Step to setup wordpress and mysql by microservices architecture

Creating Instances for Wordpress As well AS MySQL

- Choose an AMI in the classic instance wizard: I chose the ubuntu
- Instance details: Select the Instance Type you want to use. Here I am using t2.micro.
- Create a security group allowing all traffic for now.
- Create a new key or choose already existing key (If any).
- Launch your instance.

Setting Up MySQL first:

Step-1: ssh login to ec2

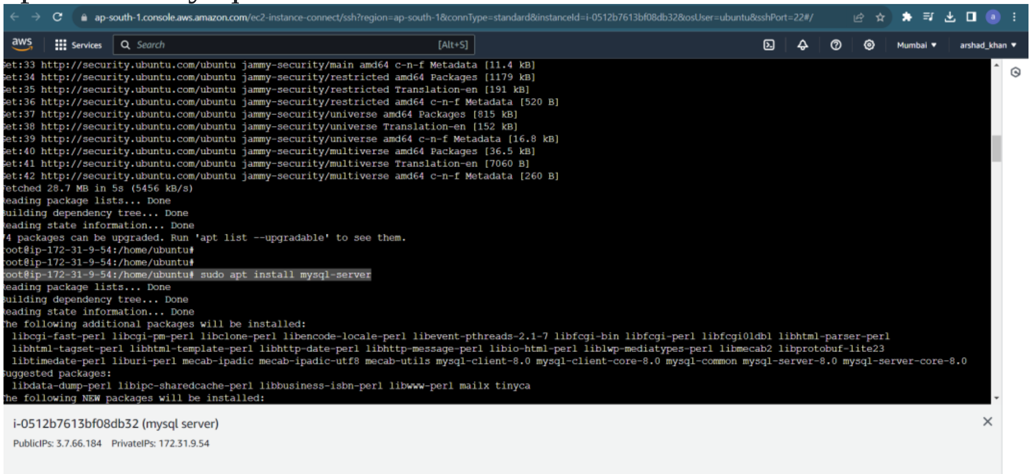
Step-2: Switch to root user

Sudo su – root

Step-3: Install mysql-server

Your WordPress installation needs to store information, such as blog posts and user comments, in a database. This procedure helps you create your blog's database and a user that is authorized to read and save information to it. To install it.

apt install mysql-server



ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh?region=ap-south-1&connType=standard&instanceId=i-0512b7613bf08db32&osUser=ubuntu&sshPort=22#/
aws
Services
Search
[Alt+S]
Mumbai ▾
arshad_khan ▾

```
et:33 http://security.ubuntu.com/ubuntu jammy-security/main amd64 c-n-f Metadata [11.4 kB]
et:34 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 Packages [1179 kB]
et:35 http://security.ubuntu.com/ubuntu jammy-security/restricted Translation-en [191 kB]
et:36 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 c-n-f Metadata [520 B]
et:37 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages [815 kB]
et:38 http://security.ubuntu.com/ubuntu jammy-security/universe Translation-en [152 kB]
et:39 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 c-n-f Metadata [16.8 kB]
et:40 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 Packages [36.5 kB]
et:41 http://security.ubuntu.com/ubuntu jammy-security/multiverse Translation-en [7060 B]
et:42 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 c-n-f Metadata [260 B]
etched 28.7 MB in 5s (5456 kB/s)
eading package lists... Done
uilding dependency tree... Done
eading state information... Done
4 packages can be upgraded. Run 'apt list --upgradable' to see them.
oot@ip-172-31-9-54:/home/ubuntu#
oot@ip-172-31-9-54:/home/ubuntu#
oot@ip-172-31-9-54:/home/ubuntu# sudo apt install mysql-server
eading package lists... Done
uilding dependency tree... Done
eading state information... Done
he following additional packages will be installed:
 libcgi-fast-perl libcgi-pm-perl libclone-perl libencode-locale-perl libevent-pthreads-2.1-7 libfcgi-bin libfcgi-perl libfcgi0ldbl libhtml-parser-perl
 libhtml-tagset-perl libhtml-template-perl libhttp-date-perl libhttp-message-perl libio-html-perl liblwp-mediatypes-perl libmecab2 libprotobuf-lite23
 libtimedate-perl liburi-perl mecab-ipadic mecab-ipadic-utf8 mecab-utils mysql-client-8.0 mysql-client-core-8.0 mysql-common mysql-server-8.0 mysql-server-core-8.0
uggested packages:
 libdata-dump-perl libipc-sharedcache-perl libbusiness-isbn-perl libwww-perl mailx tinyca
he following NEW packages will be installed:
```

i-0512b7613bf08db32 (mysql server)

PublicIPs: 3.7.66.184 PrivateIPs: 172.31.9.54

Step-4: Secure Database server

Run mysql_secure_installation to secure the database using following command:

```
mysql_secure_installation
```

When prompted, type a password for the root account.

1. Type Y to set a password, and type a secure password twice.
2. Type Y to remove the anonymous user accounts.
3. Type N to enable the remote root login.
4. Type Y to remove the test database.
5. Type Y to reload the privilege tables and save your changes.

Step-6: Create an User and Database for Wordpress

```
#login to mysql first
>> mysql -u root
>> CREATE USER 'wordpress-user'@'localhost' IDENTIFIED BY 'abc@123';
```

Note: "abc@123" is a password I set for wordpress-user. You can set your own.

```
>> CREATE DATABASE `wordpress-db`;
>> GRANT ALL PRIVILEGES ON *.* TO 'wordpress-user'@'%' IDENTIFIED
BY 'abc@123' WITH GRANT OPTION;
>> FLUSH PRIVILEGES;
```

The above three commands will create a database with name **wordpress-db** and will grant the privileges for database to wordpress-user even if we login remotely.

Setting Up Wordpress:

Step-1: SSH login to Wordpress's ec2:

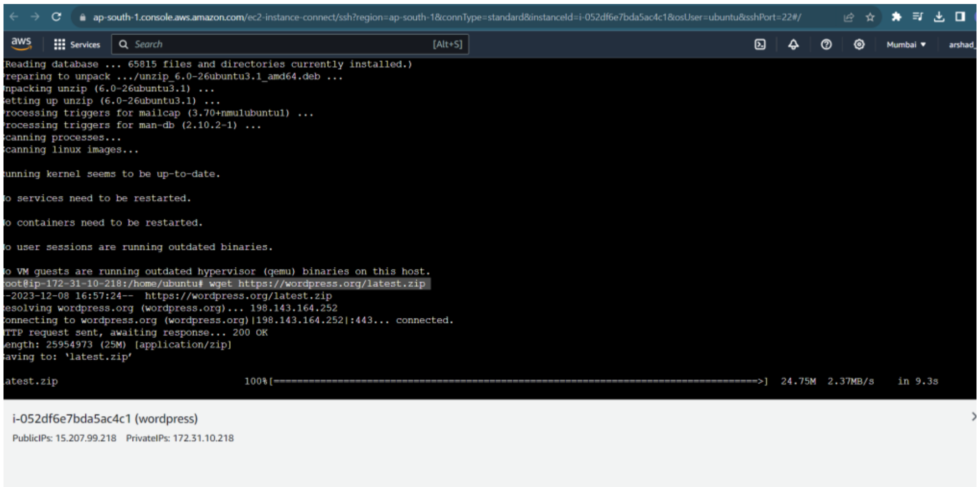
Step-2: Switch user to root:

```
sudo su - root
```

Step-3: Download and Extract wordpress package:

Download the latest WordPress installation package with the wget command and Unzip and unarchive the installation package. The installation folder is unzipped to a folder called wordpress.

wget https://wordpress.org/latest.tar.gz
unzip latest.tar.gz



ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh?region=ap-south-1&connType=standard&instanceId=i-052df6e7bda5ac4c1&osUser=ubuntu&sshPort=22#/
aws
Services
Search
[Alt+S]
Mumbai ▾
arshad_

```
(Reading database ... 65815 files and directories currently installed.)
reparing to unpack .../unzip_6.0-26ubuntu3.1_amd64.deb ...
npacking unzip (6.0-26ubuntu3.1) ...
etting up unzip (6.0-26ubuntu3.1) ...
rocessing triggers for mailcap (3.70+nmu1ubuntu1) ...
rocessing triggers for man-db (2.10.2-1) ...
canning processes...
canning linux images...

unning kernel seems to be up-to-date.

o services need to be restarted.

o containers need to be restarted.

o user sessions are running outdated binaries.

o VM guests are running outdated hypervisor (qemu) binaries on this host.
oot@ip-172-31-10-218:/home/ubuntu# wget https://wordpress.org/latest.zip
-2023-12-08 16:57:24--  https://wordpress.org/latest.zip
esolving wordpress.org (wordpress.org)... 198.143.164.252
onnecting to wordpress.org (wordpress.org)|198.143.164.252|:443... connected.
TTP request sent, awaiting response... 200 OK
ength: 25954973 (25M) [application/zip]
aving to: 'latest.zip'

atest.zip                    100%[==================================================================================>]  24.75M  2.37MB/s    in 9.3s
```

i-052df6e7bda5ac4c1 (wordpress)
PublicIPs: 15.207.99.218   PrivateIPs: 172.31.10.218

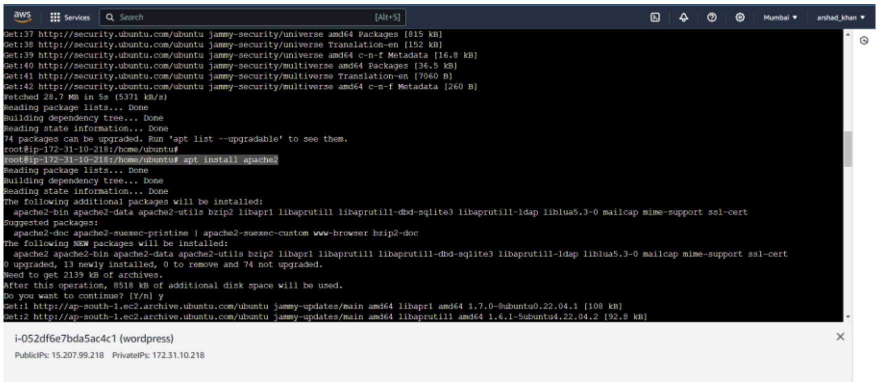
Step-4: Install Apache server and Php:

Apt install apache2

Check if apache service is running or not by public ip of ec2

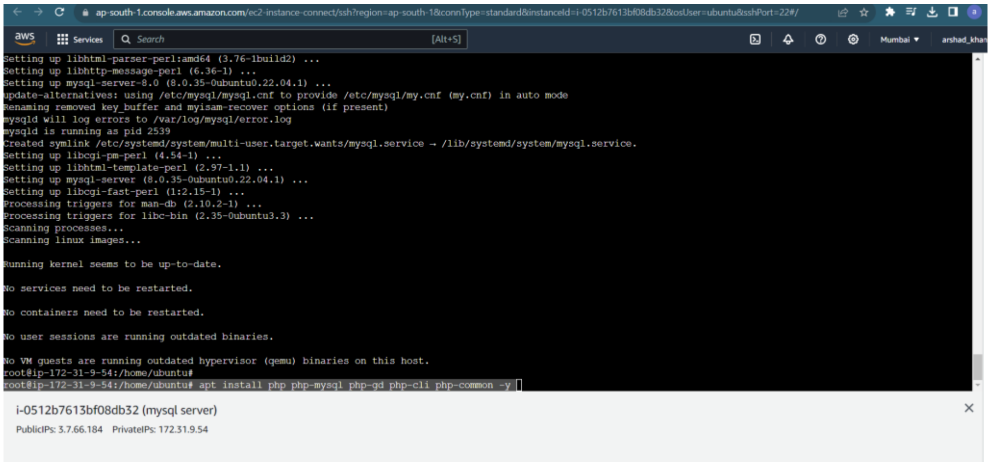
Now install php by below command

Apt install php php-mysql php-gd php-cli php-common –y



```
Get:37 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages [815 kB]
Get:38 http://security.ubuntu.com/ubuntu jammy-security/universe Translation-en [152 kB]
Get:39 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 c-n-f Metadata [16.8 kB]
Get:40 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 Packages [36.5 kB]
Get:41 http://security.ubuntu.com/ubuntu jammy-security/multiverse Translation-en [7060 B]
Get:42 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 c-n-f Metadata [260 B]
Fetched 28.7 MB in 5s (5371 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
74 packages can be upgraded. Run 'apt list --upgradable' to see them.
root@ip-172-31-10-218:/home/ubuntu#
root@ip-172-31-10-218:/home/ubuntu# apt install apache2
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils bzip2 libapr1 libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap liblua5.3-0 mailcap mime-support ssl-cert
Suggested packages:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom www-browser bzip2-doc
The following NEW packages will be installed:
  apache2 apache2-bin apache2-data apache2-utils bzip2 libapr1 libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap liblua5.3-0 mailcap mime-support ssl-cert
0 upgraded, 13 newly installed, 0 to remove and 74 not upgraded.
Need to get 2139 kB of archives.
After this operation, 8518 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libapr1 amd64 1.7.0-8ubuntu0.22.04.1 [108 kB]
Get:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libaprutil1 amd64 1.6.1-5ubuntu4.22.04.2 [92.8 kB]
```

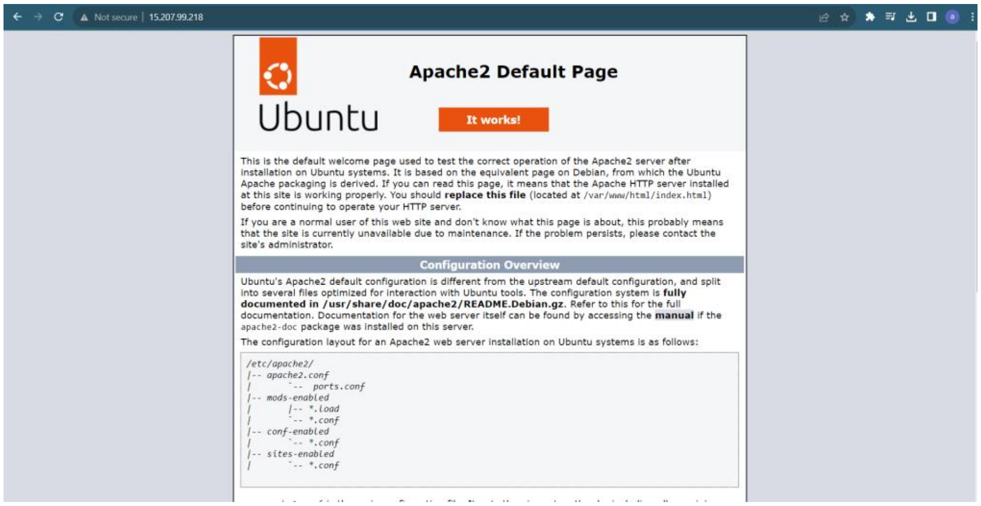
i-052df6e7bda5ac4c1 (wordpress)
PublicIPs: 15.207.99.218   PrivateIPs: 172.31.10.218


ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh?region=ap-south-1&connType=standard&instanceId=i-0512b7613bf08db32&osUser=ubuntu&sshPort=22#/
aws
Services
Search
[Alt+S]
Mumbai
arshad_khan

```
Setting up libhtml-parser-perl:amd64 (3.76-1build2) ...
Setting up libhttp-message-perl (6.36-1) ...
Setting up mysql-server-8.0 (8.0.35-0ubuntu0.22.04.1) ...
update-alternatives: using /etc/mysql/mysql.cnf to provide /etc/mysql/my.cnf (my.cnf) in auto mode
Renaming removed key_buffer and myisam-recover options (if present)
mysqld will log errors to /var/log/mysql/error.log
mysqld is running as pid 2539
Created symlink /etc/systemd/system/multi-user.target.wants/mysql.service → /lib/systemd/system/mysql.service.
Setting up libcgi-pm-perl (4.54-1) ...
Setting up libhtml-template-perl (2.97-1.1) ...
Setting up mysql-server (8.0.35-0ubuntu0.22.04.1) ...
Setting up libcgi-fast-perl (1:2.15-1) ...
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for libc-bin (2.35-0ubuntu3.3) ...
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.
root@ip-172-31-9-54:/home/ubuntu#
root@ip-172-31-9-54:/home/ubuntu# apt install php php-mysql php-gd php-cli php-common -y
```

i-0512b7613bf08db32 (mysql server)

PublicIPs: 3.7.66.184 PrivateIPs: 172.31.9.54



Apache2 Default Page

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at /var/www/html/index.html) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the apache2-doc package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|       `--  ports.conf
|-- mods-enabled
|       |-- *.load
|       `-- *.conf
|-- conf-enabled
|       `-- *.conf
|-- sites-enabled
|       `-- *.conf
```

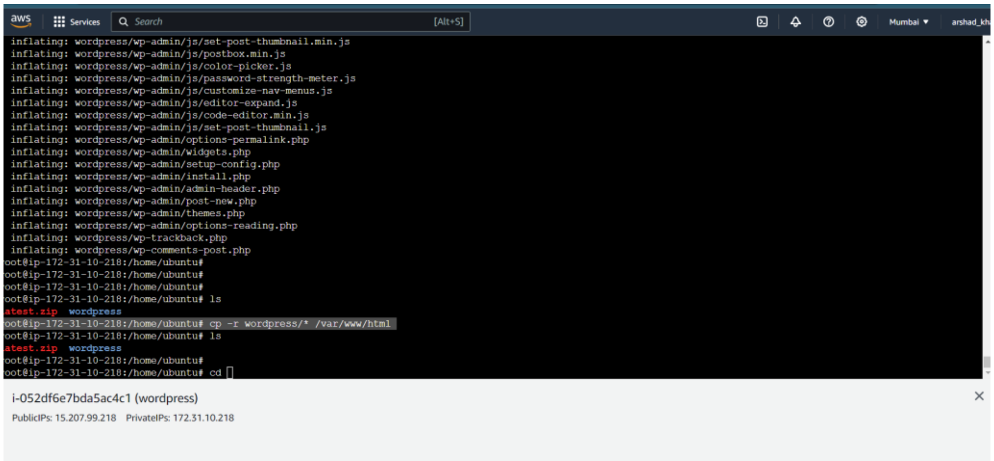
Step-5: Copy website files to /var/www/html:

Run below command for copy wordpress to apache server location

cp -r wordpress/* /var/www/html/

Run below command to remove index.html file

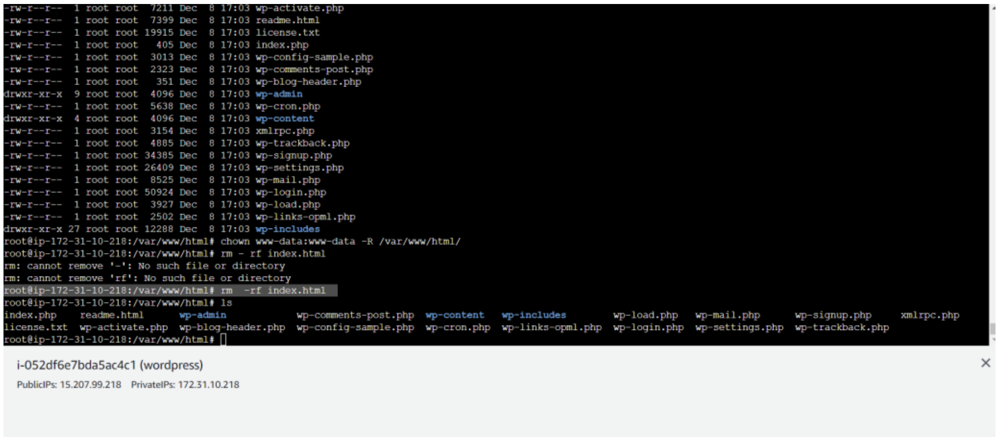
rm –rf index.html



```
 inflating: wordpress/wp-admin/js/set-post-thumbnail.min.js
 inflating: wordpress/wp-admin/js/postbox.min.js
 inflating: wordpress/wp-admin/js/color-picker.js
 inflating: wordpress/wp-admin/js/password-strength-meter.js
 inflating: wordpress/wp-admin/js/customize-nav-menus.js
 inflating: wordpress/wp-admin/js/editor-expand.js
 inflating: wordpress/wp-admin/js/code-editor.min.js
 inflating: wordpress/wp-admin/js/set-post-thumbnail.js
 inflating: wordpress/wp-admin/options-permalink.php
 inflating: wordpress/wp-admin/widgets.php
 inflating: wordpress/wp-admin/setup-config.php
 inflating: wordpress/wp-admin/install.php
 inflating: wordpress/wp-admin/admin-header.php
 inflating: wordpress/wp-admin/post-new.php
 inflating: wordpress/wp-admin/themes.php
 inflating: wordpress/wp-admin/options-reading.php
 inflating: wordpress/wp-trackback.php
 inflating: wordpress/wp-comments-post.php
oot@ip-172-31-10-218:/home/ubuntu#
oot@ip-172-31-10-218:/home/ubuntu#
oot@ip-172-31-10-218:/home/ubuntu#
oot@ip-172-31-10-218:/home/ubuntu# ls
atest.zip  wordpress
oot@ip-172-31-10-218:/home/ubuntu# cp -r wordpress/* /var/www/html
oot@ip-172-31-10-218:/home/ubuntu# ls
atest.zip  wordpress
oot@ip-172-31-10-218:/home/ubuntu#
oot@ip-172-31-10-218:/home/ubuntu# cd
```

i-052df6e7bda5ac4c1 (wordpress)

PublicIPs: 15.207.99.218   PrivateIPs: 172.31.10.218



```
-rw-r--r--  1 root root  7211 Dec  8 17:03 wp-activate.php
-rw-r--r--  1 root root  7399 Dec  8 17:03 readme.html
-rw-r--r--  1 root root 19915 Dec  8 17:03 license.txt
-rw-r--r--  1 root root   405 Dec  8 17:03 index.php
-rw-r--r--  1 root root  3013 Dec  8 17:03 wp-config-sample.php
-rw-r--r--  1 root root  2323 Dec  8 17:03 wp-comments-post.php
-rw-r--r--  1 root root   351 Dec  8 17:03 wp-blog-header.php
drwxr-xr-x  9 root root  4096 Dec  8 17:03 wp-admin
-rw-r--r--  1 root root  5638 Dec  8 17:03 wp-cron.php
drwxr-xr-x  4 root root  4096 Dec  8 17:03 wp-content
-rw-r--r--  1 root root  3154 Dec  8 17:03 xmlrpc.php
-rw-r--r--  1 root root  4885 Dec  8 17:03 wp-trackback.php
-rw-r--r--  1 root root 34385 Dec  8 17:03 wp-signup.php
-rw-r--r--  1 root root 26409 Dec  8 17:03 wp-settings.php
-rw-r--r--  1 root root  8525 Dec  8 17:03 wp-mail.php
-rw-r--r--  1 root root 50924 Dec  8 17:03 wp-login.php
-rw-r--r--  1 root root  3927 Dec  8 17:03 wp-load.php
-rw-r--r--  1 root root  2502 Dec  8 17:03 wp-links-opml.php
drwxr-xr-x 27 root root 12288 Dec  8 17:03 wp-includes
root@ip-172-31-10-218:/var/www/html# chown www-data:www-data -R /var/www/html/
root@ip-172-31-10-218:/var/www/html# rm - rf index.html
rm: cannot remove '-': No such file or directory
rm: cannot remove 'rf': No such file or directory
root@ip-172-31-10-218:/var/www/html# rm  -rf index.html
root@ip-172-31-10-218:/var/www/html# ls
index.php    readme.html      wp-admin            wp-comments-post.php  wp-content   wp-includes        wp-load.php   wp-mail.php      wp-signup.php     xmlrpc.php
license.txt  wp-activate.php  wp-blog-header.php  wp-config-sample.php  wp-cron.php  wp-links-opml.php  wp-login.php  wp-settings.php  wp-trackback.php
root@ip-172-31-10-218:/var/www/html# 
```

i-052df6e7bda5ac4c1 (wordpress)

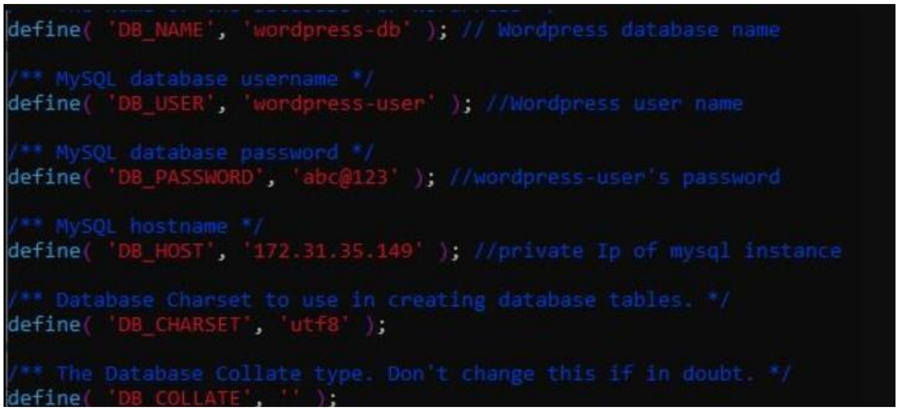
PublicIPs: 15.207.99.218   PrivateIPs: 172.31.10.218

Step-6: Edit wp-config.php file:

Copy the wp-config-sample.php file to a file called wp-config.php. This creates a new configuration file and keeps the original sample file intact as a backup. And edit the same using vim command;

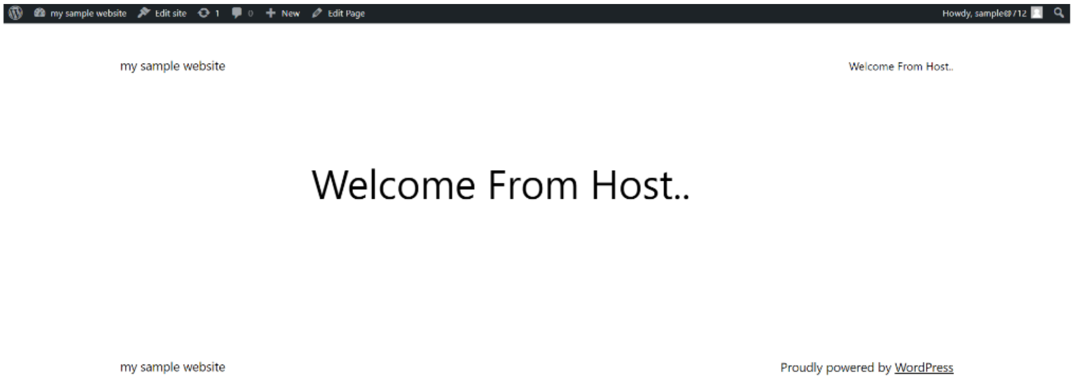
cd /var/www/html
cp wp-config-sample.php wp-config.php
vim wp-config.php

Edit the host, database name, user and password as shown in the following picture:



```
define( 'DB_NAME', 'wordpress-db' ); // Wordpress database name

/** MySQL database username */
define( 'DB_USER', 'wordpress-user' ); //Wordpress user name

/** MySQL database password */
define( 'DB_PASSWORD', 'abc@123' ); //wordpress-user's password

/** MySQL hostname */
define( 'DB_HOST', '172.31.35.149' ); //private Ip of mysql instance

/** Database Charset to use in creating database tables. */
define( 'DB_CHARSET', 'utf8' );

/** The Database Collate type. Don't change this if in doubt. */
define( 'DB_COLLATE', '' );
```

Step-7 Fill the details of database and logging



my sample website

Welcome From Host..

Welcome From Host..

my sample website

Proudly powered by WordPress