

Install "nginx"

```
root@Bryan:/home/bryan# sudo apt install nginx
Reading package lists... Done
Building dependency tree
Reading state information... Done
nginx is already the newest version (1.14.0-0ubuntu1.10).
The following packages were automatically installed and are no longer required:
  bridge-utils dns-root-data dnsmasq-base libhogweed4 libidn11 libnettle6 ubuntu-fan
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 1 not upgraded.
root@Bryan:/home/bryan#
```

Then create the conf file

```
server {
listen      80;
server_name example.com www.example.com;
index       index.php;
root        /var/www/example.com/public    # fallback for index.php
location / {
try_files $uri $uri/ /index.php?$query_string;
}location /blog {
proxy_pass http://blog.domain.com;proxy_http_version      1.1;
proxy_cache_bypass      $http_upgrade;

# Proxy headers
proxy_set_header Upgrade      $http_upgrade;
proxy_set_header Connection   "upgrade";
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-Proto $scheme;
proxy_set_header X-Forwarded-Host $host;
proxy_set_header X-Forwarded-Port $server_port;

# Proxy timeouts
proxy_connect_timeout      60s;
proxy_send_timeout         60s;
proxy_read_timeout         60s;
}
}
```

Then test the code

```
root@Bryan:/home/bryan# sudo nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
root@Bryan:/home/bryan#
```

Apache

```
root@Bryan:/home/bryan# echo "Listen 8080" | sudo tee /etc/apache2/ports.conf
Listen 8080
root@Bryan:/home/bryan#
```

The change the port for the server

```
<VirtualHost *:8080>
-   # The ServerName directive sets the request scheme, hostname and port that
-   # the server uses to identify itself. This is used when creating
-   # redirection URLs. In the context of virtual hosts, the ServerName
-   # specifies what hostname must appear in the request's Host: header to
-   # match this virtual host. For the default virtual host (this file) this
-   # value is not decisive as it is used as a last resort host regardless.
-   # However, you must set it for any further virtual host explicitly.
-   #ServerName www.example.com
-
-   ServerAdmin webmaster@localhost
-   DocumentRoot /var/www/html
-
-   # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
-   # error, crit, alert, emerg.
-   # It is also possible to configure the loglevel for particular
-   # modules, e.g.
-   #LogLevel info ssl:warn
-
-   ErrorLog ${APACHE_LOG_DIR}/error.log
-   CustomLog ${APACHE_LOG_DIR}/access.log combined
-
-   # For most configuration files from conf-available/, which are
-   # enabled or disabled at a global level, it is possible to
-   # include a line for only one particular virtual host. For example the
-   # following line enables the CGI configuration for this host only
```

Now let's check the connection with the server's

```
root@Bryan:/home/bryan# sudo netstat -tlnp
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:80             0.0.0.0:*               LISTEN      614/nginx: master p
tcp6       0      0 :::8080                :::*                    LISTEN      1177/apache2
tcp6       0      0 :::80                  :::*                    LISTEN      614/nginx: master p
root@Bryan:/home/bryan#
```

The next thing to do is create the servers: ping and pong

```
root@Bryan:/home/bryan# sudo mkdir -v /var/www/ping.net /var/www/pong.io
mkdir: created directory '/var/www/ping.net'
mkdir: created directory '/var/www/pong.io'
root@Bryan:/home/bryan#
```

Create both servers

```
root@Bryan:/home/bryan# sudo nano /etc/apache2/sites-available/ping.net.conf
root@Bryan:/home/bryan# sudo nano /etc/apache2/sites-available/pong.io.conf
```

Then enable the servers:

```
Enabling site ping.net.
To activate the new configuration, you need to run:
  service apache2 reload
root@Bryan:/home/bryan# sudo a2ensite pong.io
Enabling site pong.io.
To activate the new configuration, you need to run:
  service apache2 reload
root@Bryan:/home/bryan#
```

Then create the servers for nginx

```
server {  
    listen 80;  
    server_name ping.net www.ping.net pong.io www.pong.io;  
  
    location / {  
        proxy_pass http://127.0.0.1:8080;  
        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-Proto $scheme;  
    }  
}
```

Check the configuration file

```
root@Bryan:/home/bryan# sudo nginx -t  
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok  
nginx: configuration file /etc/nginx/nginx.conf test is successful  
root@Brvan:/home/brvan# _
```

And check the server

PHP Variables

Variable	Value
\$_SERVER["USER"]	www-data
\$_SERVER["HOME"]	/var/www
\$_SERVER["ORIG_SCRIPT_NAME"]	/php-fcgi
\$_SERVER["ORIG_PATH_TRANSLATED"]	/var/www/foobar.net/info.php
\$_SERVER["ORIG_PATH_INFO"]	/info.php
\$_SERVER["ORIG_SCRIPT_FILENAME"]	/usr/lib/cgi-bin/php-fcgi
\$_SERVER["SCRIPT_NAME"]	/info.php
\$_SERVER["REQUEST_URI"]	/info.php
\$_SERVER["QUERY_STRING"]	no value
\$_SERVER["REQUEST_METHOD"]	GET
\$_SERVER["SERVER_PROTOCOL"]	HTTP/1.0
\$_SERVER["GATEWAY_INTERFACE"]	CGI/1.1
\$_SERVER["REDIRECT_URL"]	/info.php
\$_SERVER["REMOTE_PORT"]	54614
\$_SERVER["SCRIPT_FILENAME"]	/var/www/foobar.net/info.php
\$_SERVER["SERVER_ADMIN"]	[no address given]
\$_SERVER["CONTEXT_DOCUMENT_ROOT"]	/usr/lib/cgi-bin/php-fcgi
\$_SERVER["CONTEXT_PREFIX"]	/php-fcgi
\$_SERVER["REQUEST_SCHEME"]	http
\$_SERVER["DOCUMENT_ROOT"]	/var/www/foobar.net
\$_SERVER["REMOTE_ADDR"]	159.89.164.40
\$_SERVER["SERVER_PORT"]	80
\$_SERVER["SERVER_ADDR"]	159.89.164.40
\$_SERVER["SERVER_NAME"]	foobar.net
\$_SERVER["SERVER_SOFTWARE"]	Apache/2.4.29 (Ubuntu)
\$_SERVER["SERVER_SIGNATURE"]	<address>Apache/2.4.29 (Ubuntu) Server at foobar.net Port 80</address>
\$_SERVER["PATH"]	/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
\$_SERVER["HTTP_ACCEPT"]	*/*
\$_SERVER["HTTP_USER_AGENT"]	Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:59.0) Gecko/20100101 Firefox/59.0
\$_SERVER["HTTP_CONNECTION"]	close
\$_SERVER["HTTP_X_FORWARDED_PROTO"]	http
\$_SERVER["HTTP_X_FORWARDED_FOR"]	██████████
\$_SERVER["HTTP_X_REAL_IP"]	██████████

<https://www.digitalocean.com/community/tutorials/como-configurar-nginx-como-servidor-web-y-proxy-inverso-para-apache-en-un-servidor-ubuntu-18-04-es>

<https://www.digitalocean.com/community/tutorials/como-configurar-nginx-como-servidor-web-y-proxy-inverso-para-apache-en-un-servidor-ubuntu-18-04-es>

<https://kinsta.com/es/blog/proxy-inverso/>