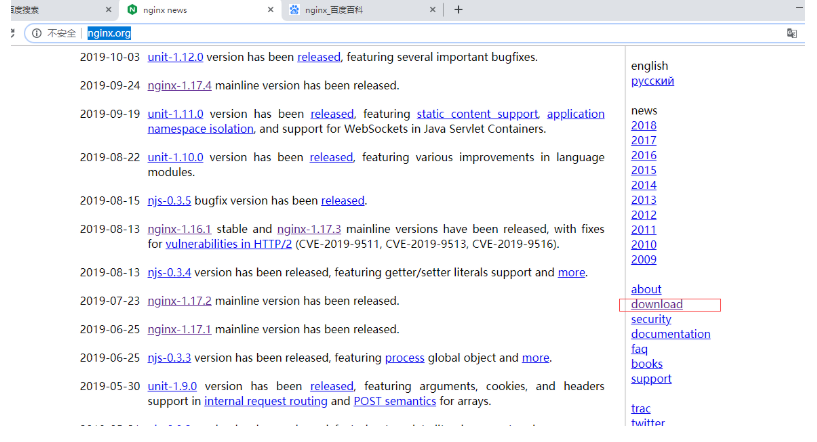
1. PCRE PCRE(Perl Compatible Regular Expressions)是一个Perl库，包括 perl 兼容的正则表达式库。nginx的http模块使用pcre来解析正则表达式，所以需要在linux上安装pcre库。 yum install -y pcre pcre-devel 注：pcre-devel是使用pcre开发的一个二次开发库。nginx也需要此库。
2. zlib zlib库提供了很多种压缩和解压缩的方式，nginx使用zlib对http包的内容进行gzip，所以需要在linux上安装zlib库。 yum install -y zlib zlib-devel
3. openssl OpenSSL 是一个强大的安全套接字层密码库，囊括主要的密码算法、常用的密钥和证书封装管理功能及SSL协议，并提供丰富的应用程序供测试或其它目的使用。 nginx不仅支持http协议，还支持https（即在ssl协议上传输http），所以需要在linux安装openssl库。 yum install -y openssl openssl-devel

### ====Linux下源码安装的一般步骤

1. 解压
2. configure 命令 =》设置编译的选项以及安装的选项
3. make =》 将源码编译成目标文件的过程 （类似于Java .java 文件编译成.class 文件的过程。complie ，。）
4. make install 将目标文件拷贝安装目录的过程。

### 8.3.1 Nginx安装

1. 官网下载nginx http://nginx.org/



1. 点击downLoad

下载页面 http://nginx.org/en/download.html

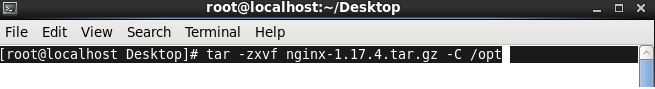


1. 下载

* fig:

1. 上传到linux中

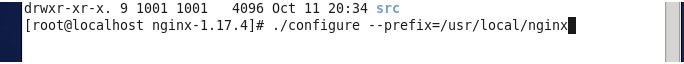
解压到指定的目录中



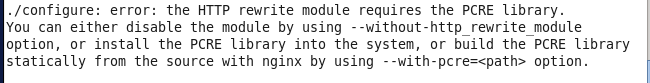
1. 如果需要解压 解压命令格式：tar -zxvf nginx-1.16.1.tar.gz
2. 切换到安装的目录中

fig:

1. #检查安装环境,并指定将来要安装的路径

* ./configure --prefix=/usr/local/nginx
* 
* **---- 知识点小贴士 ----**
* Makefile是一种配置文件， Makefile 一个工程中的源文件不计数，其按类型、功能、模块分别放在若干个目录中，makefile定义了一系列的规则来指定，哪些文件需要先编译，哪些文件需要后编译，哪些文件需要重新编译，甚至于进行更复杂的功能操作，因为 makefile就像一个Shell脚本一样，其中也可以执行操作系统的命令。
* ---- 知识点小贴士 ----  
  configure参数  
  ./configure \  
  --prefix=/usr \                                        指向安装目录  
  --sbin-path=/usr/sbin/nginx \                         指向（执行）程序文件（nginx）  
  --conf-path=/etc/nginx/nginx.conf \                    指向配置文件  
  --error-log-path=/var/log/nginx/error.log \            指向log  
  --http-log-path=/var/log/nginx/access.log \            指向http-log  
  --pid-path=/var/run/nginx/nginx.pid \                  指向pid  
  --lock-path=/var/lock/nginx.lock \                     （安装文件锁定，防止安装文件被别人利用，或自己误操作。）  
  --user=nginx \  
  --group=nginx \  
  --with-http\_ssl\_module \           启用ngx\_http\_ssl\_module支持（使支持https请求，需已安装openssl）  
  --with-http\_flv\_module \           启用ngx\_http\_flv\_module支持（提供寻求内存使用基于时间的偏移量文件）  
  --with-http\_stub\_status\_module \   启用ngx\_http\_stub\_status\_module支持（获取nginx自上次启动以来的工作状态）  
  --with-http\_gzip\_static\_module \   启用ngx\_http\_gzip\_static\_module支持（在线实时压缩输出数据流）  
  --http-client-body-temp-path=/var/tmp/nginx/client/ \ 设定http客户端请求临时文件路径  
  --http-proxy-temp-path=/var/tmp/nginx/proxy/ \ 设定http代理临时文件路径  
  --http-fastcgi-temp-path=/var/tmp/nginx/fcgi/ \ 设定http fastcgi临时文件路径  
  --http-uwsgi-temp-path=/var/tmp/nginx/uwsgi \ 设定http uwsgi临时文件路径  
  --http-scgi-temp-path=/var/tmp/nginx/scgi \ 设定http scgi临时文件路径  
  --with-pcre 启用pcre库

1. 环境有异常：



缺少 gcc pcre-devel openssl openssl-devel 环境

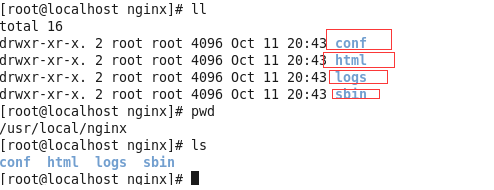
1. #使用YUM安装缺少的包

yum -y install gcc pcre-devel openssl openssl-devel

1. 编译安装

make && make install

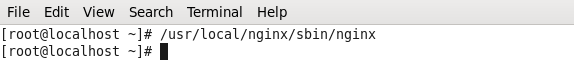
1. 查看安装是否成功



### 8.3.2 启动nginx

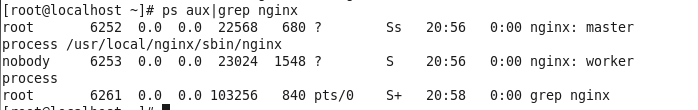
启动nginx 的方式采用绝对路径的方式

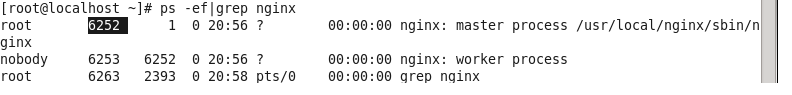
/usr/local/nginx/sbin/nginx



启动成功

启动后查看进程





### 8.3.3 网页访问

默认的端口号:80



### 8.3.4 关闭nginx

关闭nginx操作

fig:

相对路径关闭操作

!

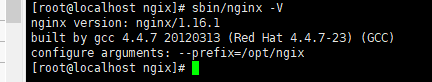
### 8.3.5 重启nginx



-v : show version and exit -V : show version and configure options then exit -t : test configuration and exit -T : test configuration, dump it and exit -q : suppress non-error messages during configuration testing -s signal : send signal to a master process: stop, quit, reopen, reload -p prefix : set prefix path (default: /opt/ngix/) -c filename : set configuration file (default: conf/nginx.conf) -g directives : set global directives out of configuration file

sbin\nginx -v 显示版本号

sbin\nginx -V 显示版本号并且显示安装配置



sbin/nginx -t

检查配置文件是否正确

sbin/nginx -T

检查配置文件是否正确并且保存配置信息

-s signal : send signal to a master process: stop, quit, reopen, reload

sbin/nginx -s stop 让服务程序关闭 ， kill pid -9

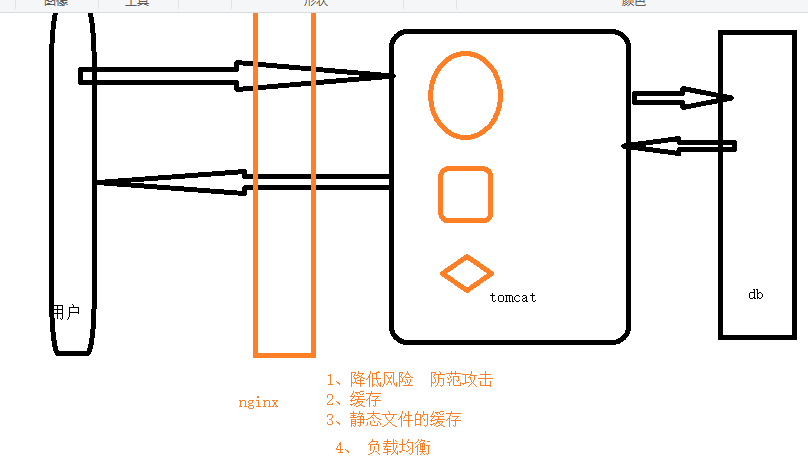
sbin/nginx -s quit 让服务程序优雅的关闭

sbin/nginx -s reopen 重新记录日志

sbin/nginx -s reload 重新加载配置文件

sbin/nginx -c 文件名 使用指定配置文件去启动nginx

nginx 的作用



## 8.4 Nginx目

## 8.4 Nginx目录结构

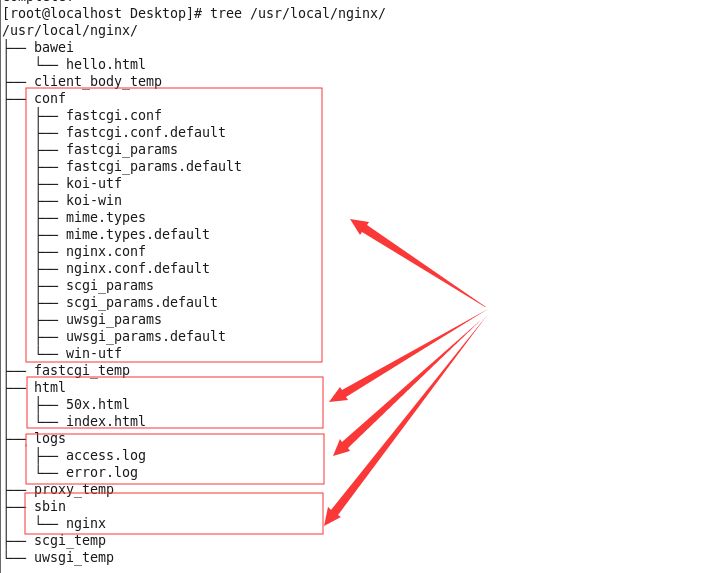
### 8.4.1 tree命令

Linux tree命令用于以树状图列出目录的内容。执行tree指令，它会列出指定目录下的所有文件，包括子目录里的文件。

首先执行 tree 的安装

yum install tree -y

### 8.4.3 执行tree命令



目录介绍

conf #这是Nginx所有配置文件的目录，极其重要  
html #这是编译安装时Nginx的默认站点目录，类似Apache的默认站点htdocs目录  
logs #这是Nginx默认的日志路径，包括错误日志及访问日志  
sbin #这是Nginx命令的目录，如Nginx的启动命令nginx

## 8.5 Nginx配置文件

如果你下载好啦，你的安装文件，不妨打开conf文件夹的nginx.conf文件，Nginx服务器的基础配置，默认的配置也存放在此。

### 8.5.1 初识nginx.conf

井号作用：在 nginx.conf 的注释符号为：

这里的注释不能使用 // /\*\*/

大括弧{}作用：

分号 ：

指令：

模块：

默认的 nginx 配置文件 nginx.conf 内容如下：

#使用哪个用户运行工作进程

#user nobody;  
  
#工作进程数量 主进程主要控制各个工作进程 工作进程用于处理用户的请求连接  
# 通常情况下根据计算机的cpu的核数设置，有几个核就设置几。  
worker\_processes 2;  
  
#error\_log logs/error.log;  
#error\_log logs/error.log notice;  
#error\_log logs/error.log info;  
  
#pid logs/nginx.pid;  
  
  
events {  
 #每个工作进程可以连接的最大的请求数量 2\*1024 = 2048 .  
 worker\_connections 1024;  
}  
  
  
http {  
 include mime.types;  
 default\_type application/octet-stream;  
  
 #log\_format main '$remote\_addr - $remote\_user [$time\_local] "$request" '  
 # '$status $body\_bytes\_sent "$http\_referer" '  
 # '"$http\_user\_agent" "$http\_x\_forwarded\_for"';  
  
 #access\_log logs/access.log main;  
  
 sendfile on;  
 #tcp\_nopush on;  
  
 #keepalive\_timeout 0;  
 keepalive\_timeout 65;  
  
 #gzip on;  
  
 server {  
 listen 80;  
 server\_name localhost;  
  
 #charset koi8-r;  
  
 #access\_log logs/host.access.log main;  
  
 location / {  
 root html;  
 index index.html index.htm;  
 }  
  
 #error\_page 404 /404.html;  
  
 # redirect server error pages to the static page /50x.html  
 #  
 error\_page 500 502 503 504 /50x.html;  
 location = /50x.html {  
 root html;  
 }  
  
 # proxy the PHP scripts to Apache listening on 127.0.0.1:80  
 #  
 #location ~ \.php$ {  
 # proxy\_pass http://127.0.0.1;  
 #}  
  
 # pass the PHP scripts to FastCGI server listening on 127.0.0.1:9000  
 #  
 #location ~ \.php$ {  
 # root html;  
 # fastcgi\_pass 127.0.0.1:9000;  
 # fastcgi\_index index.php;  
 # fastcgi\_param SCRIPT\_FILENAME /scripts$fastcgi\_script\_name;  
 # include fastcgi\_params;  
 #}  
  
 # deny access to .htaccess files, if Apache's document root  
 # concurs with nginx's one  
 #  
 #location ~ /\.ht {  
 # deny all;  
 #}  
 }  
  
  
 # another virtual host using mix of IP-, name-, and port-based configuration  
 #  
 #server {  
 # listen 8000;  
 # listen somename:8080;  
 # server\_name somename alias another.alias;  
  
 # location / {  
 # root html;  
 # index index.html index.htm;  
 # }  
 #}  
  
  
 # HTTPS server  
 #  
 #server {  
 # listen 443 ssl;  
 # server\_name localhost;  
  
 # ssl\_certificate cert.pem;  
 # ssl\_certificate\_key cert.key;  
  
 # ssl\_session\_cache shared:SSL:1m;  
 # ssl\_session\_timeout 5m;  
  
 # ssl\_ciphers HIGH:!aNULL:!MD5;  
 # ssl\_prefer\_server\_ciphers on;  
  
 # location / {  
 # root html;  
 # index index.html index.htm;  
 # }  
 #}  
  
}

### 8.5.2 nginx 文件结构

#全局块

配置好以后，这里的变量对整个nginx 都起作用

#events块

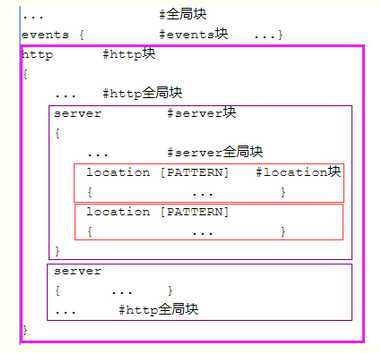
events { ... }

#http块

nginx 相应http 请求如何去处理。

http   
{ ... #http全局块 server #server块 { ... #server全局块 location [PATTERN] #location块 { ... } location [PATTERN] { ... } } server { ... } ... #http全局块 }

图形化表示：



* 1、**全局块**：配置影响nginx全局的指令。一般有运行nginx服务器的用户组，nginx进程pid存放路径，日志存放路径，配置文件引入，允许生成worker process数等。
* 2、**events块**：配置影响nginx服务器或与用户的网络连接。有每个进程的最大连接数，选取哪种事件驱动模型处理连接请求，是否允许同时接受多个网路连接，开启多个网络连接序列化等。
* 3、**http块**：可以嵌套多个server，配置代理，缓存，日志定义等绝大多数功能和第三方模块的配置。如文件引入，mime-type定义，日志自定义，是否使用sendfile传输文件，连接超时时间，单连接请求数等。
* 4、**server块**：配置虚拟主机的相关参数，一个http中可以有多个server。
* 5、**location块**：配置请求的路由，以及各种页面的处理情况。

### 8.5.3 nginx.conf 配置项解释

########### 每个指令必须有分号结束。#################  
#user administrator administrators; #配置用户或者组，默认为nobody nobody。  
#worker\_processes 2; #允许生成的进程数，默认为1  
#pid /nginx/pid/nginx.pid; #指定nginx进程运行文件存放地址  
error\_log log/error.log debug; #制定日志路径，级别。这个设置可以放入全局块，http块，server块，级别以此为：debug|info|notice|warn|error|crit|alert|emerg  
events {  
 accept\_mutex on; #设置网路连接序列化，防止惊群现象发生，默认为on  
 multi\_accept on; #设置一个进程是否同时接受多个网络连接，默认为off  
 #use epoll; #事件驱动模型，select|poll|kqueue|epoll|resig|/dev/poll|eventport  
 worker\_connections 1024; #最大连接数，默认为512  
}  
http {  
 include mime.types; #文件扩展名与文件类型映射表  
 default\_type application/octet-stream; #默认文件类型，默认为text/plain  
 #access\_log off; #取消服务日志   
 log\_format myFormat '$remote\_addr–$remote\_user [$time\_local] $request $status $body\_bytes\_sent $http\_referer $http\_user\_agent $http\_x\_forwarded\_for'; #自定义格式  
 access\_log log/access.log myFormat; #combined为日志格式的默认值  
 sendfile on; #允许sendfile方式传输文件，默认为off，可以在http块，server块，location块。  
 sendfile\_max\_chunk 100k; #每个进程每次调用传输数量不能大于设定的值，默认为0，即不设上限。  
 keepalive\_timeout 65; #连接超时时间，默认为75s，可以在http，server，location块。  
  
 upstream mysvr {   
 server 127.0.0.1:7878;  
 server 192.168.10.121:3333 backup; #热备  
 }  
 error\_page 404 https://www.baidu.com; #错误页  
 server {  
 keepalive\_requests 120; #同一个长连接请求上限次数。  
 listen 4545; #监听端口  
 server\_name 127.0.0.1; #监听地址   
 location ~\*^.+$ { #请求的url过滤，正则匹配，~为区分大小写，~\*为不区分大小写。  
 #root path; #根目录  
 #index vv.txt; #设置默认页  
 proxy\_pass http://mysvr; #请求转向mysvr 定义的服务器列表  
 deny 127.0.0.1; #拒绝的ip  
 allow 172.18.5.54; #允许的ip   
 }   
 }  
}

PS

惊群现象:https://blog.csdn.net/zhuzg2005/article/details/104500937

sendfile: sendfile是 Linux2.0+以后推出的一个系统调用,在两个文件描述符之间传递数据，也就是完全在内核中操作。这样就避免了内核缓冲区和用户缓冲区之间的数据拷贝，效率很高，这种技术又被称为零拷贝。

上面是nginx的基本配置，需要注意的有以下几点：

1、几个常见配置项：

* 1.$remote*addr 与 $http*x*forwarded*for 用以记录客户端的ip地址；
* 2.$remote\_user ：用来记录客户端用户名称；
* 3.$time\_local ： 用来记录访问时间与时区；
* 4.$request ： 用来记录请求的url与http协议；
* 5.$status ： 用来记录请求状态；成功是200；
* 6.$body*bytes*s ent ：记录发送给客户端文件主体内容大小；
* 7.$http\_referer ：用来记录从那个页面链接访问过来的；
* 8.$http*user*agent ：记录客户端浏览器的相关信息；

2、惊群现象：一个网路连接到来，多个睡眠的进程被同事叫醒，但只有一个进程能获得链接，这样会影响系统性能。

3、每个指令必须有分号结束。

## 8.6 Nginx虚拟主机配置

### 8.6.1 配置虚拟主机

虚拟主机，也叫“网站空间”，就是把一台运行在互联网上的物理服务器划分成多个“虚拟”服务器。虚拟主机技术极大的促进了网络技术的应用和普及。同时虚拟主机的租用服务也成了网络时代的一种新型经济形式。

### 8.6.2 端口绑定

（1）上传静态网站：

将前端静态页cart.html 以及图片样式等资源 上传至 /usr/local/nginx/cart 下

将前端静态页search.html 以及图片样式等资源 上传至 /usr/local/nginx/search 下

（2）修改Nginx 的配置文件：/usr/local/nginx/conf/nginx.conf

server {  
 listen 81;  
 server\_name localhost;  
 location / {  
 root cart;  
 index cart.html;  
 }   
 }  
 server {  
 listen 82;  
 server\_name localhost;  
 location / {  
 root search;  
 index search.html;  
 }   
 }

（3）访问测试：

地址栏输入http://192.168.73.129:81 可以看到购物车页面

地址栏输入http://192.168.23.129:82 可以看到搜索页面

## 8.7 课程小结

1. 了解nginx的应用场景
2. 掌握nginx的环境搭建
3. 掌握nginx的目录结构
4. 掌握配置nginx.conf配置文件中单个选项配置的含义
5. 掌握nginx的虚拟主机配置

## 8.8 课下作业

1. 搭建nginx配置文件（20分钟)
2. 配置nginx转发到tomcat项目(10分钟)
3. 虚拟主机配置(10分钟)