How to configure nginx to auto redirect to the main hostname?

Asked 12 years, 3 months ago Modified 3 years, 9 months ago Viewed 54k times

25

Each nginx config can act for a wide range of domains but I want to auto-redirect requests to the first domain name (the official one).

```
server {
    server_name a.example.com b.example.com;
}
```

I want that if someone enters `b.example.com/some`, to go directly to `a.example.com/some`

nginx http-redirect

Share Improve this question Follow

asked May 10, 2012 at 10:19

sorin

168k ● 188 ● 557 ● 833

3 Answers

Sorted by: Highest score (default)

24

This is pretty much the same thing as the GOOD example for http://wiki.nginx.org/Pitfalls#Server_Name. That is, you should use two servers:

```
server {
  server_name b.example.com;
  return 301 $scheme://a.example.com$request_uri;

  # For pre-0.8.42 installations:
  # rewrite ^ $scheme://a.example.com$request_uri? permanent;
}

server {
  server_name a.example.com;
  # Do stuff
}
```

Share Improve this answer Follow

answered May 10, 2012 at 11:04

kolbyjack

18.2k ● 5 ● 49 ● 35

2 `server_name *.example.com;` will catch any subdomain including `www` and `porno` ™ and `whatsoever`, if someone looking for redirecting anything to main domain. For http->https redirection you can include anything by using `server_name .example.com` - it will redirect both example.com and *.example.com. – Dmitry Verhoturov May 10, 2012 at 12:42

@DmitryVerhoturov when using `.example.com` what's a good way to indicate that variations such as `www.` should be redirected to `example.com`? Without using the performance-reducing "if" conditions? – PKHunter Feb 13, 2017 at 10:16

2 @PKHunter, `.example.com` is a best way to cover `*.example.com`, [here is the doc](). – Dmitry Verhoturov Feb 13, 2017 at 14:17

But what happens if you are using ssl and both sites have the same certificate? This will break SNI surely? – symcbean Sep 12, 2019 at 20:13

15

To do this in a single server block, you can use an if and the [`$server_name` variable]():

```
server_name primary.tld secondary.tld;
if ($host != $server_name) {
    rewrite ^ $scheme://$server_name permanent;
}
```

Or, to keep any query parameters:

```
server_name primary.tld secondary.tld;
if ($host != $server_name) {
    rewrite ^/(.*) $scheme://$server_name/$1 permanent;
}
```

Here, [`$server_name`]() refers to primary server name, which is the first name in the `server_name` directive, while [`$host`]() refers to the hostname given in the HTTP request.

Note that the `if` statement in nginx configuration does not always do what you would expect and its use is discouraged by some. See also [https://www.nginx.com/resources/wiki/start/topics/depth/ifisevil/]()

This answer was inspired by [this answer to another question]() which uses a similar aproach.

Share Improve this answer Follow

edited Apr 13, 2017 at 12:13
Community Bot
1 ● 1

answered Nov 3, 2016 at 21:33
Matthijs Kooijman
2,757 ● 26 ● 34

8

Combined version of @kolbyjack and @Matthijs answers with one `server` block. This config will redirect all requests with `Host` header different from *example.com* and process only *example.com* requests.

```
server {
    server_name example.com a.example.com b.example.com;

    if ($host != $server_name) {
        return 301 $scheme://$server_name$request_uri;
    }

    # processing requests to $server_name (example.com) only
    ...
}
```

Share Improve this answer Follow

answered Nov 14, 2020 at 14:41

Alexander Ushakov
5,379 ● 3 ● 29 ● 55