

Task:- Deploy application in monolithic and microservices architecture

Here's how I completed the task of deploying WordPress in both monolithic and microservices architectures on AWS EC2 instances.

1.] Monolithic Architecture:-

In the monolithic setup, WordPress and MySQL will run on the same EC2 instance.

Steps:

1. Launch an EC2 Instance:

- Go to the AWS EC2 Dashboard, choose an Ubuntu AMI, select the `t2.micro` instance type, and name it something like `monolithic-wordpress`.

- Create a security group to allow HTTP (port 80) for web access, MySQL (port 3306) from the instance itself, and SSH (port 22) from your IP.

The screenshot displays the AWS Security Groups console. On the left, two security group rules are listed:

- Security group rule 1 (TCP, 22, 0.0.0.0/0)**:
 - Type: ssh
 - Protocol: TCP
 - Port range: 22
 - Source type: Anywhere
 - Source: 0.0.0.0/0
 - Description: e.g. SSH for admin desktop
- Security group rule 2 (TCP, 80, 0.0.0.0/0)**:
 - Type: HTTP
 - Protocol: TCP
 - Port range: 80
 - Source type: Anywhere
 - Source: 0.0.0.0/0
 - Description: e.g. SSH for admin desktop

On the right, the **Summary** panel shows the following configuration:

- Number of instances: 1
- Software Image (AMI): Canonical, Ubuntu, 24.04, amd64...
ami-001f2488b35ca8aad
- Virtual server type (instance type): t2.micro
- Firewall (security group): New security group
- Storage (volumes): 1 volume(s) - 8 GiB

A banner at the bottom indicates: **Free tier:** In your first year includes 750 hours of t2.micro (or

Security group rule 3 (TCP, 3306, 0.0.0.0/0) Remove

Type Info Protocol Info Port range Info

MySQL/Aurora TCP 3306

Source type Info Source Info Description - optional Info

Anywhere ⊞ Add CIDR, prefix list or security group 0.0.0.0/0 × e.g. SSH for admin desktop

⚠ 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. ×

Add security group rule

1

Software Image (AMI)
Canonical, Ubuntu, 24.04, amd64...[read more](#)
ami-001f2488b35ca8aad

Virtual server type (instance type)
t2.micro

Firewall (security group)
New security group

Storage (volumes)
1 volume(s) - 8 GiB

ℹ Free tier: In your first year includes 750 hours of t2.micro (or ×

2. Install Required Software:

- SSH into the instance, update packages, and install Apache for web server, MySQL for the database and PHP.

3. Set Up MySQL:

- Start MySQL, create a database named `wordpress`, and set up a user with access to this database.

4. Install WordPress:

- Download WordPress, move it to the web root directory (`/var/www/html/wordpress`), and configure it to connect to MySQL (using `localhost` as the database host).

5. Finish Setup and Create Welcome Page:

- Access `http://<Instance_Public_IP>/wordpress` in a browser to complete the WordPress setup, then log in to the dashboard, create a Welcome Page, and set it as the homepage.

The commands used are as follows:-

```
aws
Services Q Search [Alt+S] Sydney devops
ubuntu@ip-172-31-8-65:/etc/apache2/sites-available$ history
1 sudo apt update && sudo apt upgrade -y
2 sudo apt install apache2 php php-mysql mysql-server -y
3 sudo systemctl start apache2
4 sudo systemctl enable apache2
5 sudo mysql_secure_installation
6 sudo mysql -u root -p
7 cd /tmp
8 wget https://wordpress.org/latest.tar.gz
9 tar -xzf latest.tar.gz
10 sudo cp -r wordpress/* /var/www/html/
11 cd /var/www/html
12 sudo cp wp-config-sample.php wp-config.php
13 sudo nano wp-config.php
14 sudo chown -R www-data:www-data /var/www/html/
15 sudo systemctl restart apache2
16 cd /var/www/html
17 sudo rm index.html
18 ls /var/www/html
19 sudo chown -R www-data:www-data /var/www/html/
20 sudo chmod -R 755 /var/www/html/
21 sudo systemctl restart apache2
22 ls /var/www/html
23 sudo tar -xzf /tmp/latest.tar.gz -C /var/www/html --strip-components=1
24 sudo chown -R www-data:www-data /var/www/html/
25 sudo chmod -R 755 /var/www/html/
26 sudo systemctl restart apache2
i-0534fa6a61a6e9676 (monolithic)
```

```
aws
Services
Search
[Alt+S]
16 cd /var/www/html
17 sudo rm index.html
18 ls /var/www/html
19 sudo chown -R www-data:www-data /var/www/html/
20 sudo chmod -R 755 /var/www/html/
21 sudo systemctl restart apache2
22 ls /var/www/html
23 sudo tar -xzf /tmp/latest.tar.gz -C /var/www/html --strip-components=1
24 sudo chown -R www-data:www-data /var/www/html/
25 sudo chmod -R 755 /var/www/html/
26 sudo systemctl restart apache2
27 sudo nano /var/www/html/wp-config.php
28 sudo systemctl status mysql
29 mysql -u wpuser -p
30 sudo mysql -u root
31 sudo nano /var/www/html/wp-config.php
32 sudo systemctl restart apache2
33 nano wp-config.php
34 cd /etc/apache2/sites-available/
35 ls
36 nano 000-default.conf
37 sudo nano 000-default.conf
38 sudo systemctl restart apache2
39 sudo nano 000-default.conf
40 sudo systemctl restart apache2
41 history
ubuntu@ip-172-31-8-65:/etc/apache2/sites-available$
i-0534fa6a61a6e9676 (monolithic)
```

It appears like this:-



2.] Microservices Architecture

For the microservices setup, WordPress and MySQL will run on separate EC2 instances.

Steps:

1. Launch EC2 Instances:

- Start two instances:

- One for MySQL with a security group that allows inbound MySQL connections (port 3306) from the WordPress instance's private IP.

Edit inbound rules [Info](#)

Inbound rules control the incoming traffic that's allowed to reach the instance.

Security group rule ID	Type Info	Protocol Info	Port range Info	Source Info	Description - optional Info	
sgr-0525f37fe6cea9a87	SSH	TCP	22	Custom	<input type="text" value="Q"/>	<input type="button" value="Delete"/>
sgr-0a189cda4b2f3b7cd	MySQL/Aurora	TCP	3306	Custom	<input type="text" value="Q"/> <input type="text" value="0.0.0.0/0"/> <input type="text" value="172.31.24.160/32"/>	<input type="button" value="Delete"/>

- One for WordPress with a security group that allows HTTP (port 80) for public web access and SSH (port 22) from your IP.

Edit inbound rules [Info](#)

Inbound rules control the incoming traffic that's allowed to reach the instance.

Security group rule ID	Type Info	Protocol Info	Port range Info	Source Info	Description - optional Info	
sgr-02b279284a16a0201	SSH	TCP	22	My IP	<input type="text" value="Q"/> <input type="text" value="116.73.90.63/32"/>	<input type="button" value="Delete"/>
sgr-058b7e79e3a0d5d69	HTTP	TCP	80	Custom	<input type="text" value="Q"/> <input type="text" value="0.0.0.0/0"/>	<input type="button" value="Delete"/>

2. Install MySQL on the MySQL Instance:

- SSH into the MySQL instance, install MySQL, and configure it to accept remote connections. Create a `wordpress` database and a user for the WordPress instance to connect.

3. Install WordPress on the WordPress Instance:

- SSH into the WordPress instance, install Apache and PHP, download WordPress, and configure it to connect to the MySQL instance by using the MySQL instance's private IP.

4. Finish Setup and Create Welcome Page:

- Go to `http://<WordPress_Instance_Public_IP>/wordpress` in a browser, complete the WordPress setup, create a Welcome Page, and set it as the homepage.



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. Don't 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

Strong

 Hide

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

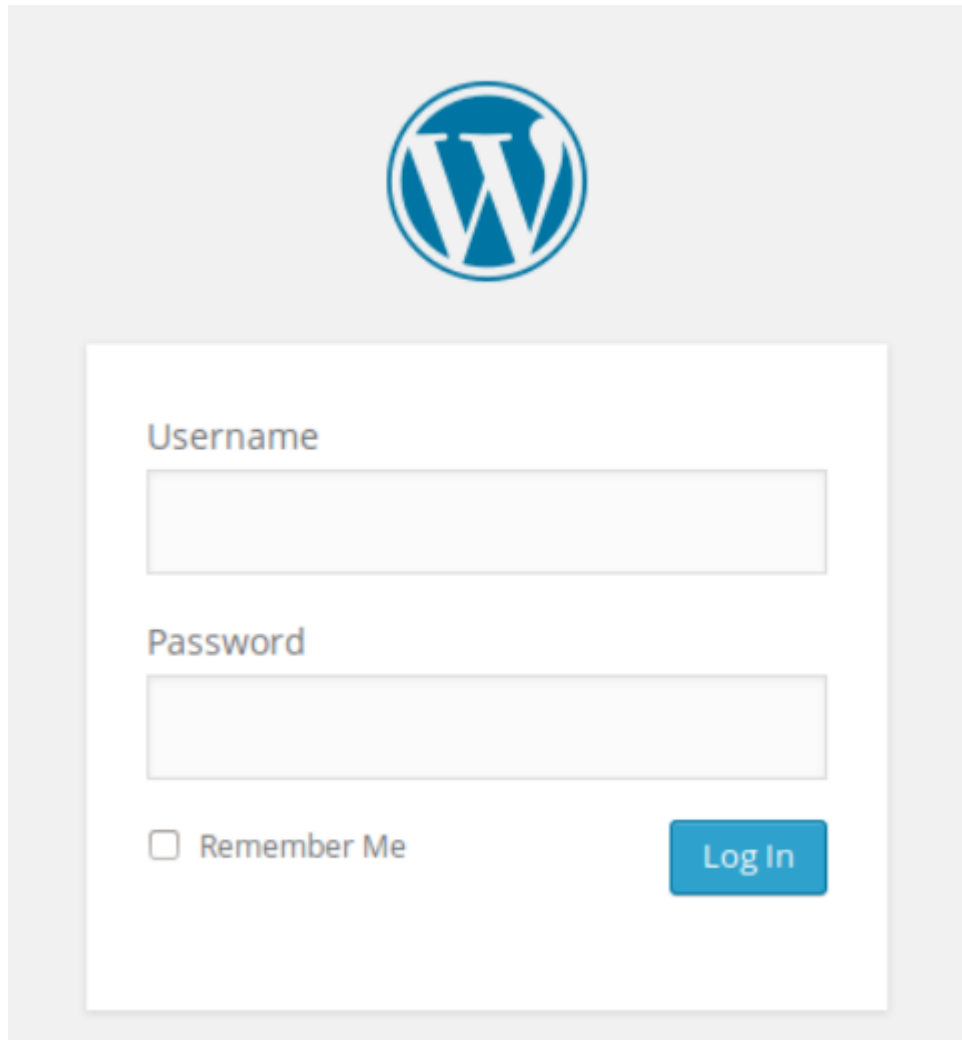
Your Email

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](#)



The commands used are as follows:-

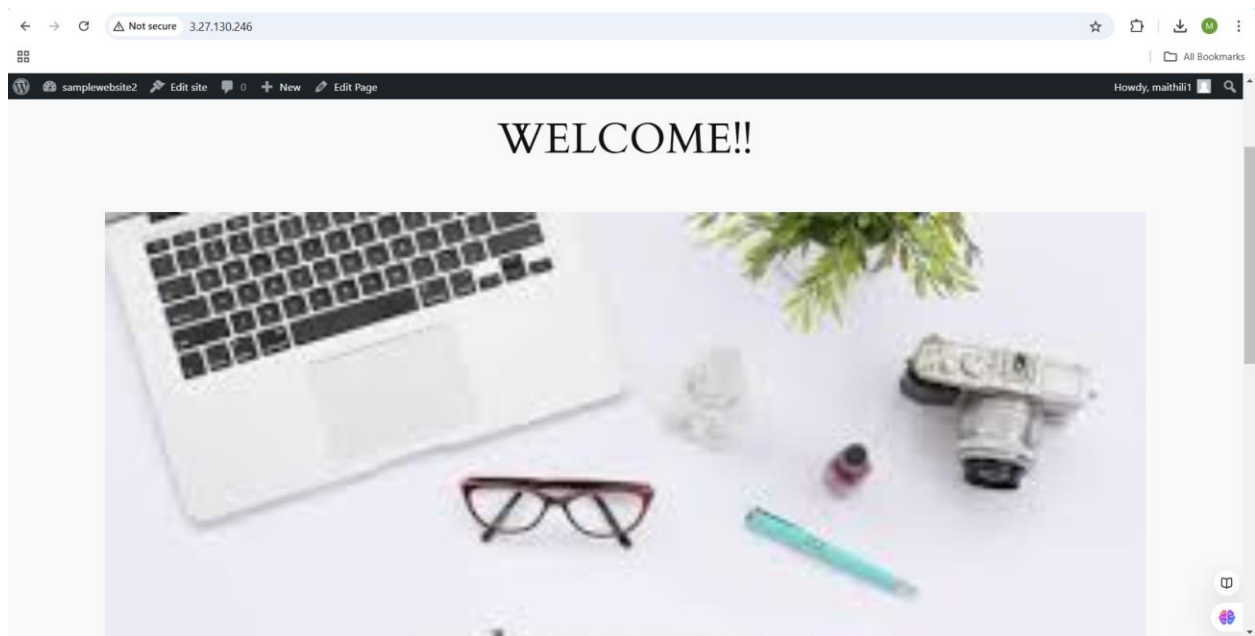
For MySQL instance:-

```
ubuntu@ip-172-31-17-76:~$ history
 1  sudo apt update
 2  sudo apt install mysql-server -y
 3  sudo mysql
 4  sudo nano /etc/mysql/mysql.conf.d/mysqld.cnf
 5  sudo systemctl restart mysql
 6  history
ubuntu@ip-172-31-17-76:~$
```

For Wordpress instance:-

```
ubuntu@ip-172-31-24-160:/var/www/html$ history
 1  sudo mv /var/www/html/wordpress/* /var/www/html/
 2  cd /var/www/html/
 3  sudo rm /var/www/html/index.html
 4  sudo chown -R www-data:www-data /var/www/html
 5  sudo chmod -R 755 /var/www/html
 6  sudo systemctl restart apache2
 7  history
ubuntu@ip-172-31-24-160:/var/www/html$
```

It appears like this:-



- Resources referred to complete the task include:- YouTube tutorial-
<https://youtu.be/8Uofkq718n8?si=GgaeV5LoFh9U19hR>