

# Install NGINX, PHP, MySQL, SSL & WordPress on Ubuntu



## # Install and Configure Nginx

- Nginx Open Source is free and open-source software
- **Step to install Nginx**

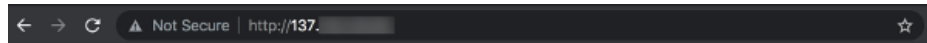
```
$ sudo apt update && sudo apt upgrade && sudo apt autoremove //Updatepackage
```

```
$ sudo apt -y install nginx //Install Nginx package
```

```
$ sudo systemctl status nginx //Check that Nginx service has been running
```

Check the default web page is accessible

- ▼ Open your browser and type **http://{PUBLIC-IP}**
- . Ensure that you can see the default Nginx web page similar to picture below.



## Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to [nginx.org](http://nginx.org).  
Commercial support is available at [nginx.com](http://nginx.com).

*Thank you for using nginx.*

### 5. Configure the firewall to allow HTTP/S traffic

```
$ sudo ufw status //Check firewall status
```

▼ If you see “**Status: active**” in the command output but did not see any entry for HTTP/S ports, you need to configure the firewall to allow HTTP and HTTPS traffic.

```
▼ $ sudo ufw allow proto tcp to 0.0.0.0/0 port 80
```

```
$ sudo ufw allow proto tcp to 0.0.0.0/0 port 443
```

```
$ sudo ufw status // if not active than enable it
```

### 6. Create a configuration for the website

```
▼ $ sudo vi /etc/nginx/sites-available/your domain name
```

```
server {
    listen 80;
    root /var/www/your domain folder;
    index index.html index.htm;
    server_name domain name;

    location / {
        try_files $uri $uri/ =404;
    }
}
```

### 7. Apply the configuration symbolic link

```
▼ $ sudo ln -s /etc/nginx/sites-available/your domain name /etc/nginx/sites-enabled/your domain name
```

### 8. We test the configuration once again and reload Nginx configuration.

```
▼ $ sudo nginx -t
```

```
$ sudo systemctl reload nginx
```

## # Install and Config mySQL

MySQL is the most popular open-source relational database management system.

### 1. Installing MySQL on Ubuntu

```
▼ $ sudo apt update && sudo apt upgrade && sudo apt autoremove //Update apt package
```

```
▼ $ sudo apt install mysql-server //MySQL install
```

### 2. To verify that the MySQL server is running,

```
▼ $ sudo systemctl status mysql
```

### 3. Securing MySQL

▼ `$ sudo mysql`

▼ Run the following SQL query:

```
mysql > ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY 'SetRootPasswordHere';
mysql > FLUSH PRIVILEGES;
exit
```

4. Then run the following command to secure it:

▼ `$ sudo mysql_secure_installation`

▼ It ask for validate plugin password — Press y|Y for Yes, any other key for No:

▼ first time **no** select then select **yes**

### 5. Login As root

▼ `sudo mysql -u root -p // add password`

```
mysql > SHOW DATABASES; // list database
mysql > CREATE DATABASE DB; //create database
mysql > SHOW DATABASES; // list database with created
mysql > USE DB; // change db DB ENTER
mysql > SHOW TABLES; // RESULT SET
mysql > exit;
```

## # Install php and php-fpm

PHP is a general-purpose scripting language geared toward web development.

#### 1. installing php

▼ `$ sudo apt update && sudo apt upgrade && sudo apt autoremove //Update apt package`

▼ `$ sudo apt install php-fpm php-mysql`

▼ `$ sudo apt install php8.0-{fpm,gd,mbstring,mysql,xml,opcache,cli,zip,soap,intl,bcmath,curl} php-ssh2`

#### 2. check php-fpm status

▼ `sudo systemctl status php8.1-fpm`

#### 3. check php is fully install or not,create info.php file

▼ `$ sudo nano /var/www/html/info.php`

```
<?php
phpinfo();

?>
```

4. Run again in browser with info.php and check mysql.ini into web page

## # Domain name and SSL

SSL is usually meant to be used to switch an existing HTTP site to work in HTTPS.

so, we can use **CERTBOT** for SSL.

### 1. installing certbot with nginx

```
▼ $ sudo apt update && sudo apt upgrade && sudo apt autoremove //Update apt package
▼ $ sudo apt install python3-certbot-nginx
▼ $ sudo certbot --nginx -d abc.com // domain
$ sudo certbot renew --dry-run //Test automatic renewal
```

### 2. Confirm that Certbot worked

▼ To confirm that your site is set up properly, visit <https://yourwebsite.com/> in your browser and look for the lock icon in the URL bar.

## AFTER SSL CREATED NGINX CONFIG FILE ,

```
server {
    root /var/www/DOMAIN-FOLDER;
    error_log /var/log/nginx/DOMAIN.com-error.log error;
    error_page 404 /index.php;

    # Add index.php to the list if you are using PHP
    index index.php index.html index.htm index.nginx-debian.html;
    server_name DOMAIN.com;

    location / {
        try_files $uri $uri/ /index.php?$query_string;
    }
    # pass PHP scripts to FastCGI server
    #
    location ~ /\.php$ {
        fastcgi_split_path_info ^(.+\.php)(/.+)$;
        fastcgi_pass unix:/var/run/php/php8.1-fpm.sock;
        #include default.d/php.conf;
        fastcgi_index index.php;
        include fastcgi_params;
        include fastcgi.conf;
    }
    # deny access to .htaccess files, if Apache's document root
    # concurs with nginx's one
    #
    location ~ /\.ht {
        deny all;
    }
    server_tokens off; // add for nginx version
    listen [::]:443 ssl ipv6only=on; # managed by Certbot
    listen 443 ssl; # managed by Certbot
    ssl_certificate /etc/letsencrypt/live/DOMAIN.com/fullchain.pem; # managed by Certbot
    ssl_certificate_key /etc/letsencrypt/live/DOMAIN.com/privkey.pem; # managed by Certbot
    include /etc/letsencrypt/options-ssl-nginx.conf; # managed by Certbot
    ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem; # managed by Certbot
}
```

## # Install Wordpress

WordPress is a free and open-source content management system (CMS) written in hypertext preprocessor (PHP) language and paired with a MySQL database with supported HTTPS. Features include a plugin architecture and a template system, referred to within WordPress as "Themes".

## 1. installing wordpress

▼ `$ sudo apt update && sudo apt upgrade && sudo apt autoremove` //Update apt package

▼ `$ wget https://wordpress.org/latest.tar.gz` // Download wordpress cms

▼ `$ tar -xzf latest.tar.gz` // Extract It

## 2. Add it specific folder in **/var/www/** inside

3. `$ cd /wordpress` //go to inside wordpress folder


## 4. you can copy **wp-config-sample.php** into **wp-config.php**

## 5. change it content

## 6. first download wordpress secret key using curl

▼ `$ curl -s https://api.wordpress.org/secret-key/1.1/salt/`

▼ or **copy** -

 <https://api.wordpress.org/secret-key/1.1/salt/>

## #WP-CONFIG.PHP FILE

```
<?php
/** Enable W3 Total Cache */
define('WP_CACHE', true); // Added by W3 Total Cache

/**
 * The base configuration for WordPress
 *
 * The wp-config.php creation script uses this file during the installation.
 * You don't have to use the web site, you can copy this file to "wp-config.php"
 * and fill in the values.
 *
 * This file contains the following configurations:
 *
 * * Database settings
 * * Secret keys
 * * Database table prefix
 * * ABSPATH
 *
 * @link https://wordpress.org/support/article/editing-wp-config-php/
 *
 * @package WordPress
 */

// ** Database settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define( 'DB_NAME', 'dbname' );

/** Database username */
define( 'DB_USER', 'dbuser' );

/** Database password */
define( 'DB_PASSWORD', 'password' );

/** Database hostname */
define( 'DB_HOST', 'localhost' );

/** Database charset to use in creating database tables. */
define( 'DB_CHARSET', 'utf8mb4' );
```

```

/** The database collate type. Don't change this if in doubt. */
define( 'DB_COLLATE', '' );

/**#@+
 * Authentication unique keys and salts.
 *
 * Change these to different unique phrases! You can generate these using
 * the {@link https://api.wordpress.org/secret-key/1.1/salt/ WordPress.org secret-key service}.
 *
 * You can change these at any point in time to invalidate all existing cookies.
 * This will force all users to have to log in again.
 *
 * @since 2.6.0
 */
define( 'AUTH_KEY',          'B]R3o>bT4W^yf*Q!Bz^Ebx/#*c8e>a%uDqAM*Zhwu7[n4#y7Q iLp8VW5TX)+^$^' );
define( 'SECURE_AUTH_KEY',  '%R|Dtn;>}g }&4^Pyfq(5f!uEs0Uu(0Y%/ ,jv%x<2ZSy2Gqsg<gJTfs$P+xjFjp' );
define( 'LOGGED_IN_KEY',    'J3t+KJ@C(5R/28FxDA}G%%!EDCO/6#=9P/%nUm]GQ!fM$P&W?M}`?'GB=Vwdjq{=' );
define( 'NONCE_KEY',        'K55_dV!rT_LUV)1{7:0+N.7m}co*?qEHZSS1)Kl#bDG5L6sg8u>9}~/rVD1+=Bfp' );
define( 'AUTH_SALT',        '&`H_W*`XlQi}gLS%y<Wz(BUu,wfR{fz(nw5Sw23e(seQOD!$:r{!;a>PM;PY-Q[X' );
define( 'SECURE_AUTH_SALT', '4+LVWU/M`Ww-4r8mf?Gw+{2|>]a)p^zR88a[9#q<v>* >=zB3+[ld`)75@wGfCN' );
define( 'LOGGED_IN_SALT',   '*q[<]T]/P-0]lu~8._9nTG5+ KP>(W#gB712H0Gc1h2n!$UFj =`nF{N0R:xtvZG' );
define( 'NONCE_SALT',       '{kYTs<I`D;Uw04) `E<t-DuY+%y`,WFZZ))uH1uz3bBHG[]YR8 hw)zI/2Adn5TQ' );

/**#@-*/

/**
 * WordPress database table prefix.
 *
 * You can have multiple installations in one database if you give each
 * a unique prefix. Only numbers, letters, and underscores please!
 */
$table_prefix = 'ntd_';

/**
 * For developers: WordPress debugging mode.
 *
 * Change this to true to enable the display of notices during development.
 * It is strongly recommended that plugin and theme developers use WP_DEBUG
 * in their development environments.
 *
 * For information on other constants that can be used for debugging,
 * visit the documentation.
 *
 * @link https://wordpress.org/support/article/debugging-in-wordpress/
 */
define( 'WP_DEBUG', false );

/* Add any custom values between this line and the "stop editing" line. */


/* That's all, stop editing! Happy publishing. */

/** Absolute path to the WordPress directory. */
if ( ! defined( 'ABSPATH' ) ) {
    define( 'ABSPATH', __DIR__ . '/' );
}

/** Sets up WordPress vars and included files. */
require_once ABSPATH . 'wp-settings.php';

```

## #WP-CONFIG.PHP FILE 2ND EXAMPLE

```

<?php
# Database Configuration
define( 'DB_NAME', 'db' );

```

```

define( 'DB_USER', 'forge' );
define( 'DB_PASSWORD', 'avccc' );
define( 'DB_HOST', 'localhost' );
define( 'DB_HOST_SLAVE', 'localhost' );
define( 'DB_CHARSET', 'utf8mb4' );
define( 'DB_COLLATE', '' );

$table_prefix = 'ntd_';

# Security Salts, Keys, Etc
define( 'AUTH_KEY',          'L0{ |F_yhQp|qt.xSfLKL*}wT%7fpeE7z}qac{enx<kL[yeE&R+xNnQ+FSfEm02'});
define( 'SECURE_AUTH_KEY',    'a5dX#r-HWC`},i[;|(e7RUH#(|Kg3}Wv}eqnNCiP)ow?/UPZnw2,,!JyM+nRnz3,') );
define( 'LOGGED_IN_KEY',       '=>eUEu-.9`lkzKv^AN?0_j9HsFCu2B#V+9?fP}%jVWGSRY6AY73)-Yb{+H44,-}V') );
define( 'NONCE_KEY',          '-6$mqh/e]-D&PqRX0p:b@M+`81D+]fx`p{7M5c9M29,+^Kq3>|#uBG+vXmo]I7Kt') );
define( 'AUTH_SALT',          '/T053n9~`%-3:!3yEuA)L_rpy1H7-?k){QU*FCy-u0Z+@:7m(GH;q!C_x~!;e+ux') );
define( 'SECURE_AUTH_SALT',    '3]CiEGi9<qmKyP52q|J0:{2_lTa/-/Q_rXLQ.Q<t_}82-;%l;~Zbo#h&3cz2,00|') );
define( 'LOGGED_IN_SALT',      'Qpn:5M}n9TL_HauaZ|4f1a[MZ@MvFP^#f*_j3zZ6tgUf-=9FL9|*Y>?i.=3dg-N-') );
define( 'NONCE_SALT',          'v?F(pLu?<Uld-6)8-oI*AS[;=n]40Ers+?s_g9f::t.|N#m<,~xpYXJ:zWZ5j5vh') );

# Localized Language Stuff

define( 'MYSQL_CLIENT_FLAGS', MYSQLI_CLIENT_SSL ); # for ssl wp-config

define( 'WP_CACHE', TRUE );

define( 'WP_AUTO_UPDATE_CORE', false );

define( 'PWP_NAME', 'domain/dir' );

define( 'FS_METHOD', 'direct' );

define( 'FS_CHMOD_DIR', 0775 );

define( 'FS_CHMOD_FILE', 0664 );

define( 'WPE_APIKEY', 'f300f943646bd82b8ed8b374bf278074f985b18e' );

define( 'WPE_CLUSTER_ID', '141012' );

define( 'WPE_CLUSTER_TYPE', 'pod' );

define( 'WPE_ISP', true );

define( 'WPE_BPOD', false );

define( 'WPE_RO_FILESYSTEM', false );

define( 'WPE_LARGEFS_BUCKET', 'largefs.wpengine' );

define( 'WPE_SFTP_PORT', 2222 );

define( 'WPE_SFTP_ENDPOINT', '' );

define( 'WPE_LBMASTER_IP', '' );

define( 'WPE_CDN_DISABLE_ALLOWED', true );

define( 'DISALLOW_FILE_MODS', FALSE );

define( 'DISALLOW_FILE_EDIT', FALSE );

define( 'DISABLE_WP_CRON', false );

define( 'WPE_FORCE_SSL_LOGIN', false );

define( 'FORCE_SSL_LOGIN', false );

/*SSLSTART*/ if ( isset($_SERVER['HTTP_X_WPE_SSL']) && $_SERVER['HTTP_X_WPE_SSL'] ) $_SERVER['HTTPS'] = 'on'; /*SSELEND*/

```

```

define( 'WPE_EXTERNAL_URL', false );

define( 'WP_POST_REVISIONS', FALSE );

define( 'WPE_WHITELABEL', 'wpengine' );

define( 'WP_TURN_OFF_ADMIN_BAR', false );

define( 'WPE_BETA_TESTER', false );

umask(0002);

$wpengine_cdn_uris=array ( );

$wpengine_no_cdn_uris=array ( );

$wpengine_content_regexs=array ( );

$wpengine_all_domains=array ( 0 => 'domain', 1 => 'domain', );

$wpengine_varnish_servers=array ( 0 => 'pod-141012', );

$wpengine_special_ips=array ( 0 => 'ip-address', );

$wpengine_netdna_domains=array ( );

$wpengine_netdna_domains_secure=array ( );

$wpengine_netdna_push_domains=array ( );

$wpengine_domain_mappings=array ( );

$memcached_servers=array ( 'default' => array ( 0 => 'unix:///tmp/memcached.sock', ), );

# WP Engine ID

# WP Engine Settings

define('WP_SITEURL', 'https://domain');
define('WP_HOME', 'https://domain');

# That's It. Pencils down
if ( !defined('ABSPATH') )
    define('ABSPATH', __DIR__ . '/');
require_once(ABSPATH . 'wp-settings.php');

```

7. `$ sudo chmod -R 777 /var/www/dir-domain/ //give permission`

## # Run wordpress particular domain name like,

WordPress › Installation

You appear to have already installed WordPress. To reinstall please clear your old database tables first.

<https://wp.postbox25.in/wp-admin/install.php>

▼ select language and create new admin user and password and login .

--OTHER .....SECURITY PURPOSE



## #wordpress security plugin,

- Let's take a peek at this list of popular WordPress security plugins:

Go to plugin menu into dashboard and search plugin list for install or active

1. [Wordfence](#)
2. [All in One WP Security & Firewall](#)
3. [WPScan](#)

## # Install apf-firewall on Ubuntu,

Advanced Policy Firewall, or **APF**, is a firewall sometimes seen on Liquid Web's servers. It is basically an interface to iptables, which is the standard interface to managing network ports on Linux machines.

1. install APF-FIREWALL

```
▼ $ sudo apt update && sudo apt upgrade && sudo apt autoremove //Update apt package
$ sudo apt-get -y install apf-firewall
```

2. Opening a port in the apf firewall

```
$ vim /etc/apf/conf.apf
```

- ▼ Within this file the file that needs to be changed starts with IG\_TCP\_CPORTS= and looks like this:

```
IG_TCP_CPORTS="20, 21, 22, 25, 53, 80, 110, 143, 443, 465, 993, 995, 2082,
2083, 2084, 2086, 2087, 2095, 2096, 3306, 3784, 7786, 30052"
```

3. After setup APF-firewall

Starting, stopping, and restarting apf can be easily done via the command line:

```
$ apf -s This will start apf if it is not running.
$ apf -r This will restart apf.
$ apf -f This will stop apf and flush all rules from the firewall.
```

## # if you want Object Caching, Page Caching, and Other Speed Optimizations, Than

Configure Redis Object Cache & Nginx FastCGI Page Cache for WordPress on Ubuntu 20.04

In the previous chapter, I walked you through the process of configuring Nginx to serve your WordPress sites over HTTPS on your Linux server. However, we need to do more if we want our sites to feel snappy. In this chapter I will guide you through the process of caching a WordPress site.

<https://spinupwp.com/hosting-wordpress-yourself-server-monitoring-caching/>



## # php curl also best example...

CURL stands for Client URL. It is a Linux Terminal command which is used to transferring data from one server to another server.

```
$ sudo apt-get install php-curl  
$ php -m | grep curl
```

## # php xml for xml-data...

```
$ sudo apt-get install php-xml
```