



REPORT

Monolithic Architecture:-

1.1 Create EC2 instance, deploy WordPress and MYSQL on the same instances

Step1: Create an Amazon ec2 free tier account <https://console.aws.amazon.com>

Step2: Go to EC2 Dashboard

Step3: Launch t2.micro instance (Enter instance name and select linux image)

Name and tags [Info](#)

Name

wp-mysql [Add additional tags](#)

▼ **Application and OS Images (Amazon Machine Image)** [Info](#)

Linux Image:-

Recents | **Quick Start**

Amazon Linux macOS **Ubuntu** Windows Red Hat SUSE Li

aws Mac ubuntu® Microsoft Red Hat SUS

Amazon Machine Image (AMI)

Ubuntu Server 22.04 LTS (HVM), SSD Volume Type **Free tier eligible**

ami-0f5ee92e2d63afc18 (64-bit (x86)) / ami-077053fb4029de92f (64-bit (Arm))

Virtualization: hvm ENA enabled: true Root device type: ebs

Step4: Make Sure the instance type is ec2 t2.micro

▼ Instance type [Info](#)

Instance type

t2.micro

Family: t2 1 vCPU 1 GiB Memory Current generation: true

On-Demand Linux base pricing: 0.0124 USD per Hour

On-Demand Windows base pricing: 0.017 USD per Hour

On-Demand RHEL base pricing: 0.0724 USD per Hour

On-Demand SUSE base pricing: 0.0124 USD per Hour

Free tier eligible

☐ All generations

[Compare instance types](#)

Additional costs apply for AMIs with pre-installed software

▼ Key pair (login) [Info](#)

Step5: Create Private key

Create key pair

Key pair name

Key pairs allow you to connect to your instance securely.

The name can include upto 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](#)

Cancel

Create key pair

Step6: Create Security group with following rules

vpc-016a0182c3293d18b

Subnet [Info](#)

No preference (Default subnet in any availability zone)

Auto-assign public IP [Info](#)

Enable

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-3' with the following rules:



- ☒ Allow SSH traffic from

Helps you connect to your instance

Anywhere
0.0.0.0/0
- ☒ 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. 

Step7: Review Instance and Launch

Number of instances [Info](#)

1

Software Image (AMI)

Canonical, Ubuntu, 22.04 LTS, ...[read more](#)

ami-0f5ee92e2d63afc18

Virtual server type (instance type)



t2.micro

Firewall (security group)

New security group

Storage (volumes)

1 volume(s) - 8 GiB

 Free tier: In your first year 

[Review commands](#)


Step8: Go to Instances, select your instance id and click on connect on top right corner

EC2 > Instances > i-07b390f57783afb93

Instance summary for i-07b390f57783afb93 (wp-mysql) [Info](#)

Updated less than a minute ago

[Refresh](#) [Connect](#) [Instance state ▼](#) [Actions ▼](#)

Instance ID i-07b390f57783afb93 (wp-mysql)	Public IPv4 address 3.108.218.3 open address	Private IPv4 addresses 172.31.35.252
IPv6 address	Instance state  Running	Public IPv4 DNS ec2-3-108-218-3.ap-south-

1.2 Installing WordPress and MySQL on same instance

These are linux terminal command to install packages in machine

Step1: apt update -y

Step2: apt upgrade -y

Step3: To install PHP and Apache, use following command:

```
sudo apt install apache2 \  
ghostscript \  
libapache2-mod-php \  
mysql-server \  
php \  
php-bcmath \  
php-curl \  
php-imagick \  
php-intl \  
php-json \  
php-mbstring \  
php-mysql \  
php-xml \  
php-zip -y
```

Step4: cd /var/www/html/

Step5: rm index.html

Step6: wget <https://wordpress.org/latest.tar.gz>

Step7: tar xvzf latest.tar.gz && mv wordpress/* .

Step8: rm latest.tar.gz

Step9: chown www-data:www-data /var/www/html

Step10: mv wp-config-sample.php wp-config.php

Step11: Configure Mysql Database

To configure WordPress, we need to create MySQL database. Let's do it!

```
sudo mysql -u root
```

```
mysql> CREATE DATABASE wordpress;
```

```
mysql> CREATE USER wordpress@localhost IDENTIFIED BY '<your-password>';
```

```
mysql> GRANT SELECT,INSERT,UPDATE,DELETE,CREATE,DROP,ALTER
```

```
-> ON wordpress.*
```

```
-> TO wordpress@localhost;
```

```
mysql> FLUSH PRIVILEGES;
```

```
mysql> quit
```

Step12: edit wp-config.php to update MySQL database credentials

```
*/

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

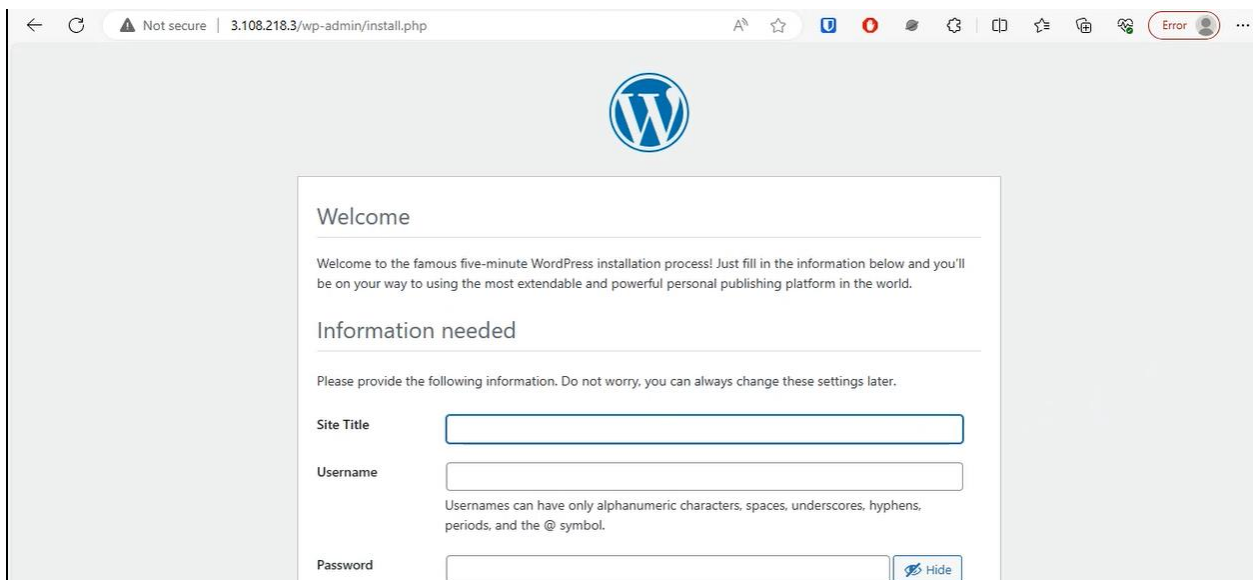
/** Database username */
define( 'DB_USER', 'wp' );

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

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

^G Get Help      ^C Write Out     ^W Where Is
^X Exit          ^R Read File    ^\ Replace
```

Step13: Go to <http://your-server-ip>



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.

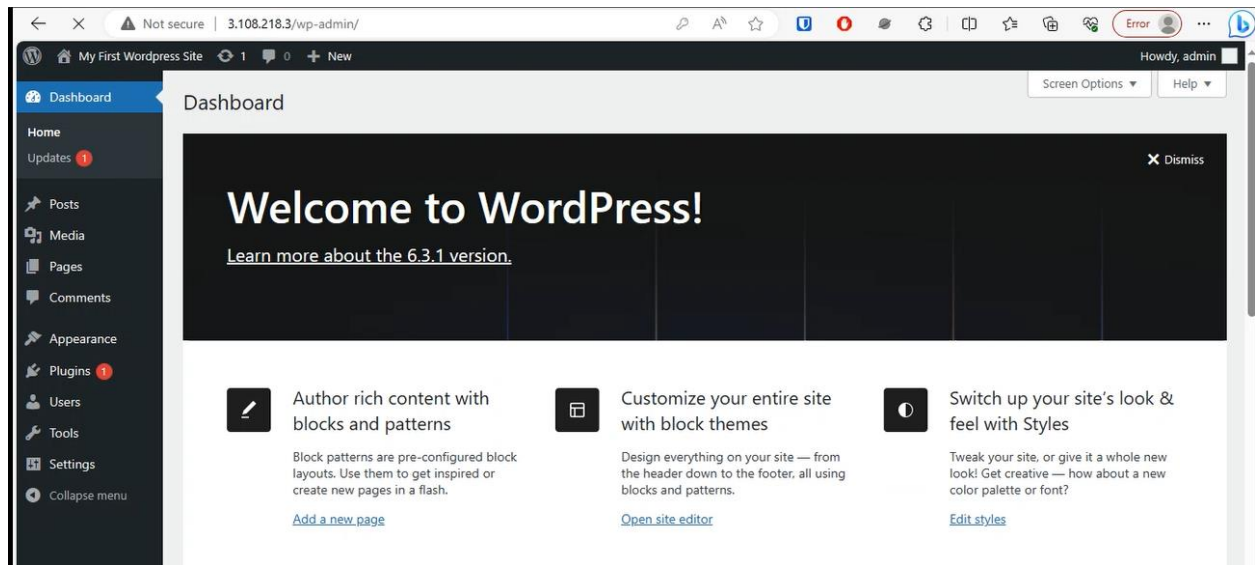
Site Title

Username

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

Password [Hide](#)

Step14: Setup Wordpress and Click on Install



Successfully Installed Wordpress and MySQL Server on same instance.

Microservices

2.1 Create EC2 instance one for WordPress and one for MYSQL

Step1: Repeat 1.1 and create two instance with same type

Step2: Create instances with named as wp-server for wordpress server and another instance with names as mysql-server for remote mysql server.

2.2 Installing WordPress on Instance 1

Step1: apt update -y

Step2: apt upgrade -y

Step3: To install PHP and Apache, use following command:

```
sudo apt install apache2 \  
ghostscript \  
libapache2-mod-php \  
php \  
php-bcmath \  
php-curl \  
php-imagick \  
php-intl \  
php-json \  
php-mbstring \  
php-mysql \  
php-xml \  
php-zip -y
```

Step4: cd /var/www/html/

Step5: rm index.html

Step6: wget <https://wordpress.org/latest.tar.gz>

Step7: tar xvf latest.tar.gz && mv wordpress/* .

Step8: rm latest.tar.gz

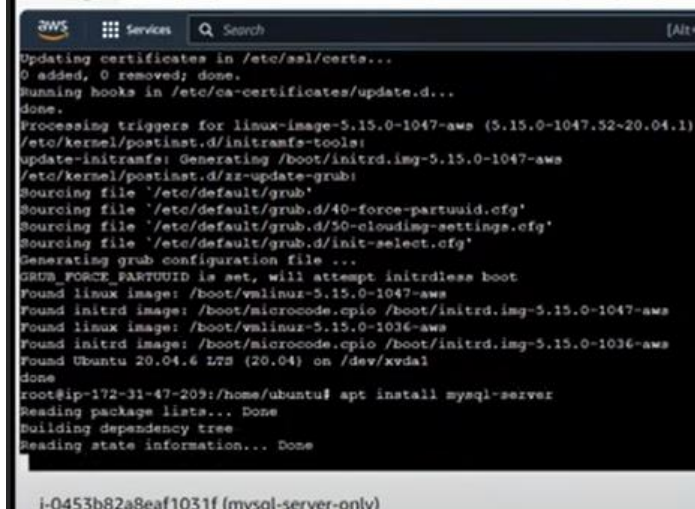
Step9: chown www-data:www-data /var/www/html

Successfully Installed WordPress

2.3 Installing Remote MySQL on Instance 2

Step1: apt update -y && apt upgrade -y

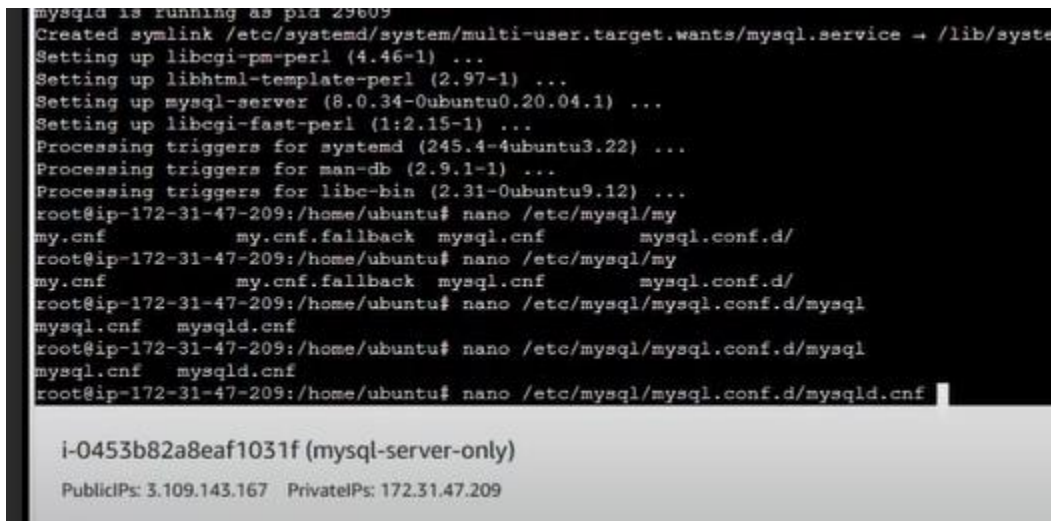
Step2: apt install mysql-server



```
Updating certificates in /etc/ssl/certs...
0 added, 0 removed; done.
Running hooks in /etc/ca-certificates/update.d...
done.
Processing triggers for linux-image-5.15.0-1047-aws (5.15.0-1047.52-20.04.1)
/etc/kernel/postinst.d/initramfs-tools:
update-initramfs: Generating /boot/initrd.img-5.15.0-1047-aws
/etc/kernel/postinst.d/zz-update-grub:
Sourcing file `/etc/default/grub'
Sourcing file `/etc/default/grub.d/40-force-partuuid.cfg'
Sourcing file `/etc/default/grub.d/50-cloudimg-settings.cfg'
Sourcing file `/etc/default/grub.d/init-select.cfg'
Generating grub configuration file ...
GRUB_FORCE_PARTUUID is set, will attempt initrdless boot
Found linux image: /boot/vmlinuz-5.15.0-1047-aws
Found initrd image: /boot/microcode.cpio /boot/initrd.img-5.15.0-1047-aws
Found linux image: /boot/vmlinuz-5.15.0-1036-aws
Found initrd image: /boot/microcode.cpio /boot/initrd.img-5.15.0-1036-aws
Found Ubuntu 20.04.6 LTS (20.04) on /dev/xvda1
done
root@ip-172-31-47-209:/home/ubuntu# apt install mysql-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
```

i-0453b82a8eaf1031f (mysql-server-only)

Step3: edit /etc/mysql/mysql.conf.d/mysqld.conf



```
mysql is running as pid 29609
Created symlink /etc/systemd/system/multi-user.target.wants/mysql.service → /lib/systemd/system/mysql.service
Setting up libcgi-pm-perl (4.46-1) ...
Setting up libhtml-template-perl (2.97-1) ...
Setting up mysql-server (8.0.34-0ubuntu0.20.04.1) ...
Setting up libcgi-fast-perl (1:2.15-1) ...
Processing triggers for systemd (245.4-4ubuntu3.22) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for libc-bin (2.31-0ubuntu9.12) ...
root@ip-172-31-47-209:/home/ubuntu# nano /etc/mysql/my
my.cnf      my.cnf.fallback  mysql.cnf      mysql.conf.d/
root@ip-172-31-47-209:/home/ubuntu# nano /etc/mysql/my
my.cnf      my.cnf.fallback  mysql.cnf      mysql.conf.d/
root@ip-172-31-47-209:/home/ubuntu# nano /etc/mysql/mysql.conf.d/mysql
mysql.cnf    mysqld.cnf
root@ip-172-31-47-209:/home/ubuntu# nano /etc/mysql/mysql.conf.d/mysql
mysql.cnf    mysqld.cnf
root@ip-172-31-47-209:/home/ubuntu# nano /etc/mysql/mysql.conf.d/mysqld.cnf
```

i-0453b82a8eaf1031f (mysql-server-only)

PublicIPs: 3.109.143.167 PrivateIPs: 172.31.47.209

The screenshot shows a terminal window within the AWS Management Console, accessed via a browser at <https://ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh?region=ap-south-1&instance-id=i-0453b82a8eaf1031f&source=instance>. The terminal is running the GNU nano 4.8 editor, editing the file /etc/mysql/my.cnf. The configuration file content is as follows:

```

# Instead of skip-networking the default is now to listen only on
# localhost which is more compatible and is not less secure.
bind-address            = 0.0.0.0
mysqlx-bind-address     = 127.0.0.1
#
# * Fine Tuning
#
key_buffer_size         = 16M
# max_allowed_packet    = 64M
# thread_stack          = 256K
#
# thread_cache_size     = -1

# This replaces the startup script and checks MyISAM tables if needed
# the first time they are touched
myisam-recover-options  = BACKUP

# max_connections       = 151

# table_open_cache      = 4000

```

At the bottom of the terminal, there is a status bar with the following information:

- PublicIPs: 3.109.143.167
- PrivateIPs: 172.31.47.209

Step6: service mysql restart

Step: Configure MySQL Database

```
sudo mysql -u root
mysql> CREATE DATABASE wp;
mysql> CREATE USER wp@'%' IDENTIFIED BY 'password';
mysql> GRANT SELECT,INSERT,UPDATE,DELETE,CREATE,DROP,ALTER
    -> ON wp.*
    -> TO wp@'%';
mysql> FLUSH PRIVILEGES;
mysql> quit
```

```
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 wp;
Query OK, 1 row affected (0.00 sec)

mysql> CREATE USER 'wp'@'%' IDENTIFIED BY 'password';
Query OK, 0 rows affected (0.01 sec)

mysql> GRANT CREATE, ALTER, DROP, INSERT, UPDATE, DELETE, SELECT, REFERENCES, RELOAD on *.* TO 'wp'@'%' WITH GRANT OPTION;
Query OK, 0 rows affected (0.00 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.00 sec)

mysql> exit
Bye
root@ip-172-31-47-209:~#
```

i-0453b82a8eaf1031f (mysql-server-only)

Successfully configured MySQL server for remote login

COMPLETE VIDEO LINK:-

<https://drive.google.com/file/d/1-F2q3P2i-8fo6MPMOnh2Z0OHVqSp2nOE/view?usp=sharing>