0x00 简介

在上节文章中我们讲了deny和allow来做一些权限控制，但是在实际应用中，权限控制的需求更加复杂，仅靠deny和allow这两个指令不能满足实际需求，所以这篇文章我们讲讲location块来完成更加精细的权限控制匹配；

水分挤完了，开始干货把！ヽ(✿ﾟ▽ﾟ)ノ

0x01 location匹配规则

````

~ # 波浪线表示执行一个正则匹配，区分大小写

~\* # 表示执行一个正则匹配，不区分大小写

^~ # ^~表示普通字符匹配，如果该选项匹配，只匹配该选项，不匹配别的选项，一般用来匹配目录

= # 进行普通字符精确匹配

@ # "@" 定义一个命名的 location块，使用在内部定向时，例如 error\_page, try\_files 该块不能被外部客户端访问，只能被Nginx内部配置指令所访问

````

0x02 location 匹配的优先级

这快姿势点我们需要重点的清楚了解：

有读者在学习 nginx 中的location 会疑惑，nginx是先匹配 “普通的location” 还是先匹配 “正则location” 呢？

解：

在同一个独立的 service 快中，nginx最先匹配的是 “普通的location” 然后在匹配 “正则location”

“普通的location” 分为两种 一种是 “严格精确匹配” 另外一种是 “最大前缀匹配”

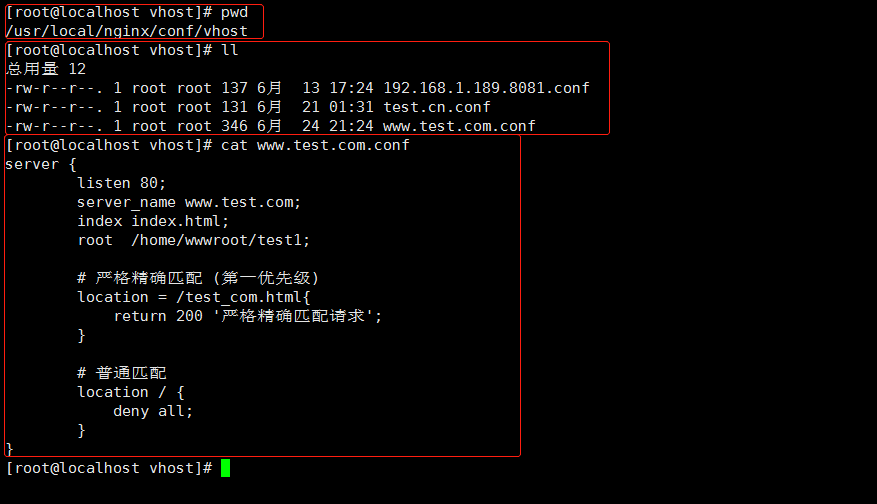
看到这里我们又会生出新的疑问：

“严格精确匹配” “最大前缀匹配” “正则location” 三者的优先级是怎么样的呢？

| **匹配优先级** | |
| --- | --- |
| **模式** | **含义** |
| location = /url | = 表示精确匹配，只有完全匹配上才能生效 (第一优先级) |
| location ^~ /url | ^~ 开头对URL路径进行前缀匹配，并且在正则之前。(第二优先级) |
| location ~ pattern | 开头表示区分大小写的正则匹配 (第三优先级)(两个第三是因为正则表达式谁在前面就比谁的优先级更高) |
| location ~\* pattern | 开头表示不区分大小写的正则匹配 (第三优先级) )(两个第三是因为正则表达式谁在前面就比谁的优先级更高) |
| location /url | 不带任何修饰符，也表示前缀匹配，但是在正则匹配之后 (第四优先级) |
| location / | 通用匹配，任何未匹配到其它location的请求都会匹配到，相当于switch中的default (第五优先级) |

0x03 location 匹配例子

3.1 精准匹配



````

server {

listen 80;

server\_name www.test.com;

index index.html;

root /home/wwwroot/test1;

# 严格精确匹配 (第一优先级)

location = /test\_com.html{

return 200 '严格精确匹配请求';

}

# 普通匹配

location / {

deny all;

}

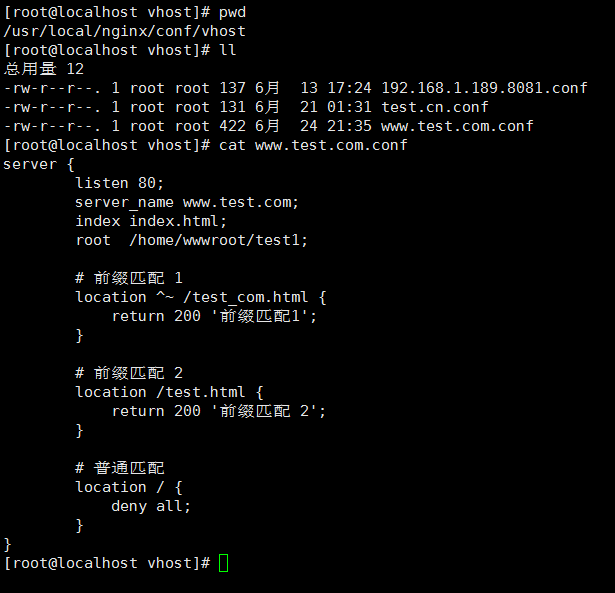
}

````

访问结果：

|  |  |
| --- | --- |
| URI 请求 | 结果 |
| curl http://www.test.com/ | 403 Forbidden |
| curl http://www.test.com/aa | 403 Forbidden |
| curl http://www.test.com/test\_com.html | 输出：严格精确匹配请求 |

3.2 前缀匹配



````

server {

listen 80;

server\_name www.test.com;

index index.html;

root /home/wwwroot/test1;

# 前缀匹配 1

location ^~ /test\_com.html {

return 200 '前缀匹配1';

}

# 前缀匹配 2

location /test.html {

return 200 '前缀匹配 2';

}

# 普通匹配

location / {

deny all;

}

}

````

访问结果：

|  |  |
| --- | --- |
| URI 请求 | 结果 |
| curl http://www.test.com/ | 403 Forbidden |
| curl http://www.test.com/aa | 403 Forbidden |
| curl http://www.test.com/test\_com.html | 输出：前缀匹配1 |
| curl http://www.test.com/test.html | 输出：前缀匹配 2 |

3.3 正则匹配



````

server {

listen 80;

server\_name www.test.com;

index index.html;

root /home/wwwroot/test1;

# 不区分大小写正则匹配

location ~\* \.html$ {

return 200 '不区分大小写正则匹配';

}

# 普通匹配

location / {

deny all;

}

}

````

访问结果：

|  |  |
| --- | --- |
| URI 请求 | 结果 |
| curl http://www.test.com/ | 403 Forbidden |
| curl http://www.test.com/aa | 403 Forbidden |
| curl http://www.test.com/aa.html | 输出：不区分大小写正则匹配 |
| curl http://www.test.com/bb.html | 输出：不区分大小写正则匹配 |
| curl http://www.test.com/cc.html | 输出：不区分大小写正则匹配 |
| curl http://www.test.com/aa.php | 403 Forbidden |