

Nginx服务其他管理配置

原创

GeorgeKai

2018-02-07 13:49:51

评论(0)

989人阅读

1.1 虚拟主机的概念和类型

1.1.1 概念

进行搭建多个网站（搭建多个虚拟主机，一个网站就是一个虚拟主机）

1.1.2 通过域名访问网站的原理（如果访问时不加端口为什么会变成第一个虚拟主机？）

- 1) 客户机通过域名访问bbs.georgekai.com，首先经过DNS解析为对应的IP地址
- 2) 客户端发送数据包到nginx服务器，数据包包括：请求的虚拟主机信息、源和目的端口，源和目的IP
- 3) 服务端接收到客户端的请求报文
 - a 先根据IP地址找对的站点信息，如果没有对应IP直接丢弃报文
 - b 如果匹配了IP，在找对应的端口号，没有对应端口也会丢弃报文
 - c 如果IP和端口号都匹配的情况下，虚拟主机不匹配，默认会显示第一个虚拟主机



@51CTO博客

1.1.3 类型

一、基于域名的虚拟主机配置（最常用）

网站目录和文件环境准备：

1) 基于域名的虚拟主机配置

```
[root@web01 conf]# cat nginx.conf
worker_processes 1;
events {
    worker_connections 1024;
}
http {
    include mime.types;
    default_type application/octet-stream;
    sendfile on;
    keepalive_timeout 65;
    server {
        listen 80;
        server_name www.etiantian.org;
        root html/www;
        index index.html index.htm;
    }
    server {
        listen 80;
        server_name bbs.etiantian.org;
        root html/bbs;
        index index.html index.htm;
    }
}
```

```

server {
    listen      80;
    server_name  blog.etiantian.org;
    root        html/blog;
    index       index.html index.htm;
}

```

网站目录和文件环境准备:

1. `mkdir /application/nginx/html/{www,bbs,blog} -p`
2. `for name in www bbs blog;do echo "10.0.0.7 web01 $name"`
`>/application/nginx/html/$name/index.html;done`
3. `for name in www bbs blog;do cat /application/nginx/html/$name/index.html;done`
`10.0.0.7 web01 www`
`10.0.0.7 web01 bbs`
`10.0.0.7 web01 blog`

检查配置文件语法, 进行服务重启或者启动:

```

[root@web01 conf]# /application/nginx/sbin/nginx -t
nginx: the configuration file /application/nginx-1.12.2/conf/nginx.conf syntax is ok
nginx: configuration file /application/nginx-1.12.2/conf/nginx.conf test is successful

[root@web01 conf]# /application/nginx/sbin/nginx

```

利用curl或者浏览器进行访问测试:

需要对虚拟主机域名进行解析 (编写hosts文件-linux里面hosts文件 windows里面hosts)

```

[root@web01 conf]# for name in www bbs blog;do curl $name.etiantian.org;sleep 1;done
10.0.0.7 web01 www
10.0.0.7 web01 bbs
10.0.0.7 web01 blog

```

通过windows抓包分析: windows配置hosts (10.0.0.7 www.georgekai.com bbs.georgekai.com
blog.georgekai.com)

PS: 当客户端访问nginx服务端, 返回的状态码为304: 表示读取缓存处理

利用抓包软件分析http访问过程:

- 1). 强化记忆http请求报文和响应报文结构信息
- 2). 利用抓包信息分析访问异常问题

二、基于端口的虚拟主机配置

```

[root@web01 conf]# cat nginx.conf
worker_processes 1;
events {
    worker_connections 1024;
}
http {
    include       mime.types;
    default_type  application/octet-stream;
    sendfile      on;
    keepalive_timeout 65;
    server {
        listen      80;
        server_name  www.etiantian.org;
        root        html/www;
        index       index.html index.htm;
    }
    server {
        listen      81;
        server_name  bbs.etiantian.org;
        root        html/bbs;
        index       index.html index.htm;
    }
    server {

```

```

        listen      80;
        server_name  blog.etiantian.org;
        root         html/blog;
        index         index.html index.htm;
    }
}

```

优化nginx操作:

vim /etc/profile ——在最下面添加一条默认搜索路径
 export PATH=\$PATH:/application/nginx/sbin/
 source /etc/profile ——使其生效

PS: 将临时变量写入profile, 执行命令就不需要输出全路径了

重启nginx服务:

nginx -t

测试访问:

bbs.georgekai.com:81

三、基于IP的虚拟主机配置

```

worker_processes 1;
events {
    worker_connections 1024;
}
http {
    include mime.types;
    default_type application/octet-stream;
    sendfile on;
    keepalive_timeout 65;
    server {
        listen 10.0.0.7:80;
        server_name www.etiantian.org;
        root html/www;
        index index.html index.htm;
    }
    # server {
    #     listen 81;
    #     server_name bbs.etiantian.org;
    #     root html/bbs;
    #     index index.html index.htm;
    # }
    # server {
    #     listen 80;
    #     server_name blog.etiantian.org;
    #     root html/blog;
    #     index index.html index.htm;
    # }
}

```

重启nginx服务: (不能平滑重启)

nginx -s stop

nginx

强调说明: 当nginx配置文件中, 涉及到IP地址信息改动, 都需要重启nginx服务, 不能采用平滑重启, 否则配置不生效

1.1.4 企业规范优化Nginx配置文件

将每个虚拟主机的配置文件分开存放

好处: 虚拟机主机分开配置, 不会使一个配置文件中内容过多, 导致排除问题时过乱

1. 创建扩展目录, 生成虚拟主机配置文件

mkdir /application/nginx/conf/extra

```
sed -n '10,15p' nginx.conf >extra/www.conf
sed -n '16,21p' nginx.conf >extra/bbs.conf
sed -n '22,27p' nginx.conf >extra/blog.conf
```

2. 修改nginx主配置文件，加载相应虚拟主机配置文件

```
[root@web01 conf]# cat nginx.conf
worker_processes 1;
events {
    worker_connections 1024;
}
http {
    include mime.types;
    default_type application/octet-stream;
    sendfile on;
    keepalive_timeout 65;
    include extra/www.conf;
    include extra/bbs.conf;
    include extra/blog.conf;
}
```

3. 重启服务，进行检验测试

```
nginx -t
nginx -s reload
```

1.1.5 Nginx虚拟主机的别名配置

1. 在域名后面添加别名信息

```
cat extra/bbs.conf
server {
    listen 80;
    server_name bbs.etiantian.org bbs.org;    ---在域名后面添加别名信息
    root html/bbs;
    index index.html index.htm;
}
```

2. 添加别名信息需要在/etc/hosts文件中添加解析

1.1.6 Nginx状态信息功能实战

为什么在编译安装时，需要配置状态模块？（`--with-http_stub_status_module`）

因为可以查看**nginx运行状态信息**。

配置state虚拟机主机配置文件：

```
vim extra/state.conf
server{
    listen 80;
    server_name state.george.com;
    location / {
        stub_status on;
        access_log off;
    }
}
```

nginx主配置文件中加载状态配置文件：

```
vim nginx.conf
include extra/state.conf;
```

将状态模块域名配置到windows系统的hosts文件中：

```
10.0.0.7 state.george.com
```

状态模块说明：

Active connections: 3

——当前客户端的连接数量（包含waiting连接数）

server accepts handled requests
协议信息）

——accepts:接收客户端连接的总数（只接受http

25 25 37

handled:处理连接的总数（处理的是accepts

接收的数据）

requests:客户端请求的总数（包括TCP建立接

连）

Reading: 0 Writing: 1 Waiting: 2 —— Reading: 监控请求头的连接数
 writing: 监控回应客户端的连接数
 waiting: 监控空闲客户端的连接请求等待数

ngx_http_stub_status_module		
参数信息	参数说明	中文说明
Active connections	The current number of active client connections including Waiting connections.	当前活动客户端连接数量包含Waiting连接数量
accepts	The total number of accepted client connections	接收客户端连接的总数量
handled	The total number of handled connections. Generally, the parameter value is the same as accepts unless some resource limits have been reached (for example, the worker_connections limit).	处理连接的总数量 通常, 这个参数值和accepts值相同, 除非已经达到一些资源限制 (例如: worker_connections连接数限制)
requests	The total number of client requests	客户端请求的总数
Reading	The current number of connections where nginx is reading the request header.	当前nginx正在读取请求头的连接数
Writing	The current number of connections where nginx is writing the response back to the client	当前nginx将响应写回客户端的连接数量
Waiting	The current number of idle client connections waiting for a request.	当前空闲客户端连接等待请求的数量

参考链接: http://nginx.org/en/docs/http/ngx_http_stub_status_module.html#stub_status

@51CTO博客

1.2 Nginx服务日志信息

1.2.1 错误日志

书写格式:

```
Syntax:errpr_log file [level];
Default:error_log logs/error.log error;
Context:main,http,mail,stream,server,location
#error_log logs/error.log;
#error_log logs/error.log notice;
#error_log logs/error.log info;
```

示例:

```
worker_processes 1;
error_log logs/error.log error;
events {
    worker_connections 1024;
}
```

日志信息的级别:

debug —有调式信息的, 日志信息最多
 info —一般信息的日志, 最常用
 notice —最具有重要性的普通条件的信息
 warning —警告级别
 err —错误级别, 阻止某个功能或者模块不能正常工作的信息
 crit —严重级别, 阻止整个系统或者整个软件不能正常工作的信息
 alert —需要立刻修改的信息
 emerg —内核崩溃等严重信息
 none —什么都不记录

从上到下, 级别从低到高, 记录的信息越来越少

详细的可以查看手册: man 3 syslog

1.2.2 访问日志

1) 创建log_format语句

```
vim nginx.conf
http {
    include mime.types;
    default_type application/octet-stream;
    sendfile on;
    keepalive_timeout 65;
    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_www.abc.com.log main;
```

2) 虚拟主机配置文件插入access_log语句

如果访问网站, 没有日志记录, 可查看虚拟主机的配置, 是不是关了 (access_log off;)

```
vim conf/extra/bbs.conf
server {
    listen      80;
    server_name bbs.george.com bbs.com;
    location / {
access_log on;
    }
```

3) 测试并重新加载

```
nginx -t
```

```
nginx -s reload 或 killall -s HUP nginx
```

PS: 可用kill -l查看所有信号(如: HUP对应的信号是1)

Nginx日志格式中默认的参数配置:

```
log_format main '$remote_addr - $remote_user [$time_local] "$request" '
                '$status $body_bytes_sent "$http_referer" '
                '"$http_user_agent" "$http_x_forwarded_for"';
```

日志格式说明:

log_format main '\$remote_addr - \$remote_user [\$time_local] "\$request" ' --- 定义日志信息要记录的内容格式

```
                '$status $body_bytes_sent "$http_referer" '
                '"$http_user_agent" "$http_x_forwarded_for" ';
access_log logs/access.log main;          --- 调用定义格式信息, 生成访问日志
$remote_addr      10.0.0.1                --- 访问客户端的源地址信息
$remote_user      -                      --- 访问客户端认证用户信息 (用户登录网站需要账号密码)
[$time_local]     --- 显示访问时间
$request          GET / HTTP/1.1          --- 请求行信息
$status           304                    --- 状态码信息 (304状态码利用缓存显示页面信息)
$body_bytes_sent  --- 服务端响应客户端的数据大小信息
$http_referer     --- 记录链接到网站的域名信息
$http_user_agent  --- 用户访问网站客户端软件标识信息
$http_x_forwarded_for --- 反向代理
```

PS: 用户利用客户端浏览器测试访问时, win10默认浏览器会有异常问

1.2.3 日志要进行切割

1. 利用shell脚本切割

```
1) vim /server/scripts/cutting.sh
```

```
#!/bin/bash
```

```
date_info=$(date +%F-%H:%M)
```

```
mv /application/nginx/logs/access.log /application/nginx/logs/access.log.$date_info
/application/nginx/sbin/nginx -s reload
```

```
2) crontab -e
```

```
# cut nginx log cron
```

```
00 */6 * * * /bin/sh /server/scripts/cutting.sh &>/dev/null
```

2. 可以使用logrotate工具(系统自带日志切割工具)

作用: logrotate的配置文件是/etc/logrotate.conf, 通常不需要对它进行修改。日志文件的轮循设置在独立的配置文件中, 它们放在/etc/logrotate.d/目录下。

logrotate全局配置参数说明:

```
vim /etc/logrotate.conf
```

weekly ###日志文件将按月轮循

rotate 4 ###一次将最多存储4个归档日志

create ###在生成轮询日志后, 会自动创建一个新的文件

dateext ###生成的轮询日志后面加上时间格式

compress ###生成的轮询日志是否压缩, 默认被注释

include /etc/logrotate.d ###存放日志文件的轮询配置

示例: 创建一个日志, 配置logrotate来轮询一个日志

```
vim /etc/logrotate.d/log-file
```

```

/var/log/log-file {

    monthly

    rotate 5

    compress

    delaycompress

    missingok

    notifempty

    create 644 root root

    postrotate

        /usr/bin/killall -HUP rsyslogd

    endscript

}

```

@51CTO博客

PS:通用格式，其中有些参数可以不设置

1.2.4 location区块的作用？

企业需求解决：

搭建好一台nginx的web服务器。配置好内网卡地址与外网卡地址

web服务的网站域名为www.etiantian.org，站点目录为html/www

要求内网用户可以访问网站http://www.etiantian.org/AV资源信息

要求外网用户禁止访问网站http://www.etiantian.org/AV资源信息

部署过程：

1) 精确控制允许和拒绝：

```

location / {
    allow 172.16.1.0/24; ——允许172.16.1.0网段访问
    deny all; ——拒绝所有访问
}

```

2) 访问/AV 资源有访问控制

定位/AV资源不能随意访问

location 进行资源定位 ——相当于if 判断，满足什么做什么

```

location /AV {
}

```

3) 创建测试访问资源信息

mkdir /application/nginx/html/www/AV

-H host:www.etiantian.org --- 表示指定访问nginx服务端哪一个虚拟主机

测试：

[root@nfs01 ~]# curl -H host:www.etiantian.org 172.16.1.7/AV/kai.html
kai123 ——可以访问

```

[root@nfs01 ~]# curl -H host:www.etiantian.org 10.0.0.7/AV/kai.html
<html>
<head><title>403 Forbidden</title></head> ——403拒绝访问
<body bgcolor="white">
<center><h1>403 Forbidden</h1></center>

```

```
<hr><center>nginx/1.12.2</center>
</body>
</html>
```

1.2.5 location语法格式

Syntax: location [= | ~ | ~* | ~~] uri { ... } #uri就是域名后面的信息

优先顺序

= ——精确匹配（不要多余的） 1
 ~ ——区分大小写匹配 3
 ~* ——不区分大小写匹配（grep -i） 3
 ~~ ——优先匹配（优先级） 2
 /AV ——指定要匹配的目录资源 3
 / ——指定匹配站点目录 **默认匹配**
 ! ——表示取反匹配

1. 编写测试文件

```
cat www.conf
server {
    listen      80;
    server_name www.etiantian.org etiantian.org;
    root        html/www;
    location / {
        return 401;
    }
    location = / {
        return 402;
    }
    location /documents/ {
        return 403;
    }
    location ^~ /images/ {
        return 404;
    }
    location ~* \.(gif|jpg|jpeg)$ {
        return 500;
    }
    access_log logs/access_www.log main;
}
```

2. 根据返回状态码，确认优先级

作用：编译以后进行网站运行状态监控

总结取状态码方法：

- 1) curl -I www.etiantian.org/kai/ 2>/dev/null|awk 'NR==1{print \$2}'
- 2) curl -I www.etiantian.org/kai/ -s|awk 'NR==1{print \$2}'
- 3) curl -s -I www.etiantian.org/kai/ -w "%{http_code}\n" -o /dev/null

PS: -I : 显示响应报文起始行和响应头部信息

-o: 将输出内容定向到空

-w: 指定需要输出显示的信息 ("%{http_code}\n" : http状态码)

-s: 不显示错误输出，将错误信息追加到空

表 7-9 不用 URI 及特殊字符组合匹配的顺序说明

顺序	不用 URI 及特殊字符组合匹配	匹配说明
1	location = / {	精确匹配 /
2	location ^~ /images/ {	匹配常规字符串，不做正则匹配检查
3	location ~* \. (gif jpg jpeg) \$ {	正则匹配
4	location /documents/ {	匹配常规字符串，如果有正则，则优先匹配正则
5	location / {	所有 location 都不能匹配后的默认匹配

@51CTO博客

用户请求的 URI	设置的状态码	说 明
当为空或/时	返回 402，即匹配了 location = / { return 402 ; } 和上述官方示例说明表 7-7 匹配(configurationA)一致	= 的精确匹配优先级最高，无论放置的顺序如何，它都将优先被匹配并执行
/index.html 或任意不匹配其它 location 的字符串	返回 401，即匹配了 location / { return 401 ; } 和上述官方示例说明表 7-7 匹配(configurationB)一致	/ 为默认匹配，即如果没有匹配上其他的 location，则最后匹配"默认匹配"的部分
/documents/document.html	返回 403，即匹配了 location /documents/ { return 403 ; } 和上述官方示例说明表 7-7 匹配(configurationC)一致	此部分为路径匹配，即匹配了路由/documents/，注意后面的/documents/1.jpg，这表示没有匹配此处的 location。而是匹配了结尾的 1.jpg
/images/1.gif	返回 404，即匹配了 location ^~ /images/ { return 404 ; }	此部分为路径匹配，但是前面增加了特殊字符^~，所以优先匹配路径，而没有匹配结尾的 1.gif
	} 和上述官方示例说明表 7-7 匹配(configurationD)一致	
/documents/1.jpg	返回 500，即匹配了 location ~* \. (gif jpg jpeg) \$ { return 500 ; } 和上述官方示例说明表 7-7 匹配(configurationE)一致	此部分匹配了 1.jpg，属于扩展名匹配，虽然有/documents/，但还是匹配了扩展名

从以上多个 location 的配置匹配可以看出匹配的优先顺序，具体见表 7-9。

@51CTO博客

1.2.6 rewrite作用

1. 作用： 实现域名地址信息跳转
 用于做伪静态
- www.etiantian.org/kai?edu.html ---动态资源
- www.etiantian.org/kai-edu.html ---伪静态

如何实现类似百度重写域名的功能？
baidu.com ==> www.baidu.com
etiantian.org ==> bbs.etiantian.org

过程:

rewrite指令实践操作一: (错误)

```
[root@web01 extra]# cat bbs.conf
server {
    listen      80;
    server_name bbs.etiantian.org bbs.org;
    rewrite ^/(.*) http://bbs.etiantian.org/$1 permanent;
    root    html/bbs;
    index  index.html index.htm;
}
```

```
[root@web01 extra]# curl -L etiantian.org
```

--- 进行访问跳转追踪

```
curl: (47) Maximum (50) redirects followed
```

```
[root@web01 extra]# curl -Lv etiantian.org
```

--- 会显示无限循环过程

PS: 以上配置进入了无限循环状态

rewrite指令实践操作二: (正确)

```
cat bbs.conf
server {
    listen 80;
    server_name etiantian.org;
    rewrite ^/(.*) http://bbs.etiantian.org/$1 permanent;
}
server {
    listen      80;
    server_name bbs.etiantian.org bbs.org;
    root    html/bbs;
    index  index.html index.htm;
}
```

PS: 以上状态不会循环, 跳转成bbs, 后再次重新匹配时, 是以bbs的域名查找, 这时候就不匹配第一个server_name了, 直接进入下一个server区块

rewrite指令实践操作三: (正确)

```
[root@web01 extra]# cat bbs.conf
server {
    listen      80;
    server_name bbs.etiantian.org bbs.org;
    if ($host ~* "^etiantian.org$") { ---if这里相当于location
        rewrite ^/(.*) http://bbs.etiantian.org/$1 permanent;
    }
    root    html/bbs;
    index  index.html index.htm;
}
```

PS: \$host就是主机头信息, ~*: 不区分大小写, 后面的是以etianitan开头, org结尾。经过rewrite就跳转为bbs, 然后在重新访问, 这个时候就不满足if语句的条件了, 那么就继续读取下一个location

1.2.7 rewrite企业应用场合:

- 可以调整用户浏览的URL, 使其看起来更规范, 合乎开发及产品人员的需求。
- 为了让搜索引擎收录网站内容, 并让用户体验更好, 企业会将动态URL地址伪装成静态地址提供服务。
- 网站换新域名后, 让旧域名的访问跳转到新的域名上, 例如: 让京东的360buy换成了jd.com。
- 根据特殊变量、目录、客户端的信息进行URL跳转等。

1.3 Nginx访问认证

部署过程:

1. 编写配置文件, 加入认证信息

```
auth_basic "kai training";
auth_basic_user_file /application/nginx/conf/htpasswd;
```

```
[root@web01 extra]# cat bbs.conf
server {
    listen      80;
    server_name bbs.etiantian.org bbs.org;
    auth_basic "kai training";
```

```
auth_basic_user_file /application/nginx/conf/htpasswd;
root    html/bbs;
index   index.html index.htm;
}
```

2. 编写认证文件

利用htpasswd命令生成密文密码信息

httpd-tools-2.2.15-59.el6.centos.x86_64 ——安装apache（需要用其中的htpasswd命令）

```
htpasswd -bc /application/nginx/conf/htpasswd kai kail23    ——生成认证文件
chmod 400 /application/nginx/conf/htpasswd
chown www.www /application/nginx/conf/htpasswd
```

PS: -c #创建一个新的密码文件
-b #采用免交互的方式输入用户的密码信

3. 重启服务进行测试

```
nginx -s reload
windows浏览器进行测试
linux系统下进行测试
```

```
[root@web01 conf]# curl -u lidao888:123456 www.dancy.com
web01 www
```

1.3.1 curl命令参数

总结curl命令参数：

```
-v      —— 显示用户访问网站详细报文信息（请求报文 响应报文）
-I      —— 显示响应报文起始行和响应头部信息
-H host: —— 修改请求报文host字段信息
-L      —— 进行访问跳转追踪
-u user:password —— 指定认证用户信息
-o      —— 将输出内容可以指定到空
-w      —— 指定需要输出显示的信息
-s      —— 不显示错误输出，将错误信息追加到空
```

小伙伴们可以关注我的微信公众号：linux运维菜鸟之旅



关注“中国电信天津网厅”公众号，首次绑定可免费领2G流量，为你的学习提供流量！



版权声明：原创作品，如需转载，请注明出处。否则将追究法律责任
