Contents

[Phân biệt alias và root. 1](#_Toc71927248)

[ALIAS + index ON 1](#_Toc71927249)

[Basic authen: 2](#_Toc71927250)

[Error custom page 2](#_Toc71927251)

[Chặn $http\_user\_agent 2](#_Toc71927252)

[Chặn GET POST Method 3](#_Toc71927253)

[NGINX block kết hợp nhiều điều kiện với nhau 3](#_Toc71927254)

[Theo dõi GET real time 4](#_Toc71927255)

[Repsonse by body hoặc bằng file: 4](#_Toc71927256)

[Chặn paramerter request 5](#_Toc71927257)

[Nginx proxy timeout 5](#_Toc71927258)

[Nginx php-fpm fastCgi timeout 5](#_Toc71927259)

# Phân biệt alias và root.

Root không sử dụng trong location, alias mới được dùng trong location.

Alias ví dụ như location /app/ thì alias trỏ cũng phải có / cuối cùng (alias /var/html/app/;)

For example, if you want to define an alias directive for a location block like the following:

|  |  |
| --- | --- |
| 1  2  3 | location /images/ {      alias /data/w3/images/;  } |

You should consider defining a root directive for that location block:

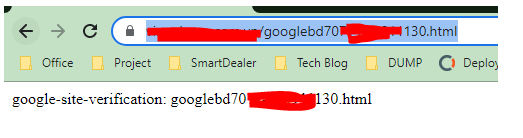
|  |  |
| --- | --- |
| 1  2  3 | location /images/ {      root /data/w3;  } |

Chú ý bug alias: location không được thiếu / ở cuối cùng, nếu ko sẽ bị lỗi path traversal (off by slash) <https://systemweakness.com/common-nginx-misconfiguration-leads-to-path-traversal-d58701e997bc> .

Bài hướng dẫn : <https://serverfault.com/questions/278351/nginx-root-versus-alias-for-serving-single-files>

Ví dụ

location =/googlebd707141f3911130.html { root /tmp/googlebd707141f3911130/; }



# ALIAS + index ON

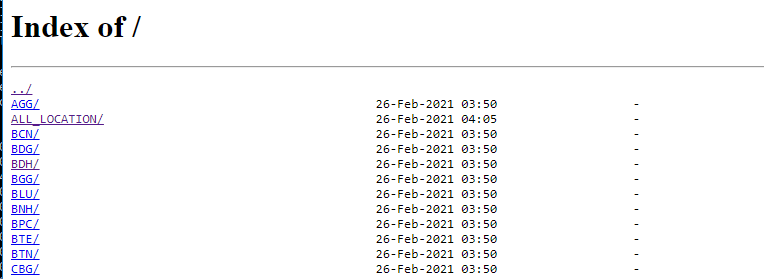
Hiển thị file and folder

location / {

alias /home/project/scripts/filter\_location/report\_location/;

autoindex on;

}



Đọc hiểu

# Basic authen:

# Error custom page

<https://github.com/HttpErrorPages/HttpErrorPages#nginx-integration>

server {

listen 80;

server\_name localhost;

root /var/www;

index index.html;

location / {

try\_files $uri $uri/ =404;

# add one directive for each http status code

error\_page 400 /ErrorPages/HTTP400.html;

error\_page 401 /ErrorPages/HTTP401.html;

error\_page 402 /ErrorPages/HTTP402.html;

error\_page 403 /ErrorPages/HTTP403.html;

error\_page 404 /ErrorPages/HTTP404.html;

error\_page 500 /ErrorPages/HTTP500.html;

error\_page 501 /ErrorPages/HTTP501.html;

error\_page 502 /ErrorPages/HTTP502.html;

error\_page 503 /ErrorPages/HTTP503.html;

}

# redirect the virtual ErrorPages path the real path

location /ErrorPages/ {

alias /var/ErrorPages/;

internal;

}

# Chặn $http\_user\_agent

1. #Chặn Agent **null**
2. **if** ($http\_user\_agent = "") {
3. **return** 403;
4. access\_log /opt/nginx/logs/myvnpt\_api.banned\_null.log modlog;
5. }

<https://hocvps.com/block-user-agent-nginx/>

1. # **case** sensitive matching
2. **if** ($http\_user\_agent ~ (Antivirx|Arian)) {
3. **return** 403;
4. }
6. # **case** insensitive matching
7. **if** ($http\_user\_agent ~\* (netcrawl|npbot|malicious)) {
8. **return** 403;
9. }

* case sensitive matching: phân biệt chữ in hoa, chữ in thường
* case insensitive matching: không phân biệt in hoa, in thường

Để tìm được header cần filter, tất nhiên bạn sẽ phải phân tích file access.log trước.

Để test kết quả bạn có thể dùng lệnh wget kèm theo option --user-agent, ví dụ:

wget --spider --user-agent "malicious bot" http://domain.com

Đây là đoạn code mình dùng để block request spam comment các bạn có  thể tham khảo thêm:

1. #Block Spam comment
2. location ~\* /wp-comments-post\.php$ {
3. **if** ($http\_user\_agent ~\* "x11; linux i686; rv:17" ) {
4. **return** 403;
5. }
6. }

# Chặn GET POST Method

<https://www.stevenrombauts.be/2017/04/deny-http-request-methods-in-nginx/>

location /api {

limit\_except POST {

deny all;

}

}

limit\_except POST GET {

allow 192.168.1.1/24;

deny all;

}

limit\_except POST {

auth\_basic "Restricted access";

auth\_basic\_user\_file /etc/nginx/htpasswd;

}

# NGINX block kết hợp nhiều điều kiện với nhau

* Khai báo một biến $BLOCK với giá trị rỗng.
* Mỗi điều kiện match sẽ Append một giá trị vào biến $BLOCK
* Nếu kết quả cuối cùng của biến $BLOCK chứa đầy đủ các giá trị mong muốn (ra chữ TRUE )thì DROP request đấy.

Cụ thể:

1. location = / {
2. set $BLOCK "";
3. if ($arg\_add\_to\_wishlist ~ "[0-9]+") {
4. set $BLOCK "${BLOCK}T";
5. }
7. if ($http\_user\_agent = "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/36.0.1985.143 Safari/537.36") {
8. set $BLOCK "${BLOCK}R";
9. }
11. if ($server\_protocol = "HTTP/1.0") {
12. set $BLOCK "${BLOCK}U";
13. }
15. if ($request\_method = "GET") {
16. set $BLOCK "${BLOCK}E";
17. }
19. if ($BLOCK = "TRUE") {
20. return 444;
21. }
22. [...]
23. }

# Theo dõi GET real time

# pip install ngxtop

$ ngxtop

Top 10 requested path with status 404:

$ ngxtop top request\_path --filter 'status == 404'

Top 10 requests with highest total bytes sent

$ ngxtop --order-by 'avg(bytes\_sent) \* count'

Top 10 remote address, e.g., who's hitting you the most

$ ngxtop --group-by remote\_addr

Print requests with 4xx or 5xx status, together with status and http referer

$ ngxtop -i 'status >= 400' print request status http\_referer

Average body bytes sent of 200 responses of requested path begin with 'foo':

$ ngxtop avg bytes\_sent --filter 'status == 200 and request\_path.startswith("foo")'

Analyze apache access log from remote machine using 'common' log format

$ ssh remote tail -f /var/log/apache2/access.log | ngxtop -f common

Nhiều cái hay

<https://www.tecmint.com/category/web-servers/nginx/>

<https://www.tecmint.com/category/monitoring-tools/>

# Repsonse by body hoặc bằng file:

**By text include nginx**

location /r/ {

return 200 "{Heloworld}";

}

**By File:**

location /r/ {

index r.json;

alias /home/user/media/json/;

}

# Chặn paramerter request

<https://serverfault.com/questions/811912/can-nginx-location-blocks-match-a-url-query-string>

1. server {
2. #... common definitions such as server, root
4. location / {
5. error\_page 418 = @queryone;
6. error\_page 419 = @querytwo;
7. error\_page 420 = @querythree;
9. **if** ( $query\_string = "service=git-receive-pack" ) { **return** 418; }
10. **if** ( $args ~ "service=git-upload-pack" ) { **return** 419; }
11. **if** ( $arg\_somerandomfield = "somerandomvaluetomatch" ) { **return** 420; }
13. # do the remaining stuff
14. # ex: try\_files $uri =404;
16. }
18. location @queryone {
19. # do stuff when queryone matches
20. }
22. location @querytwo {
23. # do stuff when querytwo matches
24. }
26. location @querythree {
27. # do stuff when querythree matches
28. }
29. }

# Nginx proxy timeout

proxy\_connect\_timeout 600;

proxy\_send\_timeout 600;

proxy\_read\_timeout 600;

send\_timeout 600;

# Nginx php-fpm fastCgi timeout

<https://rtcamp.com/tutorials/php/increase-script-execution-time/>

Sửa php.ini

max\_execution\_time = 300

Sửa php-fpm

vim /etc/php-fpm.d/www.conf

request\_terminate\_timeout = 300

sửa nginx:

location ~ \.php$ {

include /etc/nginx/fastcgi\_params;

fastcgi\_pass unix:/var/run/php5-fpm.sock;

**fastcgi\_read\_timeout 300;**

}

# Nginx route base on client IP

<https://www.cyberciti.biz/faq/nginx-redirect-backend-traffic-based-upon-client-ip-address/>

Cách 1:

## apachereadonly backend ##

upstream apachereadonly {

server 10.10.11.10:8011;

server 10.10.11.11:8011;

server 10.10.11.12:8011;

ip\_hash;

}

## apachereadwrite backend ##

upstream apachereadwrite {

server 10.10.11.13:8011;

}

location / {

proxy\_set\_header Accept-Encoding "";

proxy\_set\_header Host $http\_host;

proxy\_set\_header X-Forwarded-By $server\_addr:$server\_port;

proxy\_set\_header X-Forwarded-For $remote\_addr;

proxy\_set\_header X-Forwarded-Proto $scheme;

proxy\_set\_header X-Real-IP $remote\_addr;

## default backend

proxy\_pass http://apachereadonly;

## send traffic to apachereadwrite backend if ip is 1.2.3.4 ##

if ( $remote\_addr ~\* 1.2.3.4 ) {

proxy\_pass http://apachereadwrite;

}

proxy\_next\_upstream error timeout invalid\_header http\_500 http\_502 http\_503 http\_504;

}

Cách 2:

## define default backend and backend for special remote addresses

map $remote\_addr $backend {

default apachereadonly;

1.2.3.4 apachereadwrite;

}

location / {

proxy\_set\_header Accept-Encoding "";

proxy\_set\_header Host $http\_host;

proxy\_set\_header X-Forwarded-By $server\_addr:$server\_port;

proxy\_set\_header X-Forwarded-For $remote\_addr;

proxy\_set\_header X-Forwarded-Proto $scheme;

proxy\_set\_header X-Real-IP $remote\_addr;

proxy\_pass http://$backend;

proxy\_next\_upstream error timeout invalid\_header http\_500 http\_502 http\_503 http\_504;

}

# Nginx Change version

[**https://www.cyberciti.biz/tips/linux-unix-bsd-nginx-webserver-security.html**](https://www.cyberciti.biz/tips/linux-unix-bsd-nginx-webserver-security.html)

vim src/http/ngx\_http\_header\_filter\_module.c

Sửa thành

static char ngx\_http\_server\_string[] = "Server: TuanDA Web Server" CRLF;

static char ngx\_http\_server\_full\_string[] = "Server: TuanDA Web Server" CRLF;