**目 录**

[相关依赖包 3](#_Toc1668332741)

[服务启动 4](#_Toc931436207)

[查看网络端口信息命令 5](#_Toc2089852357)

[升级nginx服务器 6](#_Toc588161854)

[主配置文件(location在server中) 7](#_Toc343238853)

[文件目录 11](#_Toc229126110)

[用户认证 12](#_Toc287199589)

[ssl加密虚拟主机 13](#_Toc1105950301)

[构建LNMP平台 14](#_Toc820839889)

[地址重写 15](#_Toc884569276)

[调度器 16](#_Toc119874578)

[常见问题处理 17](#_Toc1320937977)

[Memcached服务 19](#_Toc1016552129)

相关依赖包

Nginx安装所需依赖包：gcc、pcre-devel、

用户认证密码生成软件：httpd-tools

SSL加密虚拟主机模块：http\_ssl\_module

Mariadb软件和依赖包：mariadb、mariadb-server、mariadb-devel

Php相关依赖包：php、php-mysql、php-fpm

TCP/UDP调度器nginx需要添加模块：stream

服务启动

启动服务：/usr/local/nginx/sbin/nginx

设施软连接：ln -s /usr/local/nginx/sbin/nginx /sbin

关闭服务：nginx -s stop

重新加载配置文件： nginx -s reload

查看软件信息：nginx -V

启动mysql服务：systemctl start mariadb

查看mysql服务状态：systemctl status mariadb

启动php-fpm服务：systemctl start php-fpm

查看网络端口信息命令

Netstat可以查看系统中启动的端口信息，可简化为“ss”

选项

-a显示所有端口的信息

-n以数字格式显示端口号

-t显示TCP连接的端口

-u显示UDP连接的端口

-l显示服务正在监听的端口信息，如httpd启动后，会一直监听80端口

-p显示监听端口的服务名称是什么（也就是程序名称）

nginx服务默认通过TCP 80端口监听客户端请求：

root@proxy ~]# netstat -anptu | grep nginx

升级nginx服务器

备份旧主程序，解压新程序，配置，编译，但是不要安装，将新程序的主程序复制到/usr/local/nginx/sbin下，升级make upgrade即可。

[root@proxy~]#/usr/local/nginx/sbin/nginx –v //查看版本

主配置文件(location在server中)

配置文件目录：/usr/local/nginx/conf/nginx.conf

优化并发

worker\_processes 2;

events {

worker\_connections 32323232;

use epoll; }

http {

用户认证

server {

listen 80;

server\_name www.a.com;

auth\_basic “aaaa” ;

auth\_basic\_user\_file “/usr/local/nginx/pass”;

root html; }

server {

限定IP

listen 192.168.4.5:80;

server\_name www.b.com;

root aaa; }

server {

SSL加密

listen 443 ssl;

server\_name www.c.com；

ssl\_certificate cert.pem；

ssl\_certificate\_key cert.key； }

server{

网址重写

listen 80;

rewrite ^/(.\*)$ http://www.aaa.com/$1； }

location / {

网页重写

root html;

index index.html index.htm;

rewrite /a.html /b.html 可加选项； }

location ~\.php$ {

动态网页

root html;

fastcgi\_pass 127.0.0.1:9000;

fastcgi\_index index.php;

include fastcgi.conf; }

访问重写

if ($http\_user\_agent ~\*firefox) {

rewrite ^(.\*)$ /bbb/$1； } }

stream { 注意在http外部

调度器

upstream aaa {

server ip1 ;

server ip2; }

server {

listen 32323232;

proxy\_pass aaa; } }

http {

client\_header\_buffer\_size 1k;

头部缓存

large\_client\_header\_buffers 4 4k;

upstream aaa {

ip\_hash

server ip1 weight=1 max\_fails=1 fail\_timeout=30;

server ip2;

server ip3 down； }

server {

调度器

listen 80;

server\_name www.d.com;

location / {

proxy\_pass http://aaa; } }

静态缓存

location ～.\*\.(jpg|gif|png)$ {

expires 30d; }

location /status {

服务状态

stub\_status on; }

gzip on;

开启压缩

gzip\_min\_length 1000;

gzip\_comp\_level 4;

gzip\_types text/plain text/css application/json applicatio

open\_file\_cache max=2000 inactive=20s;

设置缓存

open\_file\_cache\_valid 60s;

open\_file\_cache\_min\_uses 5;

open\_file\_cache\_errors off; }

文件目录

配置文件目录：/usr/local/nginx/conf/nginx.conf

网页根目录：/usr/local/nginx/html

Php-fpm配置文件：/etc/php-fpm.d/www.conf

Nginx默认访问日志：/usr/local/nginx/logs/access.log

Nginx默认错误日志：/usr/local/nginx/logs/error.log

Php默认错误日志：/var/log/php-fpm/www-error.log

用户认证

配置文件：

server {

listen 80;

server\_name localhost;

auth\_basic "提示符";  //认证提示符

auth\_basic\_user\_file "/usr/local/nginx/pass"; //认证密码文件

location / {

root html;

index index.html index.htm; }

生成密码文件

软件：yum -y install httpd-tools

命令：htpasswd -c /usr/local/nginx/pass tom

注意：1 密码文件名称要统一

2 “-c” 选项只有在第一次生成文件时使用，追加用户时不用添加“-c”选项。

ssl加密虚拟主机

加密算法有：

对称算法：AES、DES、主要用于单机加密

非对称算法：RSA、DSA，主要用于网络加密

信息摘要：MD5、SHA256，数据完整性校验，数据秒传。

生成私钥和证书

cd /usr/local/nginx/conf

openssl genrsa > cert.key

openssl req -new -x509 -key cert.key > cert.pem

配置文件：

server {

listen 443 ssl;

server\_name www.c.com;

ssl\_certicate cert.pem;

ssl\_certificate\_key cert.key; }

构建LNMP平台

需要安装：gcc、openssl-devel、pcre-devel、zlib-devel、mariadb、mariadb-server、mariadb-devel、php、php-mysql、php-fpm(yum中没有）

需要启动的服务：ngind、mariadb、php-fpm、

查询服务状态：systemctl status 服务名称

nginx配置文件：

location \.php$ {

root hrml;

fastcgi\_pass 127.0.0.1:9000;

fastcgi\_index index.php;

include fastcgi.conf; }

创建测试页面：vim /usr/local/nginx/html/test.php

<?php

$i=111;

echo $i;

?>

客户端访问页面：firefox http://192.168..../test.php

地址重写

**网页**：nginx配置文件：/usr/local/nginx/conf/nginx

location / {

root html;

index index.html index.htm;

rewrite /a.html /b.html 可加选项；}

**网址**(重定向到aaa网站)：

server {

listen 80;

server www.d.com;

rewrite /(.\*)$ http://www.aaa.com/$1; }

**访问浏览器重写**（根据不同浏览器返回不同页面，例如下列跳转到bbb目录下相对应页面）（配置文件修改不在server中也不再location中，）

if ($http\_user\_agent ~\*firefox ) {

rewrite (.\*)$ /bbb/$1; }

其中：“\*”代表不区分大小写。

地址重写选项：

rewrite 旧地址 新地址 [选项];

last 不再读其他rewrite

break 不再读其他语句，结束请求

redirect 临时重定向

permament 永久重定向

调度器

**Web服务器**

先定义集群，在server中调用集群，注意集群和调用的名称要一致。

配置文件：

http {

upstream aaa {

ip\_hash

server ip1 weight=1 max\_fails=1 fail\_timeout=30;

server ip2

server ip3 down;

... ... }

server {

listen 80;

server\_name www.d.com;

location / {

proxy\_pass http://aaa；} } }

添加ip\_hash选项会使相同客户端访问相同web服务器。

**TCP/UDP调度器**

Nginx需要添加stream模块，配置文件需要在http外部添加。端口需要自己设定一个，不要和其他服务向冲突。

配置文件：

stream {

upstream aaa {

server ip1;

server ip2;

... ... }

server {

listen 12121212121212;

proxy\_pass aaa; } }

常见问题处理

1.并发量太大，显示“Too many open files”

（1）优化nginx配置文件：修改worker——processes 2；与cpu核心数量一致。

修改worker\_connections 32323232;（每个worker的最大并发数。）

（2）优化linux内核参数 ulimit -a 查看属性

Ulimit -Hn 11111 设置硬限制

Ulimit -Sn 11111 设置软限制

修改/etc/security/limits.conf，设置永久参数

\* soft nofile 1111111

\* hard nofile 1211122

1. 头部信息过长

http {

Client\_header\_buffer\_size 1k;

Large\_client\_header\_buffers 4 4k;(最大缓存为16k；

1. 浏览器本地缓存静态数据

Location ~\*\.(jpg|gif|png)$ {

expires 30d;

1. 自定义报错页面

error\_page 404 /40x.html;

设置页面即可。

1. 查看服务器状态信息

需要适用http\_stub\_module模块

Location /status {

Stub\_status on;

}

查看状态页面信息：http：//ip/status

1. 对页面进行压缩

gzip on; //开启压缩

gzip\_min\_length 1000; //小文件不压缩

gzip\_comp\_level 4; //压缩比率

gzip\_types text/plain text/css application/json application/x-javascript text/xml application/xml application/xml+rss text/javascript;

//对特定文件压缩，类型参考mime.types

.. ..

1. 设置服务器内存缓存

http {

open\_file\_cache max=2000 inactive=20s;

open\_file\_cache\_valid 60s;

open\_file\_cache\_min\_uses 5;

open\_file\_cache\_errors off;

//设置服务器最大缓存2000个文件句柄，关闭20秒内无请求的文件句柄

//文件句柄的有效时间是60秒，60秒后过期

//只有访问次数超过5次会被缓存

Memcached服务

内存缓存服务

所需软件：memcached

配置文件：/usr/lib/systemd/system/memcached.service

LNMP相关软件：gcc,openssl-devel, pcre-devel, mariadb, mariadb-server,

Mariadb-devel, php, php-mysql, php-fpm, php-pecl-memcache

服务器需要开启动态转换

Sessiom共享

修改php-fpm配置文件，重启php-fpm服务。

php\_value[session.save\_handler] = memcache

php\_value[session.save\_path] = "tcp://192.168.2.5:11211"