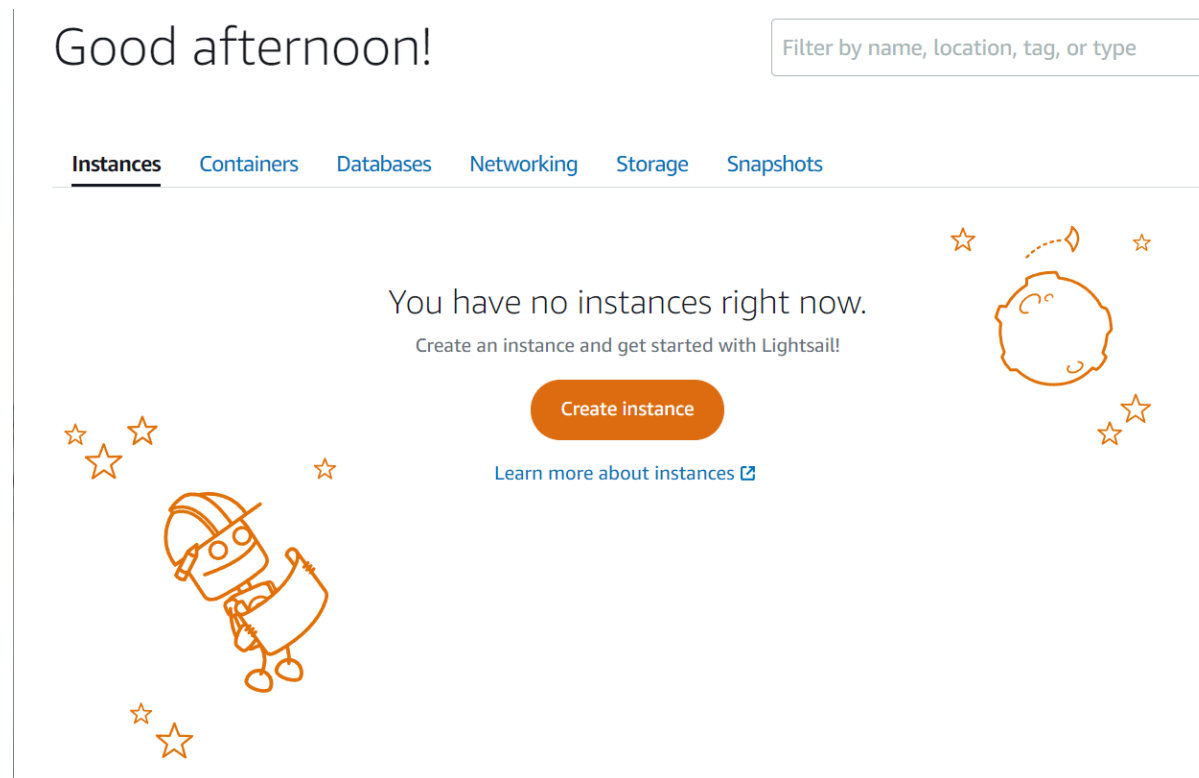
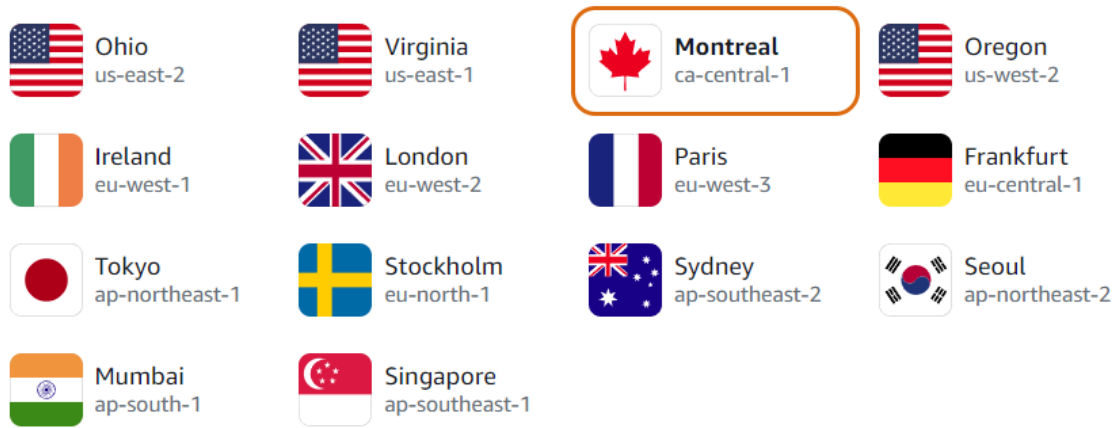


How I created cheapest Wordpress

1. Create lightsail instance



Go to the Lightsail homepage and click “Create instance”. If you don’t have an AWS account, follow the link to create the account first.

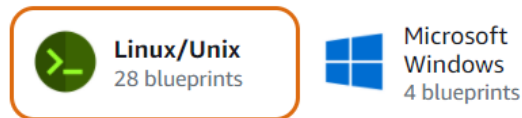


Select an Availability Zone [?](#)

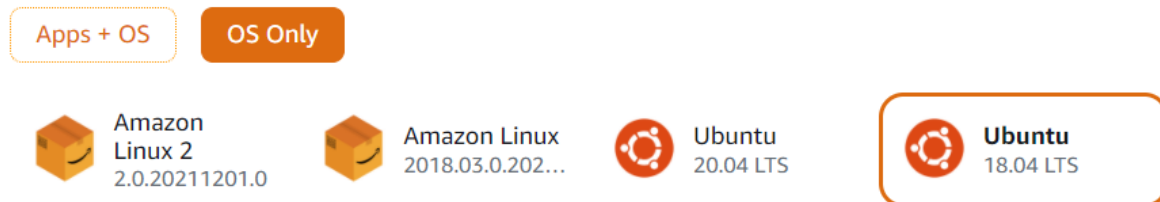


Pick your instance image [?](#)

Select a platform



Select a blueprint



Select the region where is nearest from your location. For the instance image, You could just select Wordpress in Apps + OS but I selected Ubuntu 18.04 LTS OS Only option because I want to setup Wordpress from the scratch.

Ubuntu 18.04 LTS - Bionic. Lean, fast and powerful, Ubuntu Server delivers services reliably, predictably and economically. It is the perfect base on which to build your instances. Ubuntu is free and will always be, and you have the option to get support and Landscape from Canonical.

Learn more about Ubuntu on the [AWS Marketplace](#) .

By using this image, you agree to the provider's [End User License Agreement](#) .

OPTIONAL

You can add a shell script that will run on your instance the first time it launches.

[+ Add launch script](#)

You will connect to your instance using the **default** SSH key.

[Change SSH key pair](#)

Automatic snapshots create a backup image of your instance and attached disks on a daily schedule.

Enable Automatic Snapshots

Choose your instance plan [?](#)

Click "Change SSH key pair" if you select the platform and blueprint

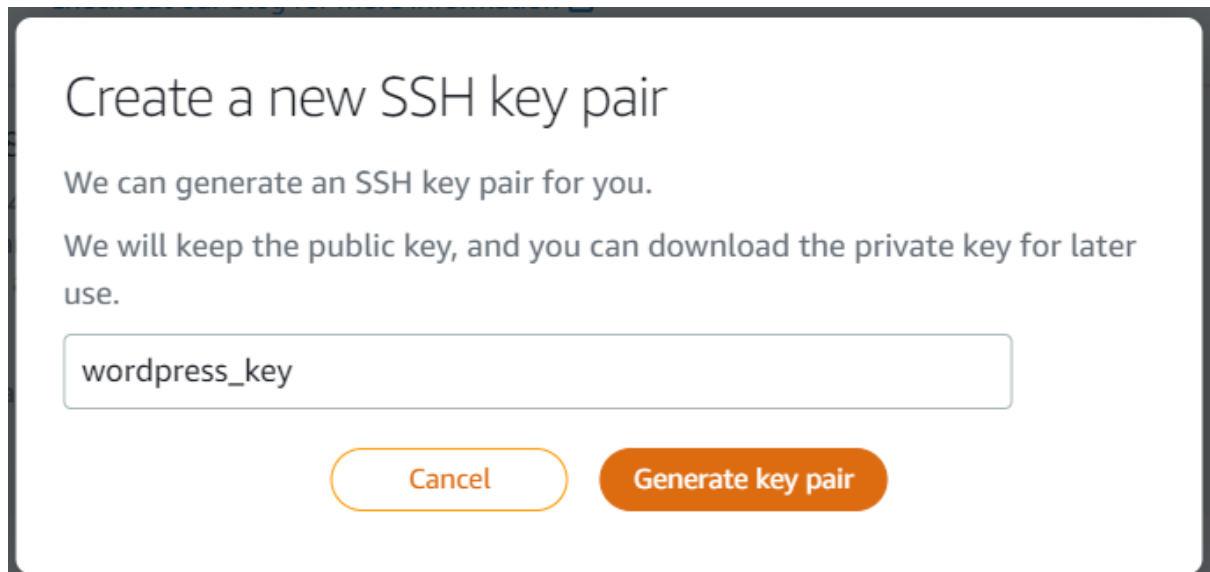
SSH key pair manager [?](#)

Select, create, or upload the key pair you would like to use to SSH into your instance.

[Learn more about SSH keys](#)

[Create New +](#) [Upload New](#) 

Click "Create New" and click "Create".



Click Generate key pair and you will download it. (Save it in safe place)

SSH key pair manager [?](#)

Select, create, or upload the key pair you would like to use to SSH into your instance.

[Learn more about SSH keys](#)

Create New [+](#) Upload New [↗](#)

- Default key [?](#) Download [↓](#)
- wordpress_key ✕

Automatic snapshots create a backup image of your instance and attached disks on a daily schedule.

Enable Automatic Snapshots

Choose your instance plan [?](#)

New! Check out our new 16 GB and 32 GB RAM bundles!

Sort by: [Price per month](#) [Memory](#) [Processing](#) [Storage](#) [Transfer](#)

	First 3 months free!	First 3 months free!	First 3 months free!		
	\$3.5	\$5	\$10	\$20	\$40
	USD	USD	USD	USD	USD
	\$3.50 USD	\$5 USD	\$10 USD	\$20 USD	\$40 USD
	512 MB	1 GB	2 GB	4 GB	8 GB
	1 vCPU	1 vCPU	1 vCPU	2 vCPUs	2 vCPUs
	20 GB SSD	40 GB SSD	60 GB SSD	80 GB SSD	160 GB SSD
	1 TB	2 TB	3 TB	4 TB	5 TB
					Price per month
					Memory
					Processing
					Storage
					Transfer

Select the key pair that you just created and choose your instance plan. You could choose \$3.5 plan if you don't use heavy memory consuming plugins or multi WordPress websites.

Identify your instance

Your Lightsail resources must have unique names.

 ×

TAGGING OPTIONS

Use tags to filter and organize your resources in the Lightsail console. Key-value tags can also be used to organize your billing, and to control access to your resources.

[Learn more about tagging.](#)

Key-only tags [?](#)

+ Add key-only tags

Key-value tags [?](#)





+ Add key-value tag

Create instance

Click "Create instance" and wait for the instance be running

Connect your project!

Networking resources allow you to specify how users and outside services connect to your Lightsail resources. They can be used to specify routing of internet traffic, speed up the delivery of your content, improve redundancy, and increase the capacity of your resources.

 <p>Static IP A static IP is a fixed, public IP address that you can attach to an instance. Learn more about static IPs</p> <p>Create static IP</p>	 <p>Distribution A content delivery network (CDN) distribution speeds up the delivery of your content to your users around the world. Learn more about distributions</p> <p>Create distribution</p>
 <p>Load balancer A load balancer adds redundancy and increases capacity by distributing traffic to multiple instances. Learn more about load balancers</p> <p>Create load balancer</p>	 <p>DNS zone A domain name system (DNS) zone defines subdomains for your domain, and routes traffic to your resources. Learn more about DNS zones</p> <p>Create DNS zone</p>

Click “Networking” and “Create static IP”

Static IP location




You are creating this static IP in **Montreal, all zones** (ca-central-1)

 [Change AWS Region and Availability Zone](#)

Attach to an instance

Attaching a static IP replaces that instance's dynamic IP address.

Static IP addresses can only be attached to instances in the same region.



Identify your static IP

Your Lightsail resources must have unique names.

Static IP addresses are free only while attached to an instance.
You can manage five at no additional cost.

Create

Select your instance and create.

2. Connect to instance

Connect to your instance [?](#)

You can connect using your browser, or your own compatible SSH client.

Use your browser

Connect using our browser-based SSH client

[Connect using SSH](#)

Use your own SSH client

You can connect to your instance using your own SSH client and the following credentials:

[Connect using an SSH client](#) [↗](#)

CONNECT TO

3.96.81.103

IPv6: 2600:1f11:202:8f00:d2ac:5242:cc66:650e

USER NAME

ubuntu

PASSWORD

This instance uses your wordpress_key (ca-central-1) key pair to sign in.

Go back to the Connect and Click “Connect using SSH” or If you want to connect to your Linux or Unix instance using SSH in terminal, go to the directory where you saved the private key of the key pair that is being used by your instance and command:

```
sudo chmod 400 private-key.pem

ssh -i ./private-key.pem username@public-ip-address
# my command for example
ssh -i ./wordpress_key.pem ubuntu@3.96.81.103
```

3. Install LEMP stack

We created and connected the instance so far. Now I'm going to install Nginx, MySQL, and PHP to Install WordPress.

```
# update Ubuntu
sudo apt update

# install nginx
sudo apt install -y nginx

# check nginx is successfully installed
sudo systemctl status nginx
```

```
# Install MySQL package (select Yes to restart services)
sudo apt install -y mysql-server
```

```
Configuring libssl1.1:amd64

There are services installed on your system which need to be restarted
when certain libraries, such as libpam, libc, and libssl, are upgraded.
Since these restarts may cause interruptions of service for the system,
you will normally be prompted on each upgrade for the list of services
you wish to restart. You can choose this option to avoid being
prompted; instead, all necessary restarts will be done for you
automatically so you can avoid being asked questions on each library
upgrade.

Restart services during package upgrades without asking?

<Yes> <No>
```

```
# Install MySQL secure
sudo mysql_secure_installation
```

```
VALIDATE PASSWORD PLUGIN can be used to test passwords
and improve security. It checks the strength of password
and allows the users to set only those passwords which are
secure enough. Would you like to setup VALIDATE PASSWORD plugin?

Press y|Y for Yes, any other key for No: y
```

Press "y"

```
There are three levels of password validation policy:

LOW      Length >= 8
MEDIUM  Length >= 8, numeric, mixed case, and special characters
STRONG  Length >= 8, numeric, mixed case, special characters and dictionary

Please enter 0 = LOW, 1 = MEDIUM and 2 = STRONG:

asdsasd
```

I suggest "2" for strong password validation policy.

```
New password: ✓
Re-enter new password: ✓

Estimated strength of the password: 100
Do you wish to continue with the password provided?(Press y|Y for Yes, any other key for No) : y
By default, a MySQL installation has an anonymous user,
allowing anyone to log into MySQL without having to have
a user account created for them. This is intended only for
testing, and to make the installation go a bit smoother.
You should remove them before moving into a production
environment.

Remove anonymous users? (Press y|Y for Yes, any other key for No) : y
Success.

Normally, root should only be allowed to connect from
'localhost'. This ensures that someone cannot guess at
the root password from the network.

Disallow root login remotely? (Press y|Y for Yes, any other key for No) : y
```

```
Success.

By default, MySQL comes with a database named 'test' that
anyone can access. This is also intended only for testing,
and should be removed before moving into a production
environment.

Remove test database and access to it? (Press y|Y for Yes, any other key for No) : y
- Dropping test database...
Success.

- Removing privileges on test database...
Success.

Reloading the privilege tables will ensure that all changes
made so far will take effect immediately.

Reload privilege tables now? (Press y|Y for Yes, any other key for No) : y
Success.

All done!
```

Create the password, remove anonymous users, disallow root login remotely, remove test database, and reload.

Install PHP

```
# install PHP
sudo apt install -y php-fpm
# install PHP packages
sudo apt-get install php7.2-mysql php7.2-curl php7.2-json php7.2-cgi php7.2-xsl php7.2-zip
```

Configure

If you successfully installed all packages, I'm going to configure nginx default website. Open the default file by using nano

```
sudo nano /etc/nginx/sites-available/default
```

Scroll down to “pass PHP scripts to FastCGI server” and uncomment(delete #) like this.

```
# pass PHP scripts to FastCGI server
location ~ \.php$ {
    include snippets/fastcgi-php.conf;
    # With php-fpm (or other unix sockets):
    fastcgi_pass unix:/var/run/php/php7.0-fpm.sock;
    #     # With php-cgi (or other tcp sockets):
    #     fastcgi_pass 127.0.0.1:9000;
}
```

Correct version to php7.2. (NOT php7.0-fpm.sock)

```
# pass PHP scripts to FastCGI server
location ~ \.php$ {
    include snippets/fastcgi-php.conf;
    # With php-fpm (or other unix sockets):
    fastcgi_pass unix:/var/run/php/php7.2-fpm.sock;
    #     # With php-cgi (or other tcp sockets):
    #     fastcgi_pass 127.0.0.1:9000;
}
```

```
root /var/www/html;

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

server_name _;

location / {
    # First attempt to serve request as file, then
    # as directory, then fall back to displaying a 404.
    try_files $uri $uri/ =404;
}

# pass PHP scripts to FastCGI server
#
location ~ /\.php$ {
    include snippets/fastcgi-php.conf;

    #
    # With php-fpm (or other unix sockets):
    fastcgi_pass unix:/var/run/php/php7.2-fpm.sock;
    #
    # With php-cgi (or other tcp sockets):
    fastcgi_pass 127.0.0.1:9000;
}

# deny access to .htaccess files, if Apache's document root
# concurs with nginx's one
#
#location ~ /\.ht {
#    deny all;
#}
}
```

Press Ctrl + X, y, and enter to save without changing filename.

restart nginx server

```
sudo service nginx restart
```

Create sample .php file

I'm going to create sample php file to check if php works. I created info.php file and paste one command line:

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

```
<?php phpinfo(); ?>
```

```
GNU nano 2.9.3 /var/www/html/info.php
<?php phpinfo(); ?>
```

^G Get Help	^O Write Out	^W Where Is	^K Cut Text	^J Justify	^C Cur Pos
^X Exit	^R Read File	^_\ Replace	^U Uncut Text	^T To Spell	^_ Go To Line

Ctrl + X, y, and enter to save. Now, you can direct to `youripaddress`/info.php

```
3.96.81.103/info.php
```



System	Linux ip-172-26-12-33 4.15.0-1021-aws #21-Ubuntu SMP Tue Aug 28 10:23:07 UTC 2018 x86_64
Build Date	Oct 25 2021 17:47:59
Server API	FPM/FastCGI
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/7.2/fpm
Loaded Configuration File	/etc/php/7.2/fpm/php.ini
Scan this dir for additional .ini files	/etc/php/7.2/fpm/conf.d
Additional .ini files parsed	/etc/php/7.2/fpm/conf.d/10-mysqlnd.ini, /etc/php/7.2/fpm/conf.d/10-opcache.ini, /etc/php/7.2/fpm/conf.d/10-pdo.ini, /etc/php/7.2/fpm/conf.d/15-xml.ini, /etc/php/7.2/fpm/conf.d/20-calendar.ini, /etc/php/7.2/fpm/conf.d/20-ctype.ini, /etc/php/7.2/fpm/conf.d/20-curl.ini, /etc/php/7.2/fpm/conf.d/20-dom.ini, /etc/php/7.2/fpm/conf.d/20-exif.ini, /etc/php/7.2/fpm/conf.d/20-fileinfo.ini, /etc/php/7.2/fpm/conf.d/20-ftp.ini, /etc/php/7.2/fpm/conf.d/20-gettext.ini, /etc/php/7.2/fpm/conf.d/20-iconv.ini, /etc/php/7.2/fpm/conf.d/20-json.ini, /etc/php/7.2/fpm/conf.d/20-mysqli.ini, /etc/php/7.2/fpm/conf.d/20-pdo_mysql.ini, /etc/php/7.2/fpm/conf.d/20-phar.ini, /etc/php/7.2/fpm/conf.d/20-posix.ini, /etc/php/7.2/fpm/conf.d/20-readline.ini, /etc/php/7.2/fpm/conf.d/20-shmop.ini, /etc/php/7.2/fpm/conf.d/20-simplexml.ini, /etc/php/7.2/fpm/conf.d/20-sockets.ini, /etc/php/7.2/fpm/conf.d/20-sysvmsg.ini, /etc/php/7.2/fpm/conf.d/20-sysvsem.ini, /etc/php/7.2/fpm/conf.d/20-sysvshm.ini, /etc/php/7.2/fpm/conf.d/20-tokenizer.ini, /etc/php/7.2/fpm/conf.d/20-wddx.ini, /etc/php/7.2/fpm/conf.d/20-xmlreader.ini, /etc/php/7.2/fpm/conf.d/20-xmlwriter.ini, /etc/php/7.2/fpm/conf.d/20-xsl.ini, /etc/php/7.2/fpm/conf.d/20-zip.ini
PHP API	20170718
PHP Extension	20170718
Zend Extension	320170718
Zend Extension Build	API320170718,NTS
PHP Extension Build	API20170718,NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	enabled
Zend Memory Manager	enabled
Zend Multibyte Support	disabled
IPv6 Support	enabled
DTrace Support	available, disabled
Registered PHP Streams	https, ftps, compress, zlib, php, file, glob, data, http, ftp, phar, zip
Registered Stream Socket Transports	tcp, udp, unix, udg, ssl, tls, tlsv1.0, tlsv1.1, tlsv1.2
Registered Stream Filters	zlib.*, string.rot13, string.toupper, string.tolower, string.strip_tags, convert.*, consumed, dechunk, convert.iconv.*

This program makes use of the Zend Scripting Language Engine:
 Zend Engine v3.2.0, Copyright (c) 1998-2018 Zend Technologies
 with Zend OPcache v7.2.24-0ubuntu0.18.04.10, Copyright (c) 1999-2018, by Zend Technologies



If you can't see this page, check if you correctly write info.php file or install php packages. If you can see this page, congratulations! you are good to install WordPress!

Install WordPress single website

Create WordPress database

```
# login in mysql as root
sudo mysql -u root -p
```

```
ubuntu@ip-172-26-12-33:~$ sudo mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 6
Server version: 5.7.36-0ubuntu0.18.04.1 (Ubuntu)

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> CREATE DATABASE wordpress DEFAULT CHARACTER SET utf8 COLLATE utf8_unicode_ci;
```

Enter the password and copy and paste this command on MySQL

```
CREATE DATABASE wordpress DEFAULT CHARACTER SET utf8 COLLATE utf8_unicode_ci;
```

Check database is successfully created

```
show databases;
```

```
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
| wordpress |
+-----+
5 rows in set (0.01 sec)
```

Done!

Create User

I'm going to create database user. Replace 'NEWUSER' and 'NEWPASSWORD' to strong password.

```
CREATE USER 'NEWUSER'@'localhost' IDENTIFIED BY 'NEWPASSWORD';
```

Grant all privilege to user and check

```
GRANT ALL ON wordpress.* TO 'NEWUSER'@'localhost' IDENTIFIED BY 'NEWPASSWORD';

#check
SHOW GRANTS FOR 'NEWUSER'@'localhost';
```

```
mysql> SHOW GRANTS FOR 'dlatlrrb'@'localhost';
+-----+
| Grants for dlatlrrb@localhost |
+-----+
| GRANT USAGE ON *.* TO 'dlatlrrb'@'localhost' |
| GRANT ALL PRIVILEGES ON `wordpress`.* TO 'dlatlrrb'@'localhost' |
+-----+
```

Don't forget to refresh privileges before exit.

```
FLUSH PRIVILEGES;

exit;
```

Install WordPress

```
# direct to temporary directory
cd /tmp
# install wordpress file
curl -LO https://wordpress.org/latest.tar.gz
# unzip
tar xzvf latest.tar.gz
# copy sample config php file to wordpress
cp /tmp/wordpress/wp-config-sample.php /tmp/wordpress/wp-config.php
# copy to nginx root directory
sudo cp -a /tmp/wordpress/. /var/www/html/wordpress/
#grant privilege to nginx
sudo chown -R www-data:www-data /var/www/html/wordpress
#
sudo chmod -R 755 /var/www/html/wordpress/
```

Configure wordpress server

```
sudo nano /etc/nginx/sites-available/wordpress
```



```
server {
    listen 80;
    listen [::]:80;
    root /var/www/html/wordpress;
    index index.php index.html index.htm;
    server_name example.net *.example.net example.com example2.org;

    client_max_body_size 100M;

    location / {
        try_files $uri $uri/ /index.php?$args;
    }

    location ~ \.php$ {
        include snippets/fastcgi-php.conf;
        fastcgi_pass unix:/var/run/php/php7.2-fpm.sock;
        fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
        include fastcgi_params;
    }

}
```

Replace server name 'example' to your domain name.

Unlink default symbolic link

```
sudo unlink /etc/nginx/sites-enabled/default

sudo systemctl reload nginx
```

Error establishing a database connection

You should see this message.

Wordpress configuration

```
sudo nano /var/www/html/wordpress/wp-config.php
```

```
// ** MySQL settings - You can get this info from your web host ** //  
/** The name of the database for WordPress */  
define( 'DB_NAME', 'database_name_here' );  
  
/** MySQL database username */  
define( 'DB_USER', 'username_here' );  
  
/** MySQL database password */  
define( 'DB_PASSWORD', 'password_here' );  
  
/** MySQL hostname */  
define( 'DB_HOST', 'localhost' );  
  
/** Database charset to use in creating database tables. */  
define( 'DB_CHARSET', 'utf8' );  
  
/** The database collate type. Don't change this if in doubt. */  
define( 'DB_COLLATE', '' );
```

```
// ** MySQL settings - You can get this info from your web host ** //  
/** The name of the database for WordPress */  
define( 'DB_NAME', 'wordpress' );  
  
/** MySQL database username */  
define( 'DB_USER', 'dlatlrrb' );  
  
/** MySQL database password */  
define( 'DB_PASSWORD', 'XXXXXXXXXX' );  
  
/** MySQL hostname */  
define( 'DB_HOST', 'localhost' );  
  
/** Database charset to use in creating database tables. */  
define( 'DB_CHARSET', 'utf8' );  
  
/** The database collate type. Don't change this if in doubt. */  
define( 'DB_COLLATE', '' );
```

Replace like this

Secret key

Scroll down to Authentication unique keys and salts and replace to new key.

```
*
* @since 2.6.0
*/
define( 'AUTH_KEY',          'put your unique phrase here' );
define( 'SECURE_AUTH_KEY',  'put your unique phrase here' );
define( 'LOGGED_IN_KEY',    'put your unique phrase here' );
define( 'NONCE_KEY',        'put your unique phrase here' );
define( 'AUTH_SALT',        'put your unique phrase here' );
define( 'SECURE_AUTH_SALT', 'put your unique phrase here' );
define( 'LOGGED_IN_SALT',   'put your unique phrase here' );
define( 'NONCE_SALT',       'put your unique phrase here' );
/**#@-*/
```

Delete all that I highlighted and paste the key on it

```
* 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
* You can change these at any point in time to invalidate all existing cookies. This will force
*
* @since 2.6.0
*/
define( 'AUTH_KEY',          'i-?w-/E0 155 80' i7u 4u76110 1011?ic 5T 0/W 01' -AvZ;t4_ ');
define( 'SECURE_AUTH_KEY',  'eHJ4z>]U [redacted] eg?@KT{0' );
define( 'LOGGED_IN_KEY',    '+ D2AvUE [redacted] cTG&#%we' );
define( 'NONCE_KEY',        'P%]3Y%TT [redacted] eieEttX' );
define( 'AUTH_SALT',        'u<@^V2&: [redacted] i`ga( pG' );
define( 'SECURE_AUTH_SALT', ':-+tGg}g [redacted] snq{0qz|' );
define( 'LOGGED_IN_SALT',   '{%?r?7=X [redacted] -u^VR&m0' );
define( 'NONCE_SALT',       'FMI7TW9l, [redacted] *ee&,pB|' );
/**#@-*/
```

```
sudo systemctl reload nginx
```



English (United States)

Afrikaans

العربية

العربية المغربية

অসমীয়া

Azərbaycan dili

گۆنئی آذربایجان

Беларуская мова

Български

বাংলা

ବିହାରୀ

Bosanski

Català

Cebuano

Čeština

Cymraeg

Dansk

Deutsch (Schweiz, Du)

Deutsch (Österreich)

Deutsch

Deutsch (Sie)

Deutsch (Südafrika)

Continue

Done!