Guided Lab: Installing WordPress on an Ubuntu EC2 Instance with LEMP Stack

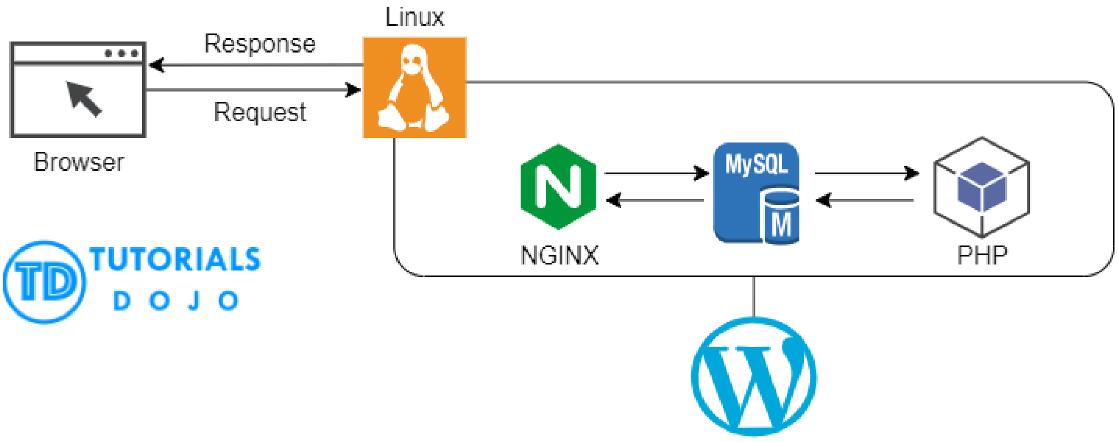
Description

In this lab, you'll learn how to set up a WordPress website on an Amazon EC2 instance running Ubuntu. We'll use the LEMP (Linux, Nginx, MySQL, PHP) stack to achieve this.

LEMP is an open-source web application stack that we may use to create web apps. LEMP's abbreviation stands for Linux Operating System, Nginx (pronounced engine-x), web server, MySQL database, and PHP programming language.

- **Linux** is a supported and maintained Linux image provided by Amazon Web Services (AWS) for use on Amazon Elastic Compute Cloud (Amazon EC2). It is designed to provide a stable, secure, and high-performance execution environment for applications running on Amazon EC2.
- Nginx, which is pronounced "engine-x," is a high-performance web server and reverse proxy. It excels at handling concurrent connections, serving static files, and load balancing. Many websites use Nginx due to its efficiency and scalability.
- **MySQL** is a widely used open-source relational database management system (RDBMS) that organizes data in tables with rows and columns, making it ideal for applications requiring structured data storage. WordPress, among others, relies on MySQL for data persistence.
- **PHP** is a server-side scripting language used for web development. It enables dynamic content generation, database connectivity, and interaction with web servers. WordPress plugins and themes often leverage PHP for customization.

WordPress is a widely used open-source content management system (CMS) that allows users to easily create and manage websites, blogs, and online stores. It offers a user-friendly interface, customizable themes, and a wide array of plugins for additional functionality.



Objectives

In this lab, you will learn how to:

1. Launch an Amazon EC2 instance.
2. Install and configure Nginx, MySQL, and PHP.
3. Download and set up WordPress.
4. Configure Nginx to serve WordPress.

Prerequisites

Before starting this lab, ensure you have taken the following labs:

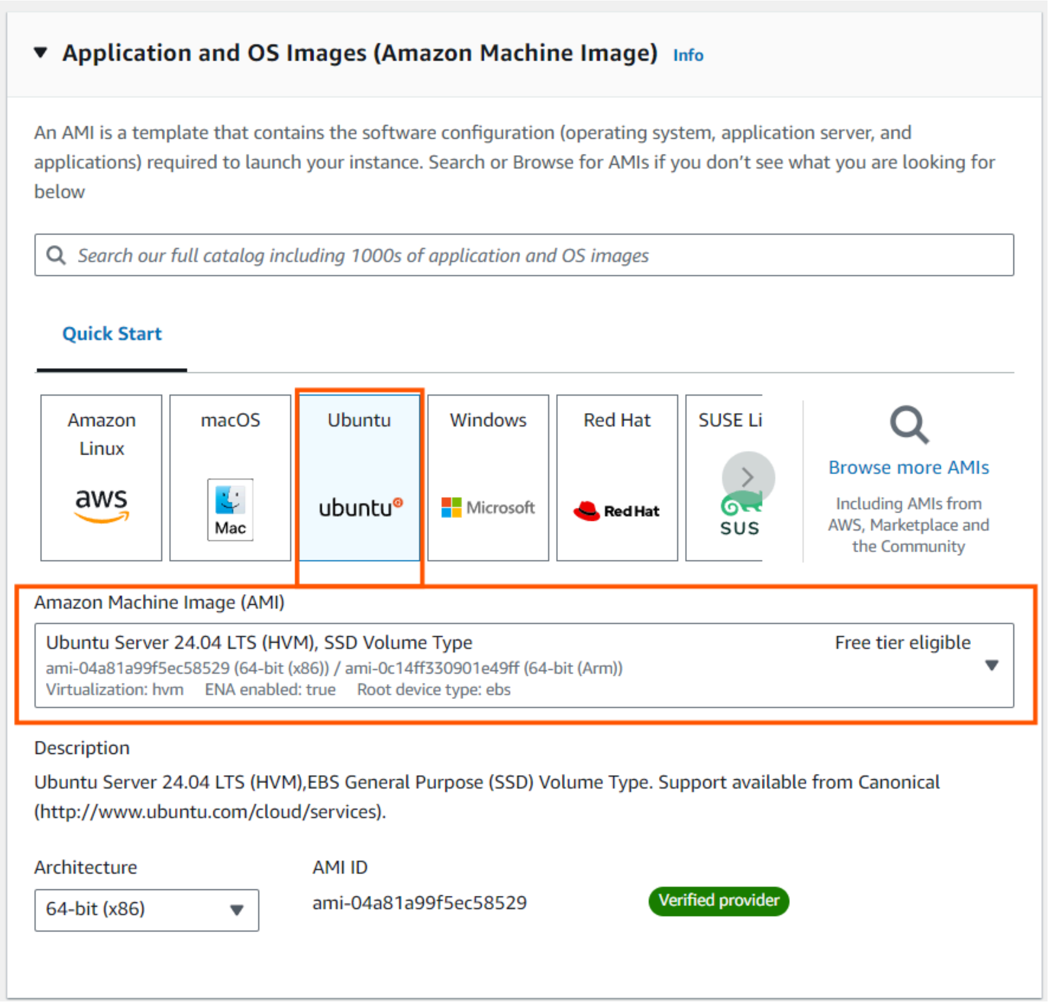
- Launching an EC2 Instance with User Data
- Creating an Amazon EC2 instance (Linux)
- Setting up a Web server on an EC2 instance

Subscribe to access AWS PlayCloud Labs

Lab Steps

Launching an EC2 instance (Ubuntu)

- Navigate to EC2.
- Launch a new instance with Ubuntu Server.



Application and OS Images (Amazon Machine Image) Info

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Search our full catalog including 1000s of application and OS images

Quick Start

Amazon Machine Image (AMI)

Ubuntu Server 24.04 LTS (HVM), SSD Volume Type — Free tier eligible
ami-04a81a99f5ec58529 (64-bit (x86)) / ami-0c14ff330901e49ff (64-bit (Arm))
Virtualization: hvm ENA enabled: true Root device type: ebs

Description

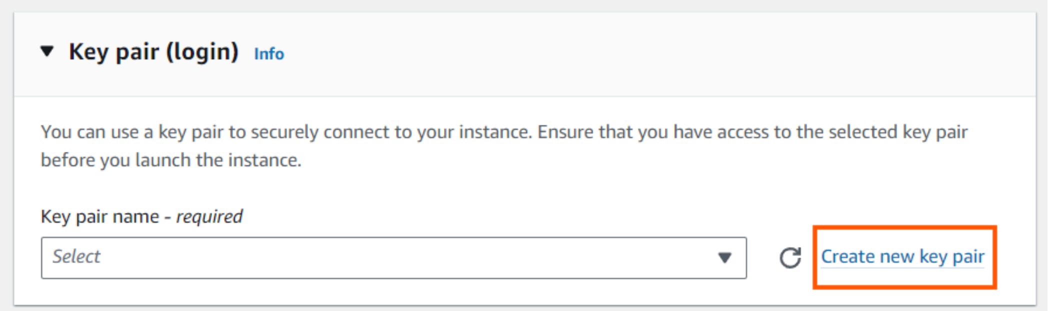
Ubuntu Server 24.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (http://www.ubuntu.com/cloud/services).

Architecture: 64-bit (x86)

AMI ID: ami-04a81a99f5ec58529

Verified provider

- Create a new key pair.



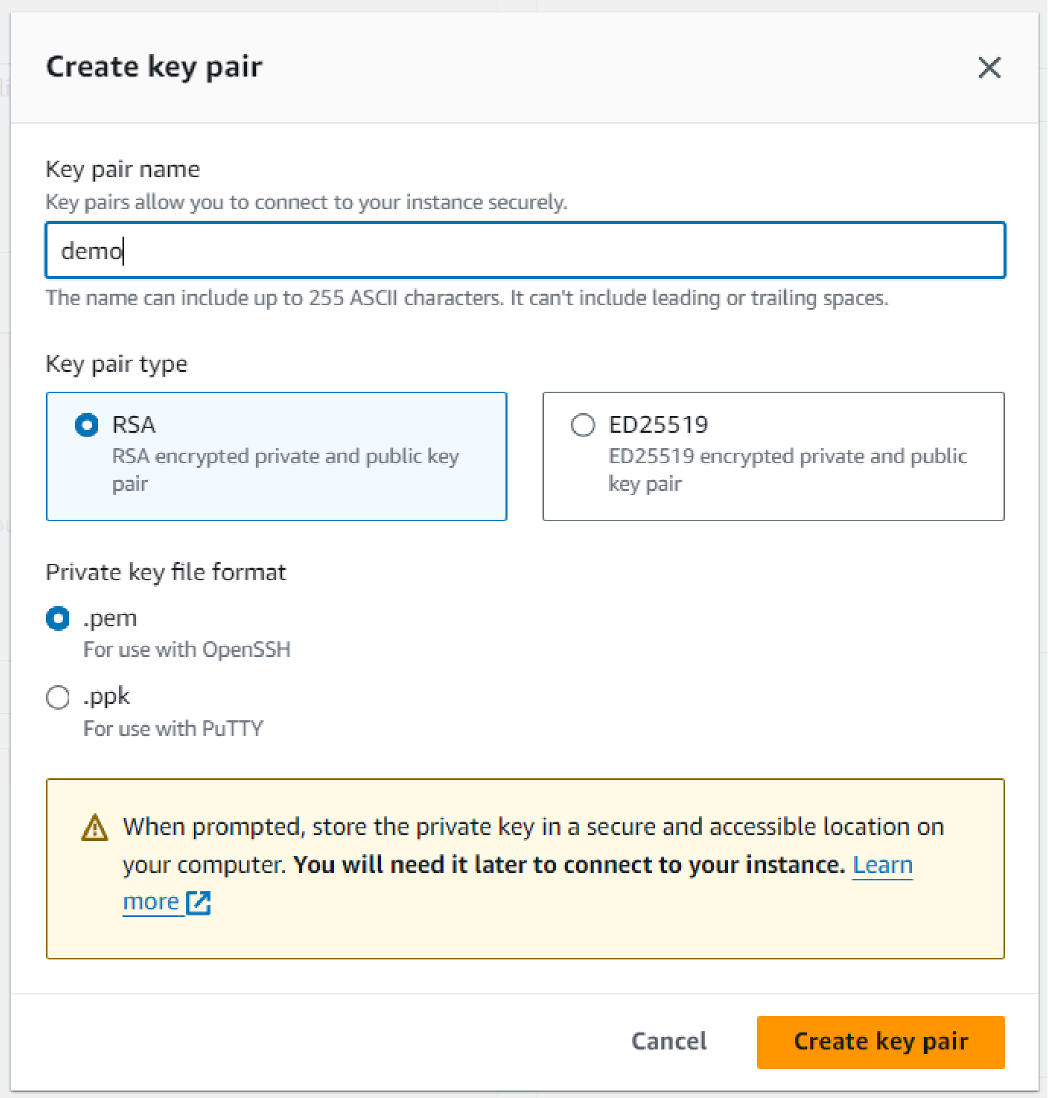
Key pair (login) Info

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - *required*

Select

- Key pair type: RSA
- Private key file format: pem



Create key pair

Key pair name

Key pairs allow you to connect to your instance securely.

demo

The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.

Key pair type

● RSA
RSA encrypted private and public key pair

○ ED25519
ED25519 encrypted private and public key pair

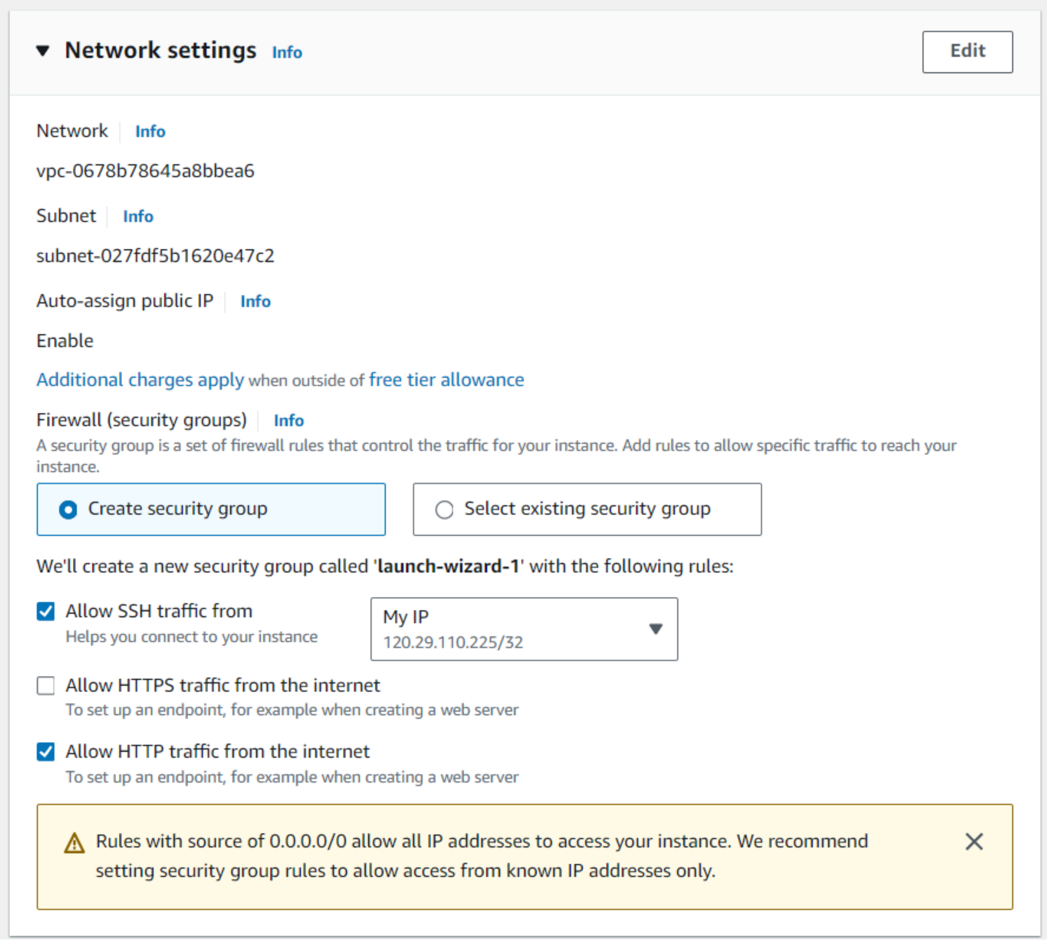
Private key file format

● .pem
For use with OpenSSH

○ .ppk
For use with PuTTY

⚠ When prompted, store the private key in a secure and accessible location on your computer. **You will need it later to connect to your instance.** Learn more

- Configure security groups to allow SSH connection to **My IP and HTTP** traffic from the internet.



▼ Network settings Info

Network Info

vpc-0678b78645a8bbea6

Subnet Info

subnet-027fdf5b1620e47c2

Auto-assign public IP Info

Enable

Additional charges apply when outside of free tier allowance

Firewall (security groups) Info

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

● Create security group ○ Select existing security group

We'll create a new security group called '**launch-wizard-1**' with the following rules:

☑ Allow SSH traffic from
Helps you connect to your instance

My IP
120.29.110.225/32

□ Allow HTTPS traffic from the internet
To set up an endpoint, for example when creating a web server

☑ Allow HTTP traffic from the internet
To set up an endpoint, for example when creating a web server

⚠ Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

- Click the **Launch Instance** button.
- Once created, Connect to Your EC2 Instance:
  - Use SSH to connect to your EC2 instance using the key pair you specified during the launch.

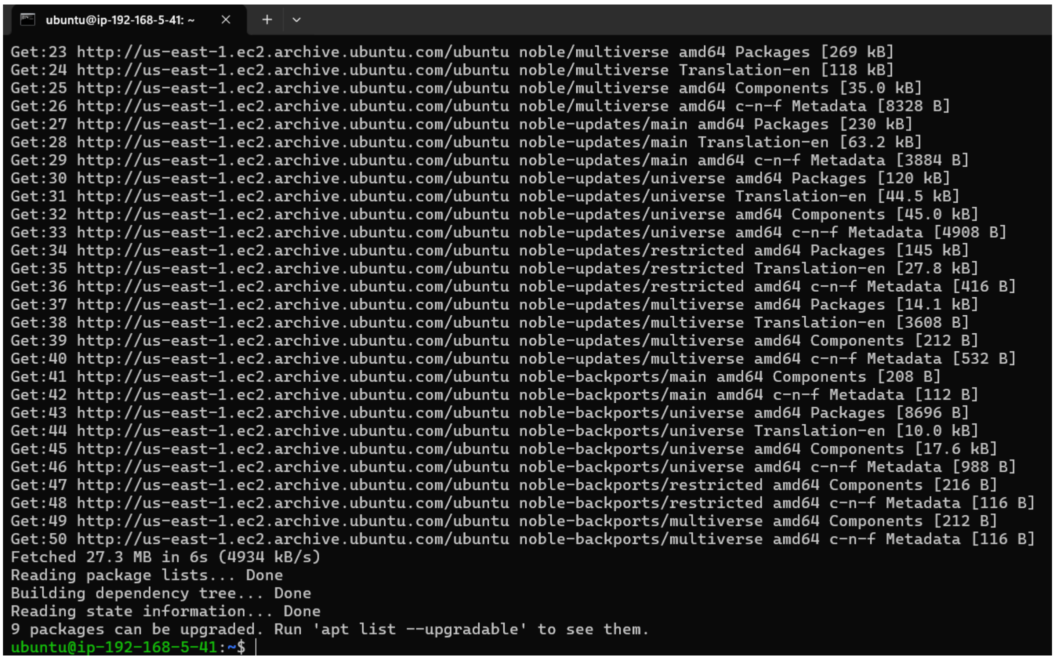
    ```
    ssh -i .pem_file ubuntu@ec2_instance_IPv4
    ```

Installing the NGINX web server

- Connect to the server via ssh.

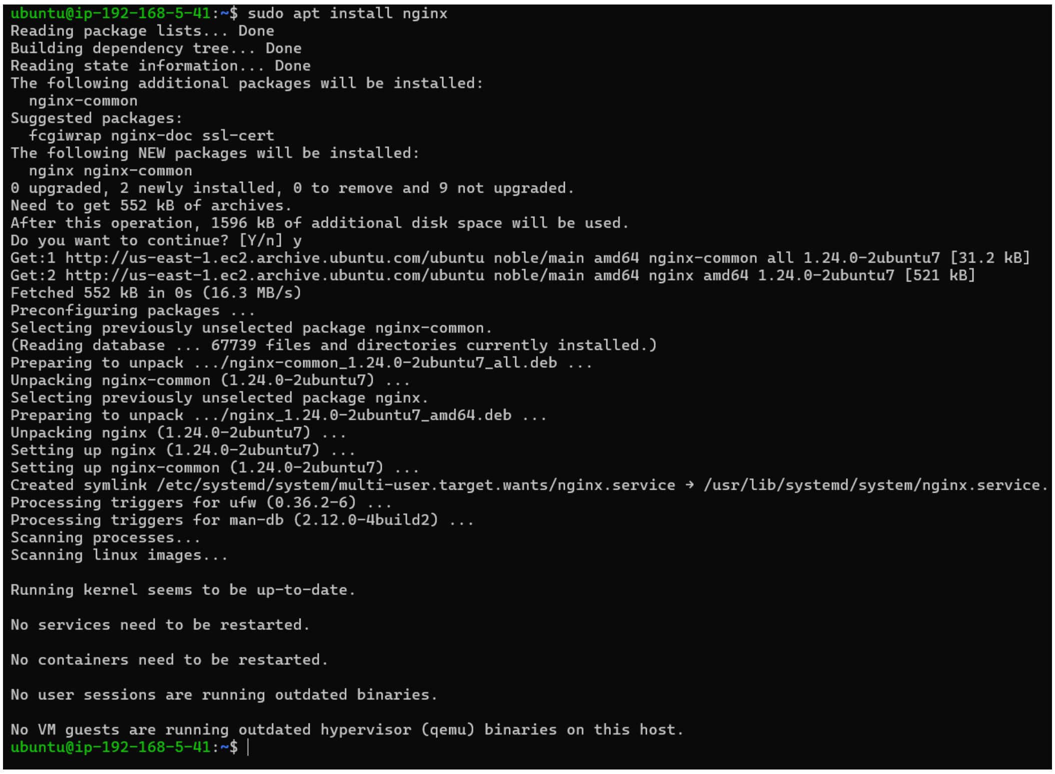
- Install the NGINX web server.

  ```
  sudo apt update
  ```



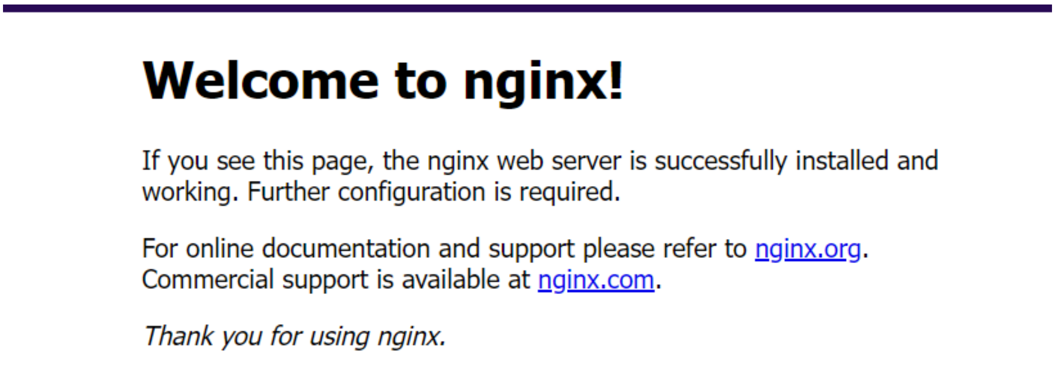
```
Get:23 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [269 kB]
Get:24 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse Translation-en [118 kB]
Get:25 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Components [35.0 kB]
Get:26 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 c-n-f Metadata [8328 B]
Get:27 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [230 kB]
Get:28 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main Translation-en [63.2 kB]
Get:29 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 c-n-f Metadata [3884 B]
Get:30 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [120 kB]
Get:31 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe Translation-en [44.5 kB]
Get:32 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Components [45.0 kB]
Get:33 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 c-n-f Metadata [4908 B]
Get:34 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Packages [145 kB]
Get:35 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted Translation-en [27.8 kB]
Get:36 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 c-n-f Metadata [416 B]
Get:37 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Packages [14.1 kB]
Get:38 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse Translation-en [3608 B]
Get:39 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Components [212 B]
Get:40 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 c-n-f Metadata [532 B]
Get:41 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/main amd64 Components [208 B]
Get:42 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/main amd64 c-n-f Metadata [112 B]
Get:43 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 Packages [8696 B]
Get:44 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe Translation-en [10.0 kB]
Get:45 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 Components [17.6 kB]
Get:46 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 c-n-f Metadata [988 B]
Get:47 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/restricted amd64 Components [216 B]
Get:48 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/restricted amd64 c-n-f Metadata [116 B]
Get:49 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 Components [212 B]
Get:50 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 c-n-f Metadata [116 B]
Fetched 27.3 MB in 6s (4934 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
9 packages can be upgraded. Run 'apt list --upgradable' to see them.
ubuntu@ip-192-168-5-41:~$ |
```

```
sudo apt install nginx
```



```
ubuntu@ip-192-168-5-41:~$ sudo apt install nginx
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  nginx-common
Suggested packages:
  fcgiwrap nginx-doc ssl-cert
The following NEW packages will be installed:
  nginx nginx-common
0 upgraded, 2 newly installed, 0 to remove and 9 not upgraded.
Need to get 552 kB of archives.
After this operation, 1596 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 nginx-common all 1.24.0-2ubuntu7 [31.2 kB]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 nginx amd64 1.24.0-2ubuntu7 [521 kB]
Fetched 552 kB in 0s (16.3 MB/s)
Preconfiguring packages ...
Selecting previously unselected package nginx-common.
(Reading database ... 67739 files and directories currently installed.)
Preparing to unpack .../nginx-common_1.24.0-2ubuntu7_all.deb ...
Unpacking nginx-common (1.24.0-2ubuntu7) ...
Selecting previously unselected package nginx.
Preparing to unpack .../nginx_1.24.0-2ubuntu7_amd64.deb ...
Unpacking nginx (1.24.0-2ubuntu7) ...
Setting up nginx (1.24.0-2ubuntu7) ...
Setting up nginx-common (1.24.0-2ubuntu7) ...
Created symlink /etc/systemd/system/multi-user.target.wants/nginx.service → /usr/lib/systemd/system/nginx.service.
Processing triggers for ufw (0.36.2-6) ...
Processing triggers for man-db (2.12.0-4build2) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-192-168-5-41:~$ |
```

- Once installed, open the Public IPv4 address of each instance in the browser using HTTP to verify if the NGINX has been successfully installed.



Welcome to nginx!

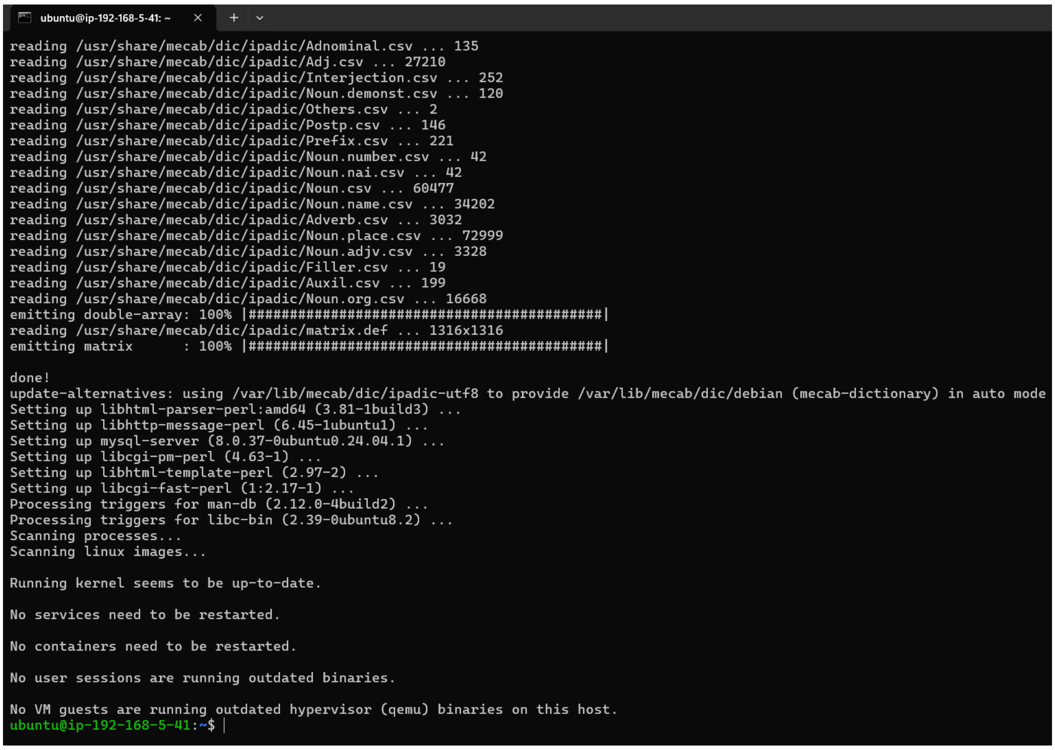
If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

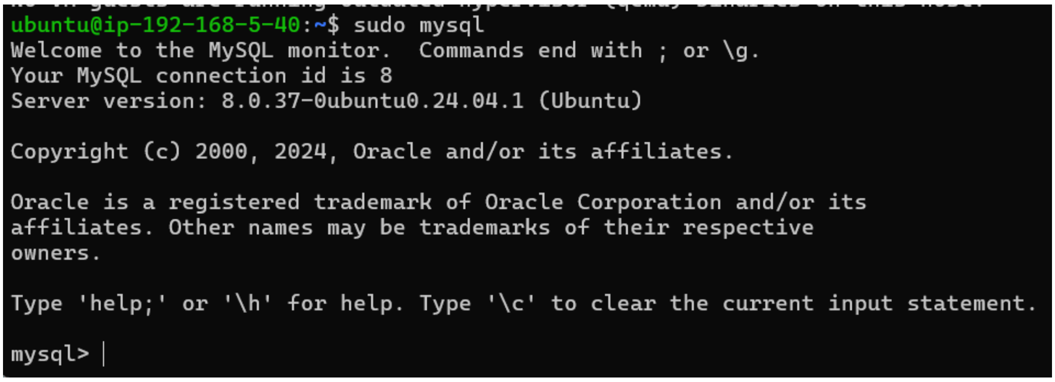
Installing MySQL

- Go back to the SHH and install the MySQL. `sudo apt install mysql-server`
    - Enter Y when prompted to confirm that you wish to install the MySQL server.



```
reading /usr/share/mecab/dic/ipadic/Adnominal.csv ... 135
reading /usr/share/mecab/dic/ipadic/Adj.csv ... 27210
reading /usr/share/mecab/dic/ipadic/Interjection.csv ... 252
reading /usr/share/mecab/dic/ipadic/Noun.demonst.csv ... 120
reading /usr/share/mecab/dic/ipadic/Others.csv ... 2
reading /usr/share/mecab/dic/ipadic/Postp.csv ... 146
reading /usr/share/mecab/dic/ipadic/Prefix.csv ... 221
reading /usr/share/mecab/dic/ipadic/Noun.number.csv ... 42
reading /usr/share/mecab/dic/ipadic/Noun.nai.csv ... 42
reading /usr/share/mecab/dic/ipadic/Noun.csv ... 60477
reading /usr/share/mecab/dic/ipadic/Noun.name.csv ... 34202
reading /usr/share/mecab/dic/ipadic/Adverb.csv ... 3032
reading /usr/share/mecab/dic/ipadic/Noun.place.csv ... 72999
reading /usr/share/mecab/dic/ipadic/Noun.adjv.csv ... 3328
reading /usr/share/mecab/dic/ipadic/Filler.csv ... 19
reading /usr/share/mecab/dic/ipadic/Auxil.csv ... 199
reading /usr/share/mecab/dic/ipadic/Noun.org.csv ... 16668
emitting double-array: 100% |###########################################|
reading /usr/share/mecab/dic/ipadic/matrix.def ... 1316x1316
emitting matrix      : 100% |###########################################|

done!
update-alternatives: using /var/lib/mecab/dic/ipadic-utf8 to provide /var/lib/mecab/dic/debian (mecab-dictionary) in auto mode
Setting up libhtml-parser-perl:amd64 (3.81-1build3) ...
Setting up libhttp-message-perl (6.45-1ubuntu1) ...
Setting up mysql-server (8.0.37-0ubuntu0.24.04.1) ...
Setting up libcgi-pm-perl (4.63-1) ...
Setting up libhtml-template-perl (2.97-2) ...
Setting up libcgi-fast-perl (1:2.17-1) ...
Processing triggers for man-db (2.12.0-4build2) ...
Processing triggers for libc-bin (2.39-0ubuntu8.2) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-192-168-5-41:~$ |
```

- To verify successful installation, type `sudo mysql`

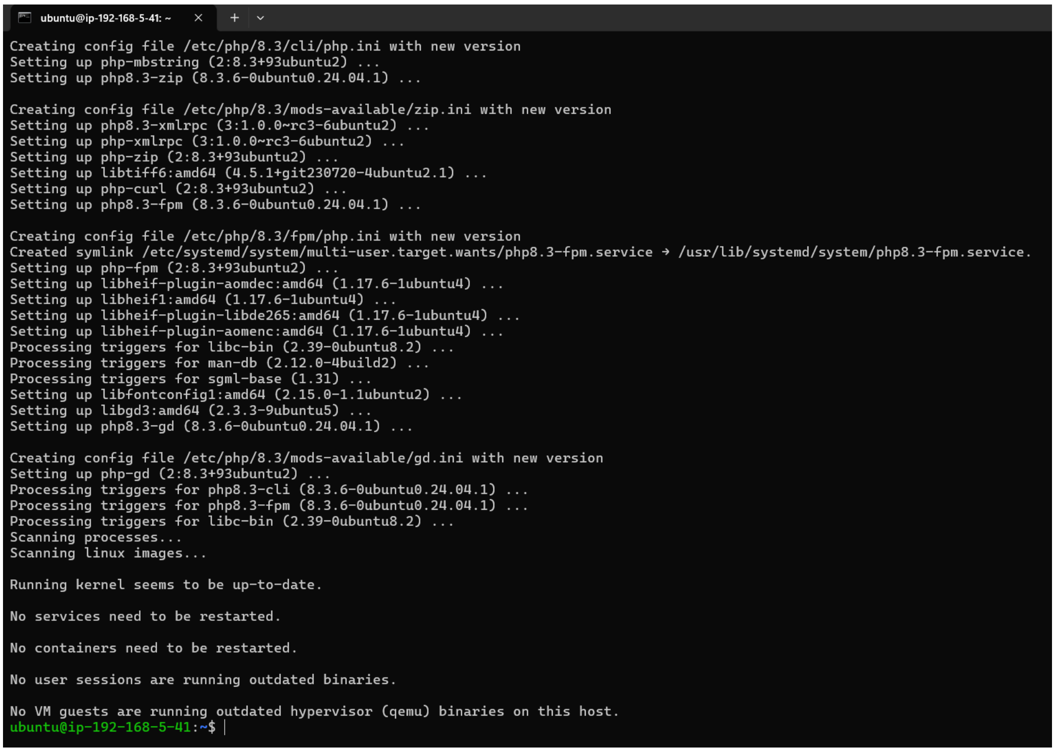


```
ubuntu@ip-192-168-5-40:~$ sudo mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.37-0ubuntu0.24.04.1 (Ubuntu)

Copyright (c) 2000, 2024, 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> |
```

Installing PHP and Required extensions

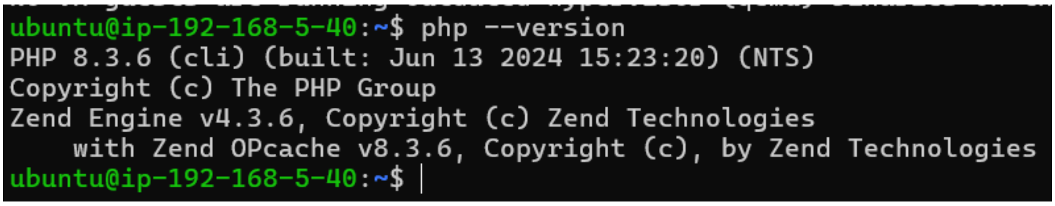
- Install PHP and extensions: `sudo apt install php-fpm php-mysql php-curl php-gd php-mbstring php-xml php-xmlrpc php-zip`

- Enter Y when prompted to confirm that you wish to install PHP.



```
Creating config file /etc/php/8.3/cli/php.ini with new version
Setting up php-mbstring (2:8.3+93ubuntu2) ...
Setting up php8.3-zip (8.3.6-0ubuntu0.24.04.1) ...

Creating config file /etc/php/8.3/mods-available/zip.ini with new version
Setting up php8.3-xmlrpc (3:1.0.0~rc3-6ubuntu2) ...
Setting up php-xmlrpc (3:1.0.0~rc3-6ubuntu2) ...
Setting up php-zip (2:8.3+93ubuntu2) ...
Setting up libtiff6:amd64 (4.5.1+git230720-4ubuntu2.1) ...
Setting up php-curl (2:8.3+93ubuntu2) ...
Setting up php8.3-fpm (8.3.6-0ubuntu0.24.04.1) ...

Creating config file /etc/php/8.3/fpm/php.ini with new version
Created symlink /etc/systemd/system/multi-user.target.wants/php8.3-fpm.service → /usr/lib/systemd/system/php8.3-fpm.service.
Setting up php-fpm (2:8.3+93ubuntu2) ...
Setting up libheif-plugin-aomdec:amd64 (1.17.6-1ubuntu4) ...
Setting up libheif1:amd64 (1.17.6-1ubuntu4) ...
Setting up libheif-plugin-libde265:amd64 (1.17.6-1ubuntu4) ...
Setting up libheif-plugin-aomenc:amd64 (1.17.6-1ubuntu4) ...
Processing triggers for libc-bin (2.39-0ubuntu8.2) ...
Processing triggers for man-db (2.12.0-4build2) ...
Processing triggers for sgml-base (1.31) ...
Setting up libfontconfig1:amd64 (2.15.0-1.1ubuntu2) ...
Setting up libgd3:amd64 (2.3.3-9ubuntu5) ...
Setting up php8.3-gd (8.3.6-0ubuntu0.24.04.1) ...

Creating config file /etc/php/8.3/mods-available/gd.ini with new version
Setting up php-gd (2:8.3+93ubuntu2) ...
Processing triggers for php8.3-cli (8.3.6-0ubuntu0.24.04.1) ...
Processing triggers for php8.3-fpm (8.3.6-0ubuntu0.24.04.1) ...
Processing triggers for libc-bin (2.39-0ubuntu8.2) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-192-168-5-41:~$
```

- To verify successful installation, type `php --version`



```
ubuntu@ip-192-168-5-40:~$ php --version
PHP 8.3.6 (cli) (built: Jun 13 2024 15:23:20) (NTS)
Copyright (c) The PHP Group
Zend Engine v4.3.6, Copyright (c) Zend Technologies
    with Zend OPcache v8.3.6, Copyright (c), by Zend Technologies
ubuntu@ip-192-168-5-40:~$
```

Configuring Nginx for WordPress

Note: For the purposes of this lab, we'll use the domain name "demo" in all our configurations. Feel free to replace "demo" with your desired name.

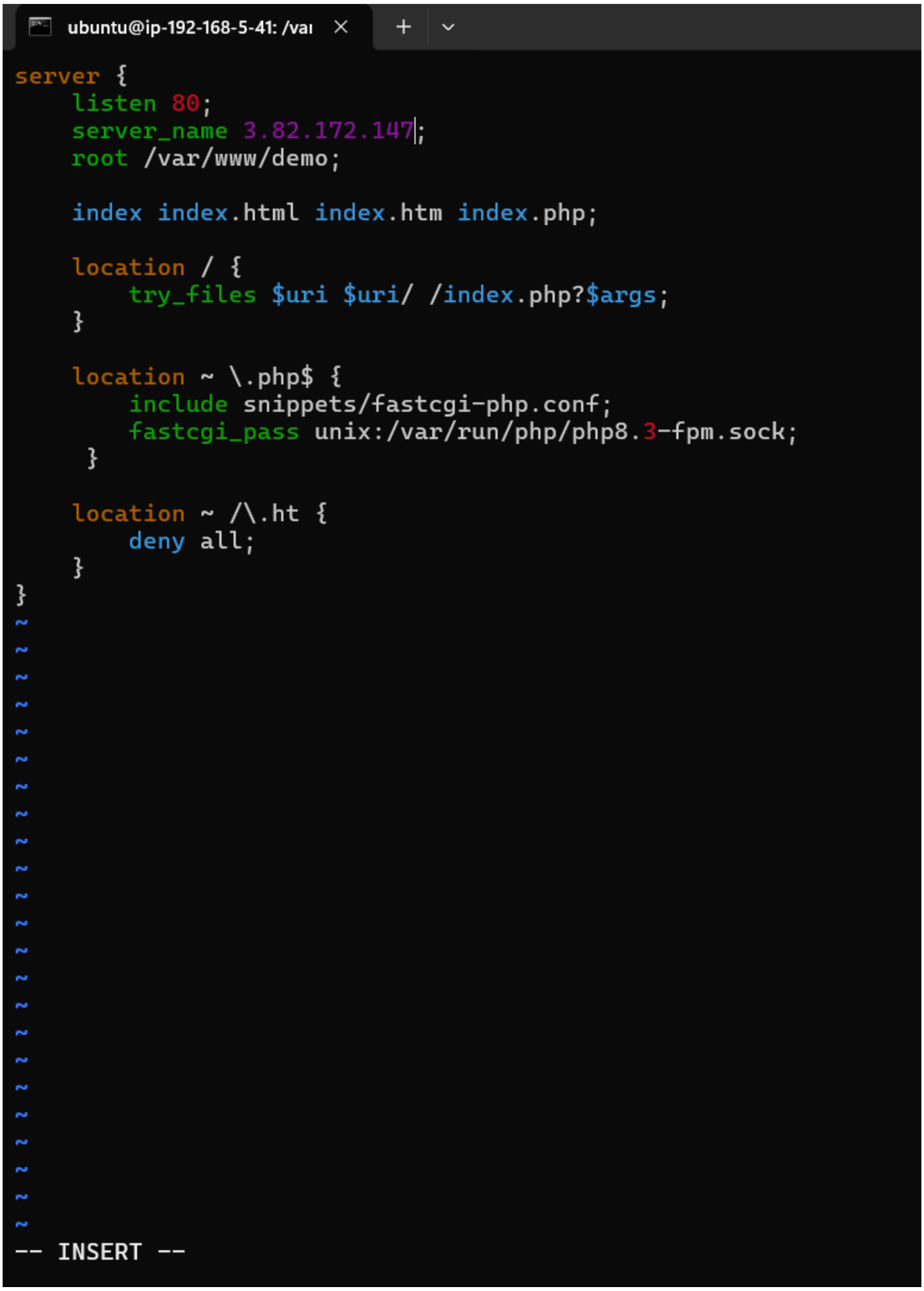
Configure Nginx to use PHP-FPM.

- Create a Directory for Your WordPress Site: `sudo mkdir /var/www/demo`
- Set Permissions for the Directory: `sudo chown -R $USER:$USER /var/www/demo` – It sets the owner and group to the current user (ubuntu).
- Create an Nginx Configuration File for Your Site: `sudo vi /etc/nginx/sites-available/demo` – This will

generate a new empty file. Copy and paste the following barebones configuration:

```
server {
listen 80;
server_name your_IPv4;
root /var/www/demo;

index index.html index.htm index.php;

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

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

location ~ /\.ht {
deny all;
}
}
```

- **Replace your_IPv4 with the Public IPv4 address of your instance.**
  Type **i** to edit the file.



```
server {
    listen 80;
    server_name 3.82.172.147;
    root /var/www/demo;

    index index.html index.htm index.php;

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

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

    location ~ /\.ht {
        deny all;
    }
}
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
-- INSERT --
```

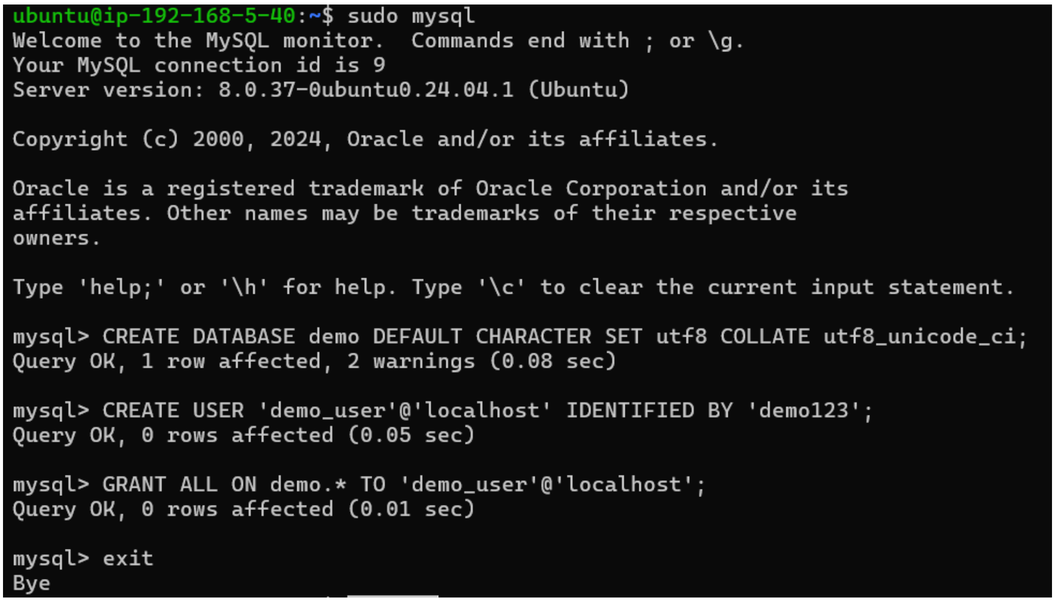
Press ESC, then `:wq!` to save the changes.

- Link to the configuration file from Nginx's sites-enabled directory to activate your setup:
  `sudo ln -s /etc/nginx/sites-available/demo /etc/nginx/sites-enabled/`
- Then, under the /sites-enabled/ directory, unlink the default configuration file:
  `sudo unlink /etc/nginx/sites-enabled/default`
- You can test your configuration for syntax errors by typing:
  `sudo nginx -t`
  If any issues are reported, return to your configuration file and double-check its contents before proceeding.

- When you’re finished, reload Nginx to make the changes take effect:
  `sudo systemctl reload nginx`

Creating a MySQL Database and User for WordPress

- Open the MySQL.
  `sudo mysql`
- Create Database:
  `CREATE DATABASE demo DEFAULT CHARACTER SET utf8 COLLATE utf8_unicode_ci;`
- Create User and Grant Permissions:
  `CREATE USER 'demo_user'@'localhost' IDENTIFIED BY 'demo123';`
  `GRANT ALL ON demo.* TO 'demo_user'@'localhost';`
- Exit when done.
  `EXIT;`



```
ubuntu@ip-192-168-5-40:~$ sudo mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 9
Server version: 8.0.37-0ubuntu0.24.04.1 (Ubuntu)

Copyright (c) 2000, 2024, 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 demo DEFAULT CHARACTER SET utf8 COLLATE utf8_unicode_ci;
Query OK, 1 row affected, 2 warnings (0.08 sec)

mysql> CREATE USER 'demo_user'@'localhost' IDENTIFIED BY 'demo123';
Query OK, 0 rows affected (0.05 sec)

mysql> GRANT ALL ON demo.* TO 'demo_user'@'localhost';
Query OK, 0 rows affected (0.01 sec)

mysql> exit
Bye
```

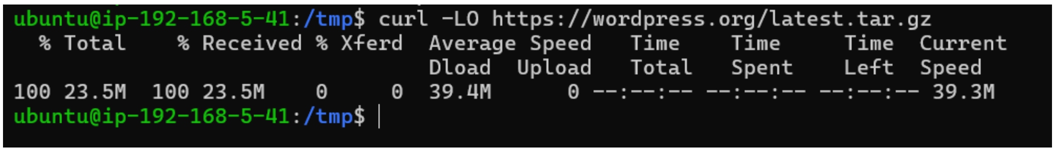
Download and Set Up WordPress

Download WordPress:

- Navigate to the `/tmp` directory.
  `cd /tmp`

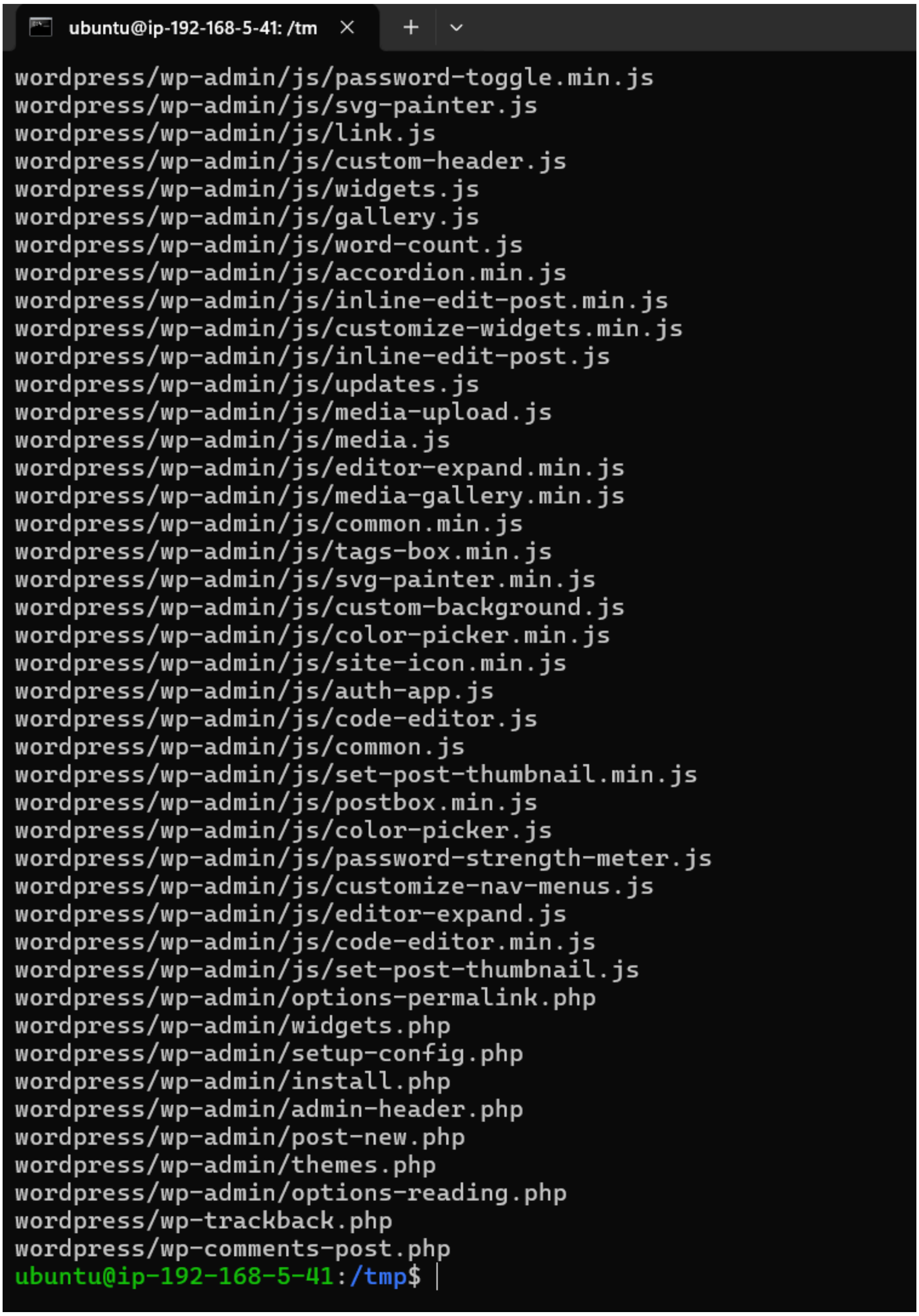
- Download the latest WordPress package, and extract it.

  ```
  curl -LO https://wordpress.org/latest.tar.gz
  ```



```
ubuntu@ip-192-168-5-41:/tmp$ curl -LO https://wordpress.org/latest.tar.gz
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 23.5M  100 23.5M    0     0  39.4M      0 --:--:-- --:--:-- --:--:-- 39.3M
ubuntu@ip-192-168-5-41:/tmp$
```

```
sudo tar -xzvf latest.tar.gz
```



```
wordpress/wp-admin/js/password-toggle.min.js
wordpress/wp-admin/js/svg-painter.js
wordpress/wp-admin/js/link.js
wordpress/wp-admin/js/custom-header.js
wordpress/wp-admin/js/widgets.js
wordpress/wp-admin/js/gallery.js
wordpress/wp-admin/js/word-count.js
wordpress/wp-admin/js/accordion.min.js
wordpress/wp-admin/js/inline-edit-post.min.js
wordpress/wp-admin/js/customize-widgets.min.js
wordpress/wp-admin/js/inline-edit-post.js
wordpress/wp-admin/js/updates.js
wordpress/wp-admin/js/media-upload.js
wordpress/wp-admin/js/media.js
wordpress/wp-admin/js/editor-expand.min.js
wordpress/wp-admin/js/media-gallery.min.js
wordpress/wp-admin/js/common.min.js
wordpress/wp-admin/js/tags-box.min.js
wordpress/wp-admin/js/svg-painter.min.js
wordpress/wp-admin/js/custom-background.js
wordpress/wp-admin/js/color-picker.min.js
wordpress/wp-admin/js/site-icon.min.js
wordpress/wp-admin/js/auth-app.js
wordpress/wp-admin/js/code-editor.js
wordpress/wp-admin/js/common.js
wordpress/wp-admin/js/set-post-thumbnail.min.js
wordpress/wp-admin/js/postbox.min.js
wordpress/wp-admin/js/color-picker.js
wordpress/wp-admin/js/password-strength-meter.js
wordpress/wp-admin/js/customize-nav-menus.js
wordpress/wp-admin/js/editor-expand.js
wordpress/wp-admin/js/code-editor.min.js
wordpress/wp-admin/js/set-post-thumbnail.js
wordpress/wp-admin/options-permalink.php
wordpress/wp-admin/widgets.php
wordpress/wp-admin/setup-config.php
wordpress/wp-admin/install.php
wordpress/wp-admin/admin-header.php
wordpress/wp-admin/post-new.php
wordpress/wp-admin/themes.php
wordpress/wp-admin/options-reading.php
wordpress/wp-trackback.php
wordpress/wp-comments-post.php
ubuntu@ip-192-168-5-41:/tmp$
```

Configure WordPress: Create a `wp-config.php` file with your database details.

- Copy the sample configuration file to `wp-config.php` for customization.

  ```
  sudo cp /tmp/wordpress/wp-config-sample.php /tmp/wordpress/wp-config.php
  ```

- Copy WordPress files to the `/var/www/demo` directory.

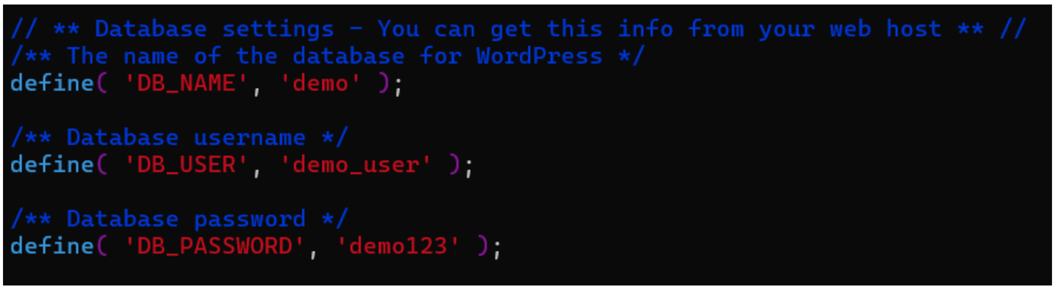
  ```
  sudo cp -a /tmp/wordpress/. /var/www/demo
  ```

- Set the ownership to `www-data`

  ```
  sudo chown -R www-data:www-data /var/www/demo
  ```

- Edit `wp-config.php` to add database details:

  ```
  sudo vi /var/www/demo/wp-config.php
  ```



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

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

/** Database password */
define( 'DB_PASSWORD', 'demo123' );
```

- Press **i** to edit.
- Press ESC, then enter `:wq!` to save the changes.
- Generate security keys:

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

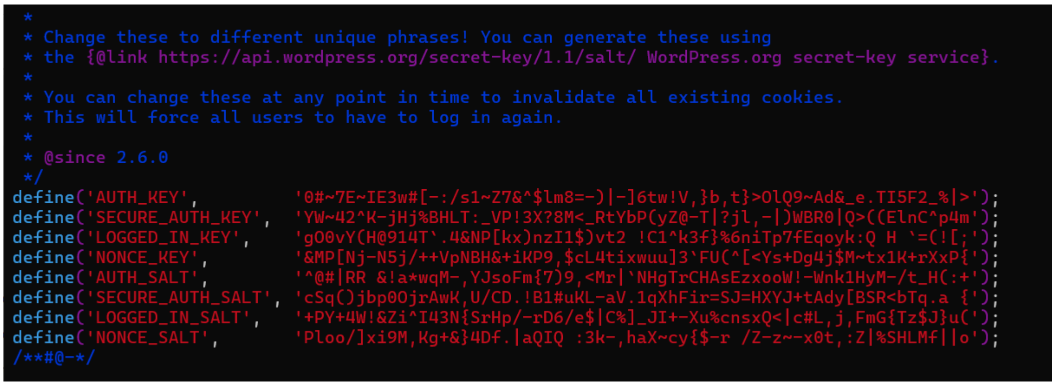
  Copy the generated security keys on a blank notepad.

- Paste the keys into `wp-config.php` file.

  ```
  sudo vi /var/www/demo/wp-config.php
  ```

  Press **i** to edit the file.

  Replace the placeholders with the generated security keys.



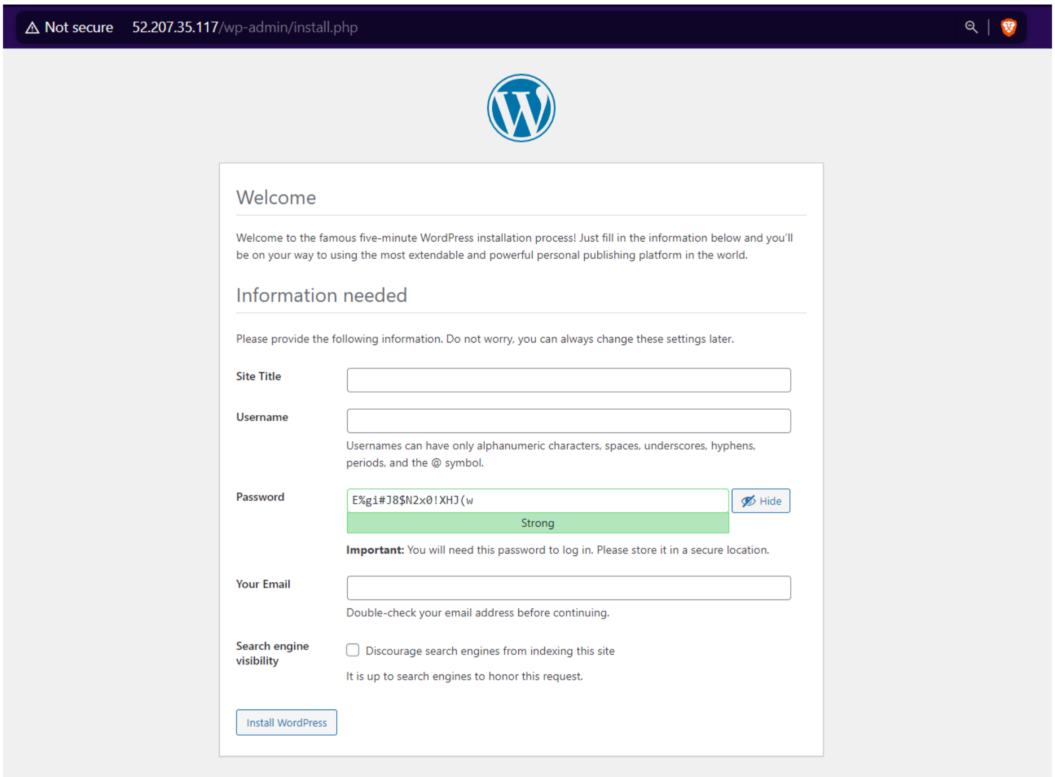
```
 *
 * Change these to different unique phrases! You can generate these using
 * the {@link https://api.wordpress.org/secret-key/1.1/salt/ WordPress.org secret-key service}.
 *
 * You can change these at any point in time to invalidate all existing cookies.
 * This will force all users to have to log in again.
 *
 * @since 2.6.0
 */
define('AUTH_KEY',         '0#~7E~IE3w#[-:/s1~Z7&^$lm8=-)|-]6tw!V,}b,t}>OlQ9~Ad&_e.TI5F2_%|>');
define('SECURE_AUTH_KEY',  'YW~42^K-jHj%BHLT:_VP!3X?8M<_RtYbP(yZ@-T|?jl,-|)WBR0|Q>((ElnC^p4m');
define('LOGGED_IN_KEY',    'gO0vY(H@914T`.4&NP[kx)nzI1$)vt2 !C1^k3f}%6niTp7fEqoyk:Q H `=(![;');
define('NONCE_KEY',        '&MP[Nj-N5j/++VpNBH&+iKP9,$cL4tixwuu]3`FU(^[<Ys+Dg4j$M~tx1K+rXxP{');
define('AUTH_SALT',        '^@#|RR &!a*wqM-,YJsoFm{7)9,<Mr|`NHgTrCHAsEzxooW!-Wnk1HyM-/t_H(:+');
define('SECURE_AUTH_SALT', 'cSq()jbp0OjrAwK,U/CD.!B1#uKL-aV.1qXhFir=SJ=HXYJ+tAdy[BSR<bTq.a {');
define('LOGGED_IN_SALT',   '+PY+4W!&Zi^I43N{SrHp/-rD6/e$|C%]_JI+-Xu%cnsxQ<|c#L,j,FmG{Tz$J}u(');
define('NONCE_SALT',       'Ploo/]xi9M,Kg+&}4Df.|aQIQ :3k-,haX~cy{$-r /Z-z~-x0t,:Z|%SHLMf||o');
/**#@-*/
```

- Press ESC, then enter `:wq!` to save the changes.
- Restart the PHP-FPM

  ```
  sudo systemctl restart php8.3-fpm
  ```

Complete WordPress Installation

- Access your domain in a web browser. `http://IPv4/wordpress` – It will automatically redirect to:



⚠ Not secure 52.207.35.117/wp-admin/install.php

Welcome

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	Usernames can have only alphanumeric characters, spaces, underscores, hyphens, periods, and the @ symbol.
Password	`E%gi#J8$N2x0!XHJ(w` 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.

Install WordPress

- Follow the WordPress setup wizard.

That's it! You've successfully installed WordPress on your Ubuntu EC2 instance using the LEMP stack. Feel free to explore themes and plugins and start creating content!