

Nginx Task

install Nginx

```
sudo apt update  
sudo apt install nginx
```

Check for service

```
sudo systemctl status nginx
```

HTML App with Nginx

```
sudo nano /var/www/html/index.html
```

```
<!DOCTYPE html>  
<html>  
<head><title>My HTML App</title></head>  
<body>  
<h1>Hello from Nginx HTML App!</h1>  
</body>  
</html>
```

HTML

Install PHP with Nginx (2 ways)

first way (A) Using PHP-FPM (Socket)

- Step 1: install php-fpm

```
sudo apt install php-fpm
sudo nano /etc/nginx/sites-available/default
```

- Step 2: inside `server { ... }`, add: note that 8.x depends in your version

```
location ~ \.php$ {
    include snippets/fastcgi-php.conf;
    fastcgi_pass unix:/run/php/php8.3-fpm.sock;
}
```

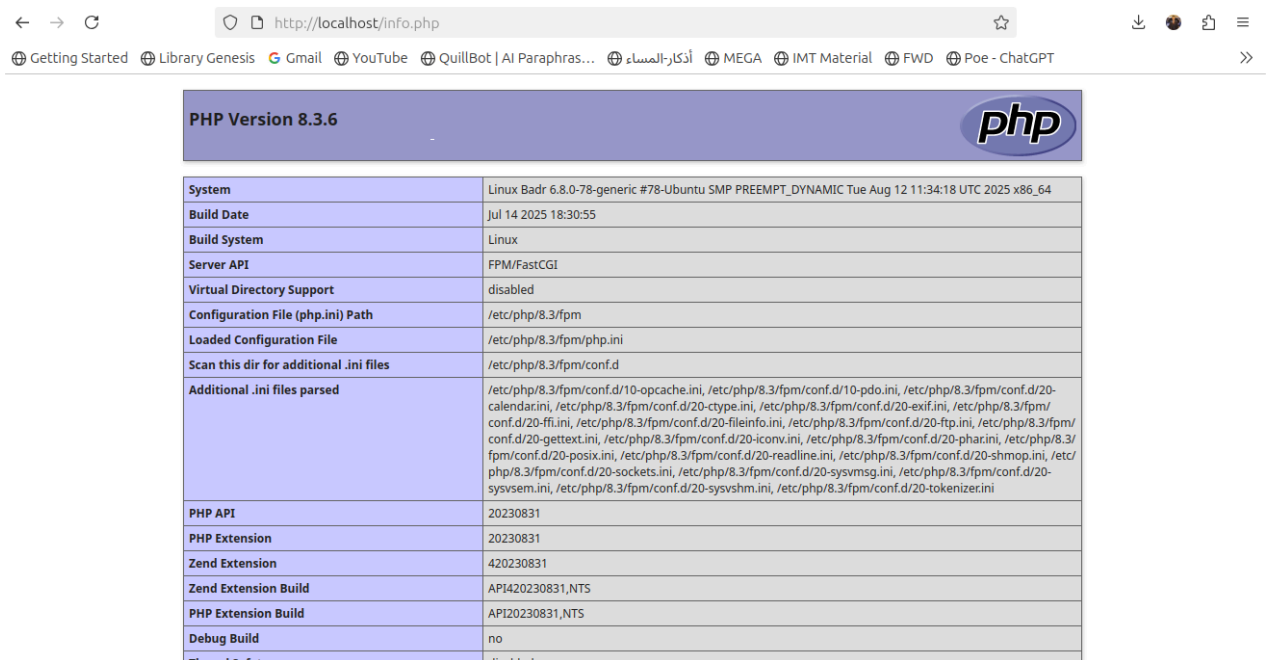
- Step 3: Test config and restart NGINX

```
nginx -t
systemctl restart nginx
```

- Step 4: create PHP page

```
echo "<?php phpinfo(); ?>" | sudo tee /var/www/html/info.php
```

- Step 5: Hit `http://localhost/info.php`



System	Linux Bahr 6.8.0-78-generic #78-Ubuntu SMP PREEMPT_DYNAMIC Tue Aug 12 11:34:18 UTC 2025 x86_64
Build Date	Jul 14 2025 18:30:55
Build System	Linux
Server API	FPM/FastCGI
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/8.3/fpm
Loaded Configuration File	/etc/php/8.3/fpm/php.ini
Scan this dir for additional .ini files	/etc/php/8.3/fpm/conf.d
Additional .ini files parsed	/etc/php/8.3/fpm/conf.d/10-opcache.ini, /etc/php/8.3/fpm/conf.d/10-pdo.ini, /etc/php/8.3/fpm/conf.d/20-calendar.ini, /etc/php/8.3/fpm/conf.d/20-ctype.ini, /etc/php/8.3/fpm/conf.d/20-exif.ini, /etc/php/8.3/fpm/conf.d/20-ffi.ini, /etc/php/8.3/fpm/conf.d/20-fileinfo.ini, /etc/php/8.3/fpm/conf.d/20-ftp.ini, /etc/php/8.3/fpm/conf.d/20-gettext.ini, /etc/php/8.3/fpm/conf.d/20-iconv.ini, /etc/php/8.3/fpm/conf.d/20-phar.ini, /etc/php/8.3/fpm/conf.d/20-posix.ini, /etc/php/8.3/fpm/conf.d/20-readline.ini, /etc/php/8.3/fpm/conf.d/20-shmop.ini, /etc/php/8.3/fpm/conf.d/20-sockets.ini, /etc/php/8.3/fpm/conf.d/20-sysmsg.ini, /etc/php/8.3/fpm/conf.d/20-syssem.ini, /etc/php/8.3/fpm/conf.d/20-sysshm.ini, /etc/php/8.3/fpm/conf.d/20-tokenizer.ini
PHP API	20230831
PHP Extension	20230831
Zend Extension	420230831
Zend Extension Build	API420230831,NTS
PHP Extension Build	API20230831,NTS
Debug Build	no
Thread Safety	disabled

Second Way (B) Using PHP-FPM (TCP Port)

- Step 1: install php-fpm

```
sudo apt install php-fpm
sudo nano /etc/nginx/sites-available/default
```

- Step 2: Edit [PHP-FPM pool](#):

```
sudo nano /etc/php/8.3/fpm/pool.d/www.conf
```

- Step 3: change listen in this file

```
listen = /run/php/php8.3-fpm.sock -> old -> socket
listen = 127.0.0.1:2020 -> new -> TCP port
```

- Step 4: restart [PHP-FPM](#)

```
sudo systemctl restart php8.1-fpm
```

- Step 5: inside [server { ... }](#), add: note that 8.x depends in your version

```
location ~ /\.php$ {
    include snippets/fastcgi-php.conf;
    fastcgi_pass 127.0.0.1:2020;
}
```

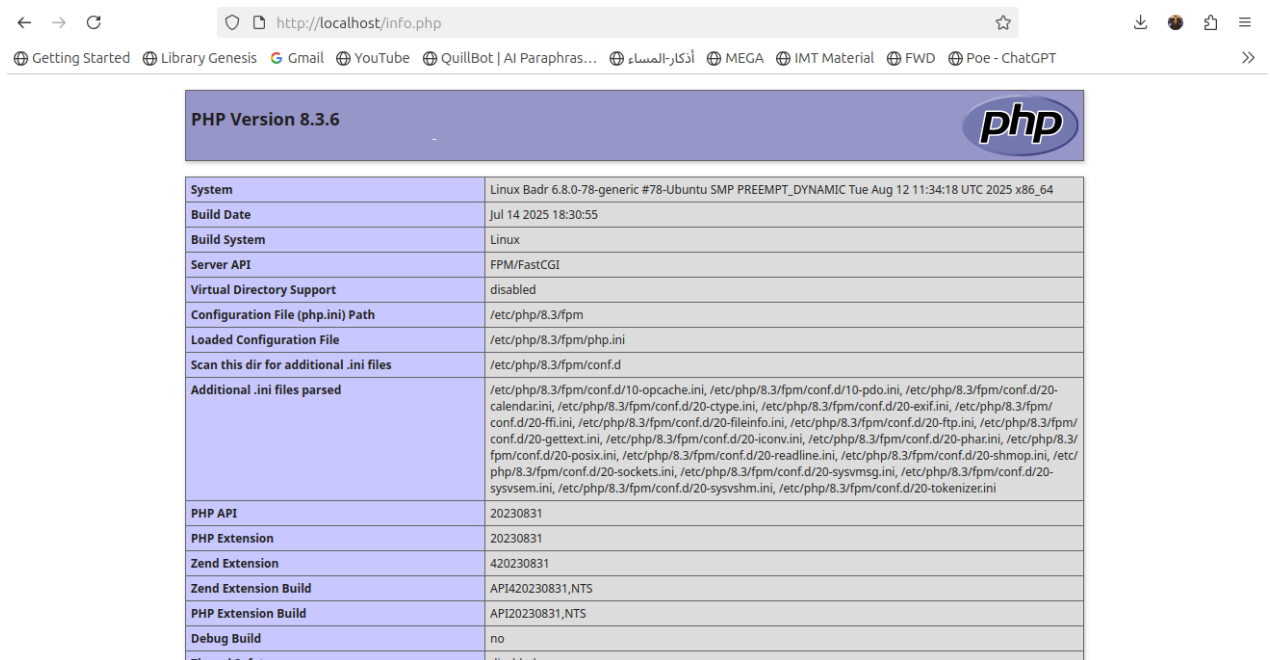
- Step 6: Test config and restart NGINX

```
nginx -t
systemctl restart nginx
```

- Step 7: create PHP page

```
echo "<?php phpinfo(); ?>" | sudo tee /var/www/html/info.php
```

- Step 8: Hit <http://localhost/info.php>



PHP Version 8.3.6	
System	Linux Bdr 6.8.0-78-generic #78-Ubuntu SMP PREEMPT_DYNAMIC Tue Aug 12 11:34:18 UTC 2025 x86_64
Build Date	Jul 14 2025 18:30:55
Build System	Linux
Server API	FPM/FastCGI
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/8.3/fpm
Loaded Configuration File	/etc/php/8.3/fpm/php.ini
Scan this dir for additional .ini files	/etc/php/8.3/fpm/conf.d
Additional .ini files parsed	/etc/php/8.3/fpm/conf.d/10-opcache.ini, /etc/php/8.3/fpm/conf.d/10-pdo.ini, /etc/php/8.3/fpm/conf.d/20-calendar.ini, /etc/php/8.3/fpm/conf.d/20-ctype.ini, /etc/php/8.3/fpm/conf.d/20-exif.ini, /etc/php/8.3/fpm/conf.d/20-ffi.ini, /etc/php/8.3/fpm/conf.d/20-fileinfo.ini, /etc/php/8.3/fpm/conf.d/20-ftp.ini, /etc/php/8.3/fpm/conf.d/20-gettext.ini, /etc/php/8.3/fpm/conf.d/20-iconv.ini, /etc/php/8.3/fpm/conf.d/20-phar.ini, /etc/php/8.3/fpm/conf.d/20-posix.ini, /etc/php/8.3/fpm/conf.d/20-readline.ini, /etc/php/8.3/fpm/conf.d/20-shmop.ini, /etc/php/8.3/fpm/conf.d/20-sockets.ini, /etc/php/8.3/fpm/conf.d/20-sysmsg.ini, /etc/php/8.3/fpm/conf.d/20-syssem.ini, /etc/php/8.3/fpm/conf.d/20-sysshm.ini, /etc/php/8.3/fpm/conf.d/20-tokenizer.ini
PHP API	20230831
PHP Extension	20230831
Zend Extension	420230831
Zend Extension Build	API420230831,NTS
PHP Extension Build	API20230831,NTS
Debug Build	no

Photo of configuration for both ways

```
GNU nano 7.2                                default *
{
    listen 3080;
    root /var/www/html;
    index site1.html;
    location ~ /\.php$ {
        include snippets/fastcgi-php.conf;
        fastcgi_pass 127.0.0.1:2020;
    }
}
server
{
    listen 3081;
    root /var/www/html;
    index site2.html;
    location ~ /\.php$ {
        include snippets/fastcgi-php.conf;
        fastcgi_pass unix:/run/php/php8.3-fpm.sock;
    }
}

^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute  ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify  ^/ Go To Line
```

Edit Nginx Cache Config

- Step 1: Open Config

```
sudo nano /etc/nginx/nginx.conf
```

- Step 2: inside `http{ ... }` add:

```
proxy_cache_path /var/cache/nginx levels=1:2 keys_zone=my_cache:10m  
max_size=1g inactive=60m use_temp_path=off;
```

- Step 3: in your site config:

```
location / {  
    proxy_cache my_cache;  
    proxy_pass http://127.0.0.1:9000;  
    proxy_cache_valid 200 1m;  
}
```

- Step 4: Test config and restart NGINX

```
nginx -t  
systemctl restart nginx
```

Loadbancer Task

- We will use this example:
 - Loadbalancer Example: We have 2 sites that are site1.html and site2.html and need to make loadbalancer for these 2 sites.
- Step 1: create the 2 html files in /var/www/html

```
echo "<h1>Hello from site 1</h1>" | sudo tee /var/www/html/site1.html  
echo "<h1>Hello from site 2</h1>" | sudo tee /var/www/html/site2.html
```

- Step 2: Create the 2 sites server configurations

```
sudo tee /etc/nginx/sites-available/default <<EOF  
server {  
    listen 8081;  
    root /var/www/html/;  
    index site1.html;  
}  
server {  
    listen 8082;
```

```
    root /var/www/html/;  
    index site2.html;  
}  
EOF
```

- Step 3: Create the loadbalancer configurations

```
sudo tee /etc/nginx/sites-available/loadbalancer <<EOF  
upstream sites {  
    server 127.0.0.1:8081;  
    server 127.0.0.1:8082;  
}  
  
server {  
    listen 80;  
  
    location / {  
        proxy_pass http://sites;  
        proxy_set_header Host $host;  
        proxy_set_header X-Real-IP $remote_addr;  
    }  
}  
EOF
```

- Step 4: Link loadbalancer configurations file in sites enable

```
sudo ln -s /etc/nginx/sites-available/loadbalancer /etc/nginx/sites-enabled/
```

- Step 5: Test config and restart NGINX

```
nginx -t  
systemctl restart nginx
```

- Final result:

```
badr@Badr:/etc/nginx/sites-available$ curl localhost
<h1>Hello From Site2</h1>
badr@Badr:/etc/nginx/sites-available$ curl localhost
<h1>Hello From Site2</h1>
badr@Badr:/etc/nginx/sites-available$ curl localhost
<h1>Hello From Site2</h1>
badr@Badr:/etc/nginx/sites-available$ curl localhost
<h1>Hello From site1</h1>
badr@Badr:/etc/nginx/sites-available$ curl localhost
<h1>Hello From Site2</h1>
badr@Badr:/etc/nginx/sites-available$ curl localhost
<h1>Hello From site1</h1>
badr@Badr:/etc/nginx/sites-available$ curl localhost
<h1>Hello From Site2</h1>
badr@Badr:/etc/nginx/sites-available$ curl localhost
<h1>Hello From site1</h1>
badr@Badr:/etc/nginx/sites-available$ curl localhost
<h1>Hello From Site2</h1>
badr@Badr:/etc/nginx/sites-available$ curl localhost
<h1>Hello From site1</h1>
badr@Badr:/etc/nginx/sites-available$
```