

# TECHPLEMENT – Cloud AWS Intern

Name – Dhiraj Dinkar Salunke

Github - <https://github.com/Dhiraj4899>

LinkedIn - <https://www.linkedin.com/in/dhirajsalunke18/>

## ➤ For Monolithic -

Certainly! Here are the steps to deploy WordPress and MySQL on the same Ubuntu EC2 instance using a t2.micro instance type:

### Our first step is to create Ubuntu Ec2 Instance on Amazon Cloud (AWS)

Steps to create instance on AWS:

1. Sign in to the AWS Management Console and open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.

Verify that the appropriate Region is selected.

2. Choose Launch Instance.

Complete the following fields:

- a. **Name** – Type a name for the instance.
  - b. **Application and OS Images (Amazon Machine Image)** – Select Ubuntu.
  - c. **Instance type** – Select t2.micro
  - d. **Key pair login** – Create your own key pair.
  - e. **Network settings** – Keep the default(allow SSH, HTTP and HTTPS)
  - f. **Configure storage** – Increase the volume to 256 GiB.
  - g. **Advanced settings** – Keep the default.
3. Launch the instance and SSH into it.

Do the following:

- a. Select Instances in the left navigation, then select the instance ID.
- b. Choose Connect in the top-right.
- c. Choose SSH client and follow the instructions on the screen.

- d. Open a terminal and navigate to the downloaded .pem file (likely in  
~/Downloads).
- e. The first time you follow these procedures, you will receive the message  
"The authenticity of host (...) can't be established." Type yes.

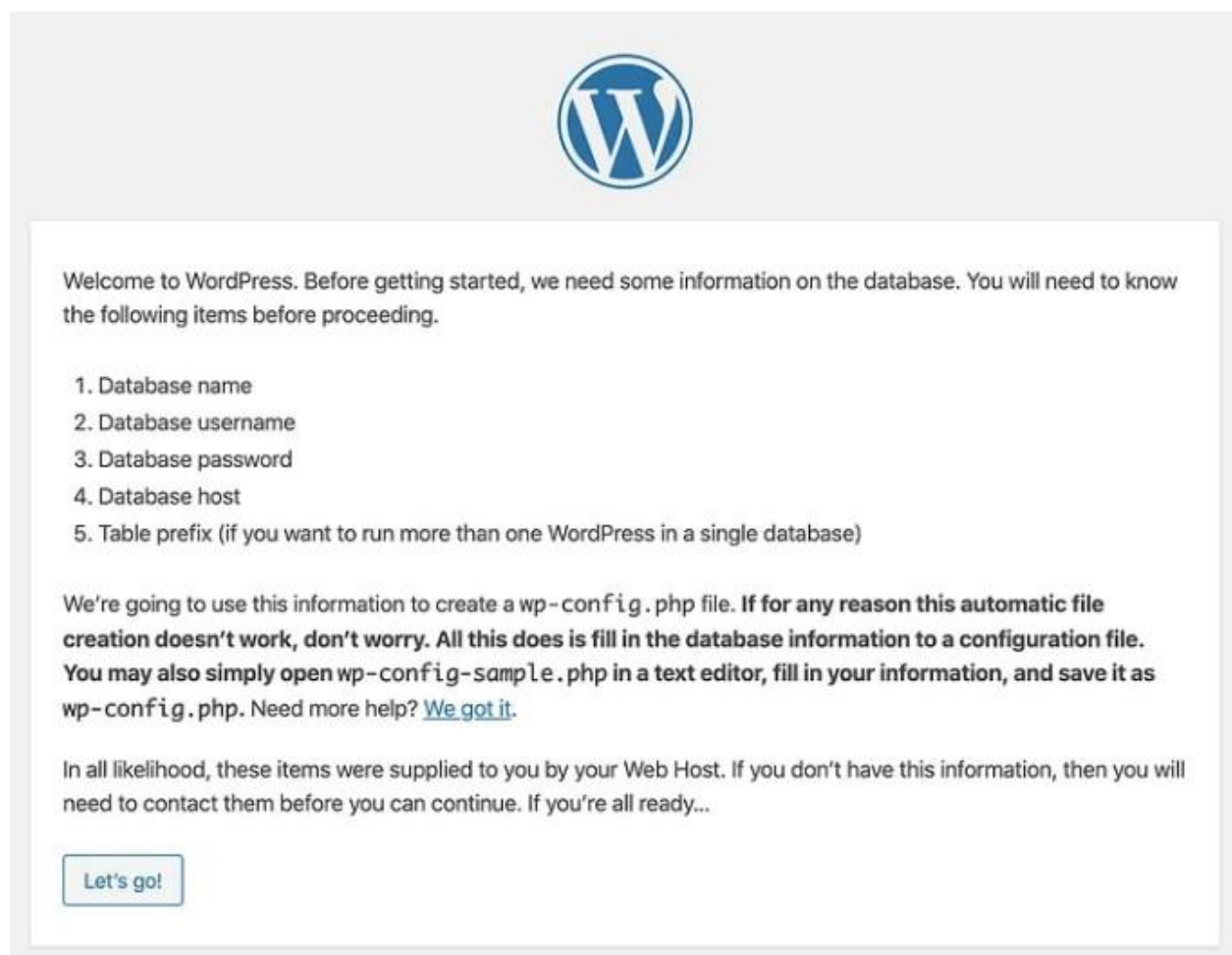
This is how you can create an instance and login into it using SSH.

**Now our very next step is to write the commands to install wordpress and mysql on same instance. And those commands are as below:**

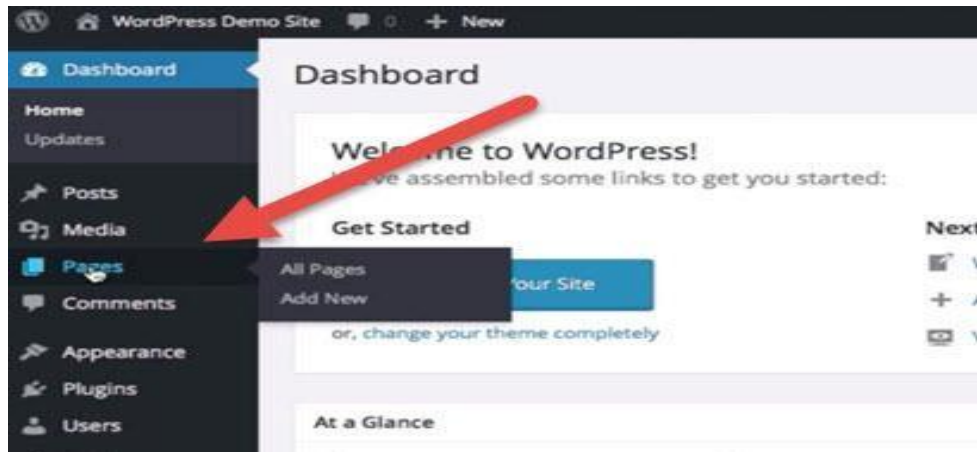
1. Install Apache server on Ubuntu  
apt install apache2
2. Install php runtime and php mysql connector apt install php  
libapache2-mod-php php-mysql
3. Install MySQL server apt install mysql-server
4. Login to MySQL server sudo mysql -u root
5. Change authentication plugin to mysql\_native\_password (change the  
password to something strong)  
ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql\_native\_password by  
'Newpassword@14';
6. Create a new database user for wordpress (change the password to  
something strong)  
CREATE USER 'wp\_user'@localhost IDENTIFIED BY 'Newpassword@14';
7. Create a database for wordpress CREATE DATABASE wp;
8. Grant all privileges on the database 'wp' to the newly created user  
GRANT ALL PRIVILEGES ON wp.\* TO 'wp\_user'@localhost;
9. Download wordpress cd /tmp  
wget https://wordpress.org/latest.tar.gz
10. Unzip  
tar -xvf latest.tar.gz

11. Move wordpress folder to apache document root `sudo mv wordpress/  
/var/www/html`
12. Now change directory to wordpress create a nano file `wp-config.php`  
`cd wordpress/`
13. Create a nano file `wp-config.php` and paste the code in the nano file  
`nano wp-config.php`

If you're now typing your IP address of the instance in your browser, you should see the WordPress installation guide.



Now click on "Let's go!". And now add your user name and password and you can create your own welcome page on wordpress. Now here you can create any web applications according to your requirements.



*This is how your page will look like and by pressing on pages you can create your own web pages and can make your welcome page.*

### ➤ **For microservices:-**

Create two instances using AML: Ubuntu one for Wordpress and one for MYSQL. Configure the necessary security groups and create a welcome page in wordpress.

In these instances we will first create two ubuntu instances one for MYSQL and one for Wordpress. As i have already mentioned the whole process of launching the instances earlier. Now I will directly take you to the command line interface from where we will deploy Wordpress and MYSQL on particular instances.

We have to change the security rules while making instances.

**For MYSQL instance** edit the network settings. Just allow SSH(port 22) and MYSQL/Aurora(3306).

**For Wordpress instance** edit the network settings. Just allow SSH(port 22) and HTTP(80).

Now after these changes just launch your instance and connect those instances using SSH. After connecting those instances now we have to write the code for each instance.

• **For MYSQL instance:** 1. Install Apache server on Ubuntu

apt install apache2

2. Install php runtime and php mysql connector

apt install php libapache2-mod-php php-mysql

3. Install MySQL server apt install mysql-server

mysql-client

4. Login to MySQL server sudo mysql -u root

5. Change authentication plugin to

mysql\_native\_password (change the password  
to something strong)

ALTER USER 'root'@'%' IDENTIFIED WITH mysql\_native\_password by 'Newpassword@123';

6. Create a new database user for wordpress

(change the password to something strong)

CREATE USER 'wp\_user'@localhost IDENTIFIED BY 'Newpassword@123';

7. Create a database for wordpress CREATE

DATABASE wp;

8. Grant all privileges on the database 'wp' to the

newly created user GRANT ALL PRIVILEGES

ON wp.\* TO 'wp\_user'@'%';

9. Configure MySQL (if necessary):

You might need to configure MySQL to allow remote connections if you plan to connect to MySQL from a client outside of your EC2 instance. Edit the MySQL configuration file:

sudo nano /etc/mysql/mysql.conf.d/mysqld.cnf

10. Find the line that says bind-address and change its value to the private IP address of your EC2 instance or 0.0.0.0 to allow connections from any IP address.

Save the file and exit.

Restart MySQL service to apply changes:

sudo systemctl restart mysql

11. To know the private IP of your MYSQL instance just type

curl ifconfig.me

Now this IP will be used when we install wordpress.

You now have MySQL installed and running on your Ubuntu instance in AWS. Make sure to properly configure security groups to control access to your MySQL server.

- **For Wordpress instance:** 1. Install Apache server on Ubuntu apt  
install apache2

2. Install php runtime and php mysql connector  
apt install php libapache2-mod-php php-mysql

3. Download and Configure WordPress (Download the latest version of Wordpress)  
cd /tmp  
wget <https://wordpress.org/latest.tar.gz>

4. Extract the downloaded archive: tar xzvf  
latest.tar.gz

5. Move WordPress files to Apache's root  
directory: sudo mv wordpress/\* /var/www/html/

6. Set appropriate permissions: sudo chown -R  
www-data:www-data /var/www/html/ sudo chmod -  
R 755 /var/www/html/

7. Configure wordpress: sudo mv /var/www/html/wp-config-sample.php  
/var/www/html/wp-config.php

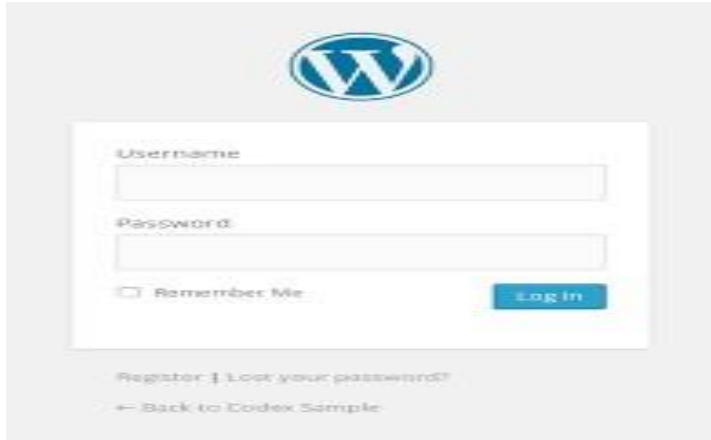
8. Edit the configuration file:  
sudo nano /var/www/html/wp-config.php  
Update the database details (DB\_NAME, DB\_USER, DB\_PASSWORD) with the details you set  
up earlier.

9. Restart apache sudo systemctl restart apache2

Complete WordPress Installation:

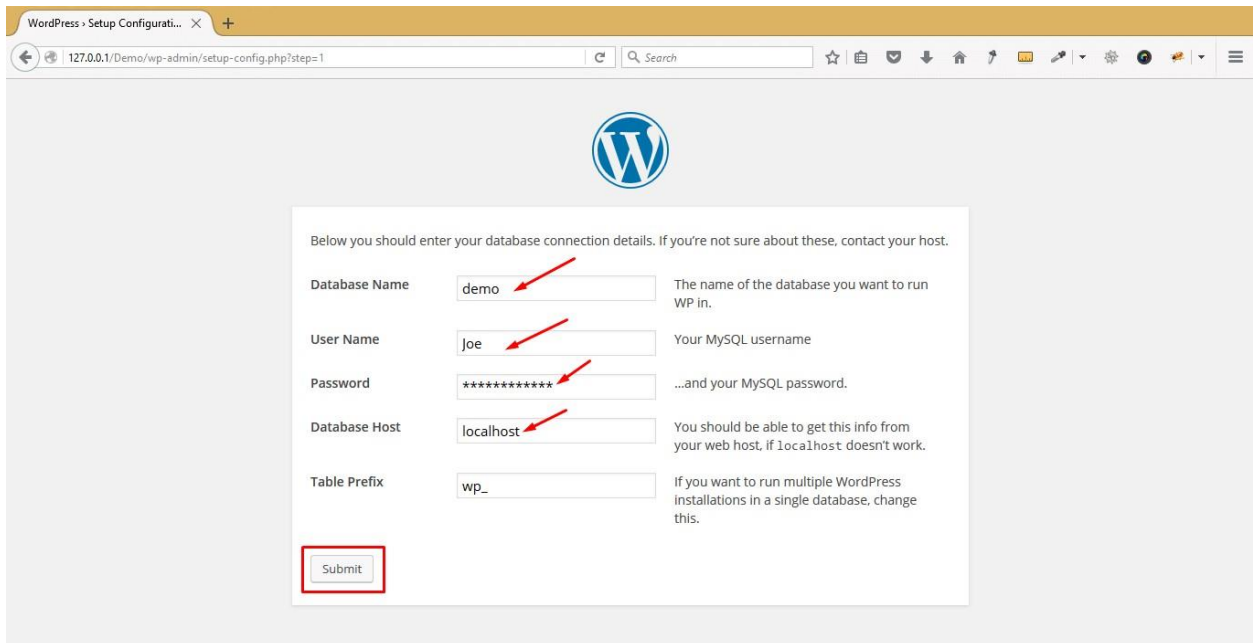
Open your web browser and navigate to your EC2 instance's public IP address.  
Follow the WordPress installation prompts, providing the requested information.

**This is how your wordpress login page looks like**



The image shows the WordPress login interface. At the top is the WordPress logo. Below it is a white box containing a 'Username' input field, a 'Password' input field, a 'Remember Me' checkbox, and a blue 'Log In' button. At the bottom of the box are links for 'Register | Lost your password?' and a link to 'Back to Codex Sample'.

*One thing you need to remember is the IP address we got after using command `curl ifconfig.me` in the sql server. Enter this IP in the Database Host.*



The image shows the WordPress 'Setup Configuration' screen in a web browser. The browser's address bar shows the URL '127.0.0.1/Demo/wp-admin/setup-config.php?step=1'. The page features the WordPress logo and a form for database configuration. The form includes the following fields and instructions:

- Database Name:** 'demo' (with a red arrow pointing to the field). Instruction: 'The name of the database you want to run WP in.'
- User Name:** 'Joe' (with a red arrow pointing to the field). Instruction: 'Your MySQL username'
- Password:** '\*\*\*\*\*' (with a red arrow pointing to the field). Instruction: '...and your MySQL password.'
- Database Host:** 'localhost' (with a red arrow pointing to the field). Instruction: 'You should be able to get this info from your web host, if localhost doesn't work.'
- Table Prefix:** 'wp\_'. Instruction: 'If you want to run multiple WordPress installations in a single database, change this.'

A red box highlights the 'Submit' button at the bottom left of the form.

**Below image is the result of how your welcome page will look like after editing**

# Welcome to Football Hub

Stay updated with the latest football news, match Highlights, and analysis. Your go-to source for everything football!



Explore the world of football and dive into the excitement of the beautiful game.

[Explore Now](#)



