

# Best way to configure Nginx SSL + force HTTP to redirect to HTTPS + force www to non-www on Serverpilot free plan (by using Nginx configuration file only)



September 29, 2016  126.8k NGINX DIGITALOCEAN CONFIGURATION MANAGEMENT WORDPRESS UBUNTU

By: Jannemankanerwatvan

I have set up a Wordpress website running on my Serverpilot/Digital Ocean droplet. I have succesfully managed to:

- Manually Install a PositiveSSL certificate (from Namecheap)
- Change the APPNAME.conf file to force HTTP to redirect to HTTPS and to force www to non-www

So everything works fine but I'm not sure if I have configured the Serverpilot Nginx configuration file correctly. My questions are:

**What I did:** I have implemented the 301 redirects rules in Serverpilots Nginx configuration file (APPNAME.conf) so the redirects are handled in the most efficient way possible by the server.\*\*

**My question:** is this indeed the best/fastest way?

**What I did:** The config files should be protected from serverpilot updates, so when serverpilot updates it's system, no files get overwritten, no errors will take place. I followed this [Serverpilot Article](#)

**My question:** Is my configuration protected from being overwritten by serverpilot?

**What I did:** Reduce the amount of redirects. All requests (<http://example.com>, <http://www.example.com>, <https://www.example.com>) should be redirected to <https://example.com> in max. one redirect step. I have tested my website in [HTTP Status Code Checker](#). All redirects are done in max. one step.

**My question:** When I test for instance [Google.nl](http://google.nl) in this tool I see that the redirect from <http://google.nl> to <https://www.google.nl> is performed in two steps. Why is that? Should I also perform a redirect from <http://www.example.com> to <https://example.com> in two 301 redirects?

## My Serverpilot Nginx configuration

### Step 1: Installing SSL certificat

I have installed the PositiveSSL certificate on the server following step 1 - 4 of this tutorial:

<https://www.blogmehow.com/how-to-manually-install-ssl-on-serverpilot-free-plan-1331/>

### Step 2: Create 3 serverblocks in the APPNAME.custom.conf file

I have renamed the APPNAME.conf file to APPNAME.custom.conf file in this `/etc/nginx-sp/vhosts.d` opened it and changed it so it has three serverblocks. One to redirect http to https, one to redirect the https www-name to no-www, and one to actually handle the requests.:

```
server {
    listen 80;
    listen [::]:80;
    server_name example.com www.example.com;
    return 301 https://example.com$request_uri;
}

server {
    listen 443 ssl;
    listen [::]:443 ssl;
    server_name www.example.com;

    ssl on;

    ssl_certificate /root/certs/APPNAME/APPNAME_nl.chained.crt;
    ssl_certificate_key /root/certs/APPNAME/ssl.key;

    ssl_session_timeout 1d;
    ssl_session_cache shared:SSL:20m;
    ssl_session_tickets off;
```

```

ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
ssl_prefer_server_ciphers on;

ssl_ciphers 'ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES256-G

ssl_stapling on;
ssl_stapling_verify on;

ssl_trusted_certificate /root/certs/APPNAME/APPNAME_n1.chained.crt;

root    /srv/users/serverpilot/apps/APPNAME/public;

access_log  /srv/users/serverpilot/log/APPNAME/APPNAME_nginx.access.log  main;
error_log   /srv/users/serverpilot/log/APPNAME/APPNAME_nginx.error.log;

proxy_set_header    Host                $host;
proxy_set_header    X-Real-IP           $remote_addr;
proxy_set_header    X-Forwarded-For    $proxy_add_x_forwarded_for;
proxy_set_header    X-Forwarded-SSL on;
proxy_set_header    X-Forwarded-Proto $scheme;

include /etc/nginx-sp/vhosts.d/APPNAMEd/*.nonssl_conf;
include /etc/nginx-sp/vhosts.d/APPNAME.d/*.conf;

return 301 https://example.com$request_uri;
}

server {
listen 443 ssl;
listen [::]:443 ssl;
server_name example.com;

ssl on;

ssl_certificate /root/certs/APPNAME/APPNAME_n1.chained.crt;
ssl_certificate_key /root/certs/APPNAME/ssl.key;

ssl_session_timeout 1d;
ssl_session_cache shared:SSL:20m;
ssl_session_tickets off;

ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
ssl_prefer_server_ciphers on;

```

```

ssl_ciphers 'ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES256-G

ssl_stapling on;
ssl_stapling_verify on;

ssl_trusted_certificate /root/certs/APPNAME/APPNAME_n1.chained.crt;

root    /srv/users/serverpilot/apps/APPNAME/public;

access_log  /srv/users/serverpilot/log/APPNAME/APPNAME_nginx.access.log  main;
error_log   /srv/users/serverpilot/log/APPNAME/APPNAME_nginx.error.log;

proxy_set_header    Host                $host;
proxy_set_header    X-Real-IP           $remote_addr;
proxy_set_header    X-Forwarded-For    $proxy_add_x_forwarded_for;
proxy_set_header    X-Forwarded-SSL on;
proxy_set_header    X-Forwarded-Proto $scheme;

include /etc/nginx-sp/vhosts.d/APPNAME.d/*.nonssl_conf;
include /etc/nginx-sp/vhosts.d/APPNAME.d/*.conf;
}

```

I hope someone can take the time to have a look at my configuration and let me know how I can optimize my setup even more taking into account my goals.

Kind regards,

Jan

**\*\*** According to what I have found in [This article](#), this is the most efficient and fastest way to do a 301 redirect: The best way to accomplish this is using three server blocks: one to redirect http to https, one to redirect the https www-name to no-www, and one to actually handle requests. The reason for using extra server blocks instead of ifs is that server selection is performed using a hash table, and is very fast. Using a server-level if means the if is run for every request, which is wasteful. Also, capturing the requested uri in the rewrite is wasteful, as nginx already has this information in the \$uri and \$request\_uri variables (without and with query string, respectively).

1 comment

 [jakeunltd](#) August 10, 2018

You are amazing. Worked exactly how I wanted it to the first time around.

---

[Log In to Comment](#)

## 6 Answers



[pbritto](#) January 27, 2017

Thanks a lot for sharing this setup! I was struggling with a nginx config to redirect **http://** to **https://** and then **www.domain.com** to **domain.com**. But after I saw your setup I decided to try it out with a few tweaks and now my vhost looks roughly like this:

```
server {
    listen 80;
    server_name www.domain.com domain.com;

    # redirects both www and non-www to https
    return 301 https://domain.com$request_uri;
}

server {
    listen 443 ssl http2;
    server_name www.domain.com;

    # redirects www to non-www. wasn't work for me without this server block
    return 301 $scheme://domain.com$request_uri;
}

server {

    ## SSL settings

    listen 443 ssl http2 default_server;
    listen [::]:443 ssl http2 default_server;

    server_name domain.com;

    include snippets/ssl-domain.com.conf;
    include snippets/ssl-params.conf;

    # other vhost configuration
```

}

Maybe it could help you shorten your code as well. You should try to make a separate file for your SSL configuration and just include it with the **include** as I did in my 3rd server block. It's cleaner and easier to maintain.

[Reply](#)

^ mfissehaye July 5, 2018

0 This saved a lot of hours of work. Thank you.

^  
0

[ryanpq](#) MOD October 6, 2016

Thanks for sharing this. I am not aware of a good technical reason for doing the redirect in two steps but would guess that Google uses some more complex logic to determine where to redirect a visitor in order to catch edge cases and make Google usable for everyone. Google also likely started redirecting to www before they implemented the redirect to https.

[Reply](#)

^  
0

[bakhtiyorh](#) October 13, 2016

anyone who uses serverpilot, please join to the unofficial slack discussion community hub, for sharing experiences about doing missing (premium) features in free plan 😊

<http://serverpilot-slack.herokuapp.com/>

[Reply](#)

^  
0

[chxc](#) June 21, 2017

Using a https error code 497, it is possible to do it in two server blocks.

This is tested and working on my site.

```
server {
    listen 80;
    server_name www.domain.com domain.com;

    # redirects both www and non-www to ssl port with http (NOT HTTPS, forcing error
    return 301 http://domain.com$request_uri;
}
```

```
server {  
    listen 433 ssl http2 default_server;  
    listen [::]:443 ssl http2 default_server;  
  
    server_name domain.com;  
    error_page 497 https://domain.com$request_uri;  
  
    include snippets/ssl-domain.com.conf;  
    include snippets/ssl-params.conf;  
  
    # other vhost configuration  
}
```

Hope this helps!

[Reply](#)



[rootat3l1ot](#) May 26, 2018

[deleted]

[Reply](#)



[rootat3l1ot](#) May 26, 2018

Hi, I'm not able to redirect www version of my site to <https://indreport.com> using .htaccess.

[Reply](#)

## Have another answer? Share your knowledge.

Answer a question...

[Log In to Answer](#)

---

## Related Questions

[My domain does not work...](#)

[I've Paid 50\\$, but Account locked now ?! Why?](#)

[How can I include the www. in my domain?](#)

[Permission problems running WordPress on Nginx](#)

[Connecting to Casandra from remote machine](#)

## Introducing Projects on DigitalOcean

Organize your resources according to how you work.

[READ MORE](#)



Copyright © 2018 DigitalOcean™ Inc.

[Community](#) [Tutorials](#) [Questions](#) [Projects](#) [Tags](#) [Newsletter](#) [RSS](#) 

---

[Distros & One-Click Apps](#) [Terms, Privacy, & Copyright](#) [Security](#) [Report a Bug](#) [Write for DOnations](#) [Shop](#)



