







Custom INDEX, STYLE, SCRIPT Website

i-076db25640759fd2b (New_Jewel_Store)

Details	Status and alarms	Monitoring	Security	Networking	Storage	Tags
▼ Instance summary Info						
Instance ID  i-076db25640759fd2b	Public IPv4 address  52.23.241.85 open address 			Private IPv4 addresses  172.31.84.90		
IPv6 address —	Instance state  Running			Public DNS  ec2-52-23-241-85.compute-1.amazonaws.com		

Screenshot 2: PHP Installation

```
Last login: Sun Jun 8 12:41:16 on tty000
sadiasarker@sadias-MacBook-Air ~ % ssh -i ~/Downloads/New_Jewel.pem ec2-user@253.241.85

#_#
#####
~\#####\
~~\###|
~~\#|
~~~~V-~-->
      |
      |_____/m/\

Amazon Linux 2023

https://aws.amazon.com/linux/amazon-linux-2023

Last login: Sun Jun 8 18:28:01 2025 from 203.132.71.102
[ec2-user@ip-172-31-84-90 ~]$ cd /var/www/html
sudo mkdir wordpress
cd wordpress
[ec2-user@ip-172-31-84-90 wordpress]$ sudo curl -O https://wordpress.org/latest.tar.gz
sudo tar -xzf latest.tar.gz
sudo mv wordpress/* .
sudo rm -rf wordpress latest.tar.gz

% Total    % Received % Xferd Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left   Speed

100 25.6M 100 25.6M    0     0 42.1M    0 --:--:-- --:--:-- --:--:-- 42.2M
[ec2-user@ip-172-31-84-90 wordpress]$ sudo yum install php php-mysqlnd php-fpm php-json php-mbstring -y
sudo yum install mariadb-server -y
sudo systemctl start mariadb
sudo systemctl enable mariadb
Last metadata expiration check: 2:55:01 ago on Sun Jun 8 16:51:06 2025.
Dependencies resolved.

=====
Package           Arch      Version                Repository            Size
=====
Installing:
php8.4             x86_64    8.4.6-1.amzn2023.0.1  amazonlinux          17 k
php8.4-fpm         x86_64    8.4.6-1.amzn2023.0.1  amazonlinux          1.9 M
php8.4-mbstring    x86_64    8.4.6-1.amzn2023.0.1  amazonlinux          534 k
php8.4-mysqlnd     x86_64    8.4.6-1.amzn2023.0.1  amazonlinux          145 k
Installing dependencies:
libiodbc           x86_64    1.0.19-4.amzn2023     amazonlinux           176 k
libxslt            x86_64    1.1.43-1.amzn2023.0.1 amazonlinux           183 k
nginx-filesystem   noarch    1:1.26.3-1.amzn2023.0.1 amazonlinux           9.6 k
php8.4-cli        x86_64    8.4.6-1.amzn2023.0.1  amazonlinux          3.5 M
php8.4-common     x86_64    8.4.6-1.amzn2023.0.1  amazonlinux          773 k
php8.4-pdo        x86_64    8.4.6-1.amzn2023.0.1  amazonlinux           94 k
php8.4-process    x86_64    8.4.6-1.amzn2023.0.1  amazonlinux           46 k
php8.4-xml        x86_64    8.4.6-1.amzn2023.0.1  amazonlinux          988 k
Installing weak dependencies:
php8.4-opcache     x86_64    8.4.6-1.amzn2023.0.1  amazonlinux          488 k
php8.4-sodium      x86_64    8.4.6-1.amzn2023.0.1  amazonlinux           45 k

Transaction Summary
-----
Install 14 Packages

Total download size: 8.7 M
```

Installing PHP, MariaDB, and required dependencies on Amazon Linux 2023 via SSH connection

Screenshot 3: MySQL Mariadb

```
mariadb105-errmsg-3:10.5.29-1.amzn2023.0.1.x86_64
mariadb105-gssapi-server-3:10.5.29-1.amzn2023.0.1.x86_64
mariadb105-server-3:10.5.29-1.amzn2023.0.1.x86_64
mariadb105-server-utils-3:10.5.29-1.amzn2023.0.1.x86_64
mysql-selinux-1.0.4-2.amzn2023.0.3.noarch
perl-B-1.00-477.amzn2023.0.6.x86_64
perl-DBD-MariaDB-1.22-1.amzn2023.0.4.x86_64
perl-DBI-1.643-7.amzn2023.0.3.x86_64
perl-Data-Dumper-2.174-168.amzn2023.0.2.x86_64
perl-File-Copy-2.34-477.amzn2023.0.6.noarch
perl-FileHandle-2.03-477.amzn2023.0.6.noarch
perl-Math-BigInt-1.1.9998.39-2.amzn2023.0.2.noarch
perl-Math-BigRat-0.2624-500.amzn2023.0.2.noarch
perl-Math-Complex-1.59-477.amzn2023.0.6.noarch
perl-Sys-Hostname-1.23-477.amzn2023.0.6.x86_64
perl-base-2.27-477.amzn2023.0.6.noarch

Complete!
[ec2-user@ip-172-31-84-90 wordpress]$ sudo systemctl start mariadb
sudo systemctl enable mariadb
Created symlink /etc/systemd/system/mysql.service → /usr/lib/systemd/system/mariadb.service.
Created symlink /etc/systemd/system/mysqld.service → /usr/lib/systemd/system/mariadb.service.
Created symlink /etc/systemd/system/multi-user.target.wants/mariadb.service → /usr/lib/systemd/system/mariadb.service.
[ec2-user@ip-172-31-84-90 wordpress]$ sudo mysql_secure_installation

NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MariaDB
SERVERS IN PRODUCTION USE! PLEASE READ EACH STEP CAREFULLY!

In order to log into MariaDB to secure it, we'll need the current
password for the root user. If you've just installed MariaDB, and
haven't set the root password yet, you should just press enter here.

Enter current password for root (enter for none):
OK, successfully used password, moving on...

Setting the root password or using the unix_socket ensures that nobody
can log into the MariaDB root user without the proper authorisation.

You already have your root account protected, so you can safely answer 'n'.

Switch to unix_socket authentication [Y/n] y
Enabled successfully!
Reloading privilege tables..
... Success!

You already have your root account protected, so you can safely answer 'n'.

Change the root password? [Y/n] n
... skipping.

By default, a MariaDB installation has an anonymous user, allowing anyone
to log into MariaDB 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.
```

Securing and starting the MariaDB database server using MySQL Secure Installation

Manual WordPress Site Setup

For my second website, I wanted to challenge myself by manually deploying WordPress on my AWS EC2 instance instead of using a prebuilt solution. This process involved several technical steps, including setting up

the environment, configuring the database, and troubleshooting errors until the site finally went live.

Launching the EC2 Server

I used Amazon EC2 to launch a Linux instance and connected to it via SSH using my PEM key. I verified that the connection was successful by checking the welcome message and Linux version in the terminal.

Preparing the Environment

Inside the EC2 terminal, I installed Apache as the web server, and then installed PHP and its dependencies like php-mysql, php-json, and php-fpm. I also installed MariaDB for database management.

Installing and Securing MariaDB

After installing MariaDB, I started and enabled the mariadb service. I ran the secure installation script to remove test users, disable remote root login, and clean up the default database settings. This was important to make sure the database was secure before connecting it to WordPress.

Creating the WordPress Database

Using the MySQL terminal, I created a new database called wordpress, and added a user named wpuser with the password wppassword. I granted full privileges to this user on the WordPress database and then flushed the privileges to apply the changes.

Downloading and Configuring WordPress

I downloaded the latest WordPress files using the terminal and extracted them into the /var/www/html/wordpress directory. I copied the sample configuration file and renamed it wp-config.php, then manually updated the database name, username, and password to match the ones I had set in MySQL.

Fixing Permissions

To make sure WordPress could access its files properly, I changed ownership of the WordPress directory to the Apache user (apache:apache) using the chown command.

Going Live

After restarting Apache, I opened the browser and navigated to <http://52.23.241.85/wordpress>. The WordPress setup screen loaded successfully, which confirmed the database connection was working. I completed the installation by setting up the site title, username, and password through the browser.

I didn't edit this website as much as it was raw html setup and honestly it was a bit hard to navigate through it. I still added my store name added sparkles but no images. I connected WordPress to this site and had a new website layout and added images and made another store there as it was easier for me to navigate through.