



## 1. Get Started with EC2

Go to <https://aws.amazon.com/ec2/> and click the orange **Get started with Amazon EC2** button. Sign up for an AWS account if you don't already have one. Create an EC2 Instance.

- Select t2.micro.
- Select Ubuntu Server 20.04
- Add Storage: 30 GB
- Configure Security Group: HTTP from Anywhere
- Configure Security Group: HTTPS from Anywhere
- Create keypair

Copy out the public IP Address.

## 2. Register Domain name

Use namecheap and register domain name at a cheap rate. Mostly \$11 a domain.

## 3. Link Domain Name

Goto **route53** >> **Create Hosted Zone** >> **Type in:**

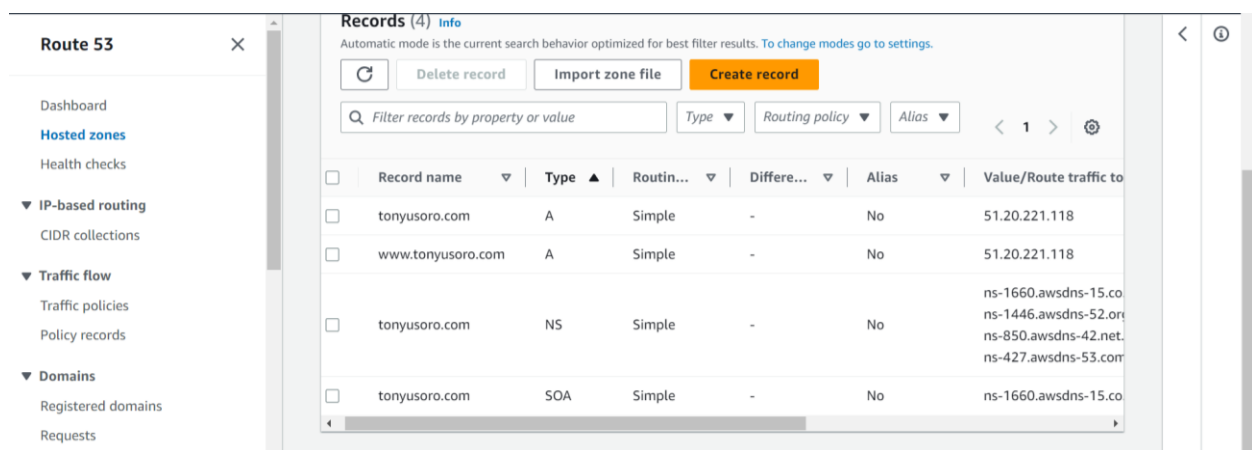
**Domain name:** e.g. **tonyusoro.com**

**Select:** **Public hosted zone**

**Click:** **Create Hosted Zone**

Once successful: Click **Create Record**:

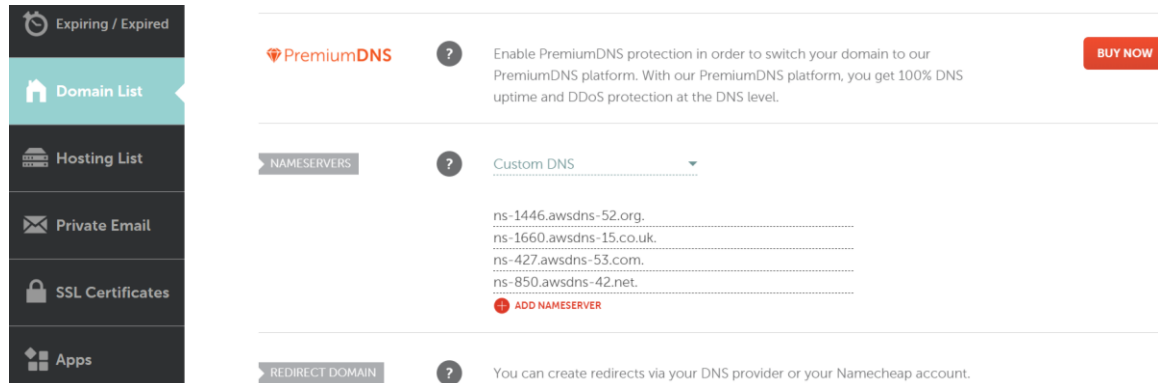
Create A record for both [tonyusoro.com](http://tonyusoro.com) and [www.tonyusoro.com](http://www.tonyusoro.com) to the public IP address copied from the instance.



Now copy NS Records from (**Value/Route traffic to**) to the domain name on [namecheap.com](http://namecheap.com)

Get to the domain then click: **Manage**

**Select: Custom Domain**  
**Paste One after another all 4 NS Records.**

The screenshot shows the Namecheap DNS management interface. On the left is a sidebar with navigation links: 'Expiring / Expired', 'Domain List' (highlighted), 'Hosting List', 'Private Email', 'SSL Certificates', and 'Apps'. The main content area has a 'PremiumDNS' section with a 'BUY NOW' button and a description. Below that is the 'NAME SERVERS' section, which shows a dropdown menu set to 'Custom DNS'. It lists four name servers: ns-1446.awsdns-52.org, ns-1660.awsdns-15.co.uk, ns-427.awsdns-53.com, and ns-850.awsdns-42.net. There is an 'ADD NAMESERVER' button. At the bottom is the 'REDIRECT DOMAIN' section with a description about creating redirects.

**NOTE: Allow it 48 Hours to populate on DNS Servers Worldwide.**

## 4. Now Login to EC2 via SSH

```
chmod 400 awsec2.pem
```

```
ssh -i awsec2.pem ubuntu@IP
```

## 5. Update System and Install LEMP Packages

Execute the following to upgrade Ubuntu server packages.

```
sudo apt update
```

```
sudo apt upgrade
```

Use the apt package manager to install PHP, MariaDB, and the Nginx web server.

```
sudo apt install nginx mariadb-server php-fpm php-mysql
```

## 6. Install WordPress

After logging in to your server as described above, execute the following commands to install WordPress on Ubuntu.

```
cd /var/www
sudo wget https://wordpress.org/latest.tar.gz
sudo tar -xzf latest.tar.gz
sudo rm latest.tar.gz
sudo chown -R www-data:www-data wordpress
sudo find wordpress/ -type d -exec chmod 755 {} \;
sudo find wordpress/ -type f -exec chmod 644 {} \;
```

## 7. Setup the Database

Secure your MariaDB installation by adding a password and disabling other features. When prompted, answer Y.

```
sudo mysql_secure_installation
```

Access the MariaDB console with the password that you just created.

```
sudo mysql -u root -p
```

Within the MariaDB console, create a database for WordPress. Please choose your own database name, user name, and a password.

```
create database example_db default character set utf8 collate utf8_unicode_ci;
create user 'example_user'@'localhost' identified by 'example_pw';
grant all privileges on example_db.* TO 'example_user'@'localhost';
flush privileges;
exit
```

## 8. Configure Nginx Web Server

Navigate to the directory which contains configuration files for the Nginx web server, and create a new configuration file with the text editor of your choice. In this example, the text editor is [vim](#).

```
cd /etc/nginx/sites-available/  
sudo vim wordpress.conf
```

Use this configuration as a template for your website. Please change the *server\_name* and make sure that the *php-handler* socket exists (you may have a different version of PHP installed).

```
upstream php-handler {  
    server unix:/var/run/php/php8.1-fpm.sock;  
}  
server {  
    listen 80;  
    server_name tonyusoro.com www.tonyusoro.com;  
    root /var/www/wordpress;  
    index index.php;  
    location / {  
        try_files $uri $uri/ /index.php?$args;  
    }  
    location ~ \.php$ {  
        include snippets/fastcgi-php.conf;  
        fastcgi_pass php-handler;  
    }  
}
```


Make a symbolic link to tell Nginx about your website, and apply the changes by restarting the web server.

```
sudo ln -s /etc/nginx/sites-available/wordpress.conf /etc/nginx/sites-enabled/  
sudo nginx -t    >> Check to see it is successful.  
sudo systemctl restart nginx
```

## 9. Finish WordPress Install

Assuming that DNS propagation has finished, you can now access your website via your domain name in a web browser. You will be prompted to finish the WordPress installation, part of which is entering the database name, user, and password that you created earlier.

NOTE: USE ANOTHER BROWSER AS BROWSER CACHEING MAY MAKE YOU THINK YOUR SITE ISNT READY YET....USE ANOTHER BROWSER EG FIREFOX.



Below you should enter your database connection details. If you're not sure about these, contact your host.

Database Name	<input type="text" value="example_db"/>	The name of the database you want to use with WordPress.
Username	<input type="text" value="example_user"/>	Your database username.
Password	<input type="text" value="example_pw"/>	Your database password.
Database Host	<input type="text" value="localhost"/>	You should be able to get this info from your web host, if localhost doesn't work.
Table Prefix	<input type="text" value="wp_"/>	If you want to run multiple WordPress installations in a single database, change this.

Upon completion of the installation, you can access your WordPress administrator dashboard at <http://example.com/wp-admin/> where example.com is your domain name.

Welcome to the famous five-minute WordPress installation process! Just fill in the information below and you'll be on your way to using the most extendable and powerful personal publishing platform in the world.

### Information needed

Please provide the following information. Do not worry, you can always change these settings later.

**Site Title**

**Username**

Username can have only alphanumeric characters, spaces, underscores, hyphens, periods, and the @ symbol.

**Password**  [Hide](#)

Strong

**Important:** You will need this password to log in. Please store it in a secure location.

**Your Email**

Double-check your email address before continuing.

**Search engine visibility** ☐ Discourage search engines from indexing this site

It is up to search engines to honor this request.

## 10. Install PHP Packages Required by WordPress

From your WordPress administrator dashboard, navigate to Tools > Site Health > Status and you may see a critical issue that says "One or more required modules are missing".

To fix this, go back to your EC2 instance's console window and install these packages.

```
sudo apt install php-curl php-dom php-mbstring php-imagick php-zip php-gd php-intl
```

## 11. Install an SSL Certificate for HTTPS

Secure your website with an SSL certificate from Let's Encrypt. To do this, execute the following commands.

```
sudo apt install snapd
```

```
sudo snap install core; snap refresh core
```

```
sudo snap install --classic certbot
```

```
sudo ln -s /snap/bin/certbot /usr/bin/certbot
```

```
sudo certbot --nginx >> if that doesn't work try this >> sudo certbot --nginx certonly
```

Finally back in your WordPress administrator dashboard, go to Settings > General and change the *WordPress Address* and *Site Address* to start with **https**.

## Next Steps

Now that you're all set up a fresh install of WordPress on your EC2 instance, check out my list of [15 important things to do after installing WordPress](#) next.

Also for your convenience, the following is a detailed video walkthrough of the steps in this tutorial.