

## Deploy WordPress web application by using AWS RDS(MYSQL) service (manually).

- ◆ Now first go to aws account and log into aws and search RDS and click on create database.

The screenshot shows the AWS RDS Management console. The left sidebar has 'Amazon RDS' selected under 'Services'. The main area is titled 'Databases (0)' with a 'Create database' button. A search bar and filter options ('DB identifier', 'Status', 'Role', 'Engine', 'Region ...', 'Size') are at the top. Below, it says 'No instances found'. The bottom right corner shows the AWS CloudShell interface.

- ◆ Now select the database creation method (select standard method) because by selecting easy creation methods then its disables free tier template.

The screenshot shows the 'Create database' wizard. Under 'Choose a database creation method', 'Standard create' is selected. Under 'Engine options', 'MySQL' is selected. On the right, a 'MySQL' section provides information about MySQL and lists its features. The bottom right corner shows the AWS CloudShell interface.

- ◆ Now select the database engine with version but I select the mysql database engine because it's a huge usage in real time organizations.

Screenshot of the AWS RDS MySQL engine selection screen. The MySQL engine is selected. A modal window titled "MySQL" provides details about MySQL, including its popularity, supported features like automated backup and point-in-time recovery, and replication options.

**MySQL**

MySQL is the most popular open source database in the world. MySQL on RDS offers the rich features of the MySQL community edition with the flexibility to easily scale compute resources or storage capacity for your database.

- Supports database size up to 64 TiB.
- Supports General Purpose, Memory Optimized, and Burstable Performance instance classes.
- Supports automated backup and point-in-time recovery.
- Supports up to 15 Read Replicas per instance, within a single Region or 5 read replicas cross-region.

- Now select the template as free tier and by selecting the free tier it disable the availability zone section options.

Screenshot of the AWS RDS database creation wizard. The "Templates" step shows the "Free tier" option selected. The "Availability and durability" step is shown below, with deployment options limited to "Single DB instance". The "Settings" step is partially visible at the bottom.

**Templates**

Choose a sample template to meet your use case.

**Free tier**  
Use RDS Free Tier to develop new applications, test existing applications, or gain hands-on experience with Amazon RDS.

**Availability and durability**

**Deployment options**

The deployment options below are limited to those supported by the engine you selected above.

- Multi-AZ DB Cluster  
Creates a DB cluster with a primary DB instance and two readable standby DB instances, with each DB instance in a different Availability Zone (AZ). Provides high availability, data redundancy and increases capacity to serve read workloads.
- Multi-AZ DB instance (not supported for Multi-AZ DB cluster snapshot)  
Creates a primary DB instance and a standby DB instance in a different AZ. Provides high availability and data redundancy, but the standby DB instance doesn't support connections for read workloads.
- Single DB instance (not supported for Multi-AZ DB cluster snapshot)  
Creates a single DB instance with no standby DB instances.

**Settings**

- ◆ Now give some name to your database and give username and password as credentials for your databases.

The screenshot shows the AWS RDS MySQL creation wizard. On the left, under 'DB instance identifier', the value 'wordpress' is entered. Under 'Master username', 'wordpress' is also entered. There are two options for 'Credentials management': 'Managed in AWS Secrets Manager - most secure' (radio button) and 'Self managed' (radio button, selected). Below these are 'Auto generate password' (unchecked) and 'Master password' fields. On the right, a sidebar titled 'MySQL' provides a brief overview and a bulleted list of features:

- Supports database size up to 64 TiB.
- Supports General Purpose, Memory Optimized, and Burstable Performance instance classes.
- Supports automated backup and point-in-time recovery.
- Supports up to 15 Read Replicas per instance, within a single Region or 5 read replicas cross-region.

- ◆ Now select the storage type as general purpose (SSD (gp2)) and enter the storage values as maximum (1000gb) and minimum (20gb).

The screenshot shows the 'Storage' configuration step. Under 'Storage type', 'General Purpose SSD (gp3)' is selected. In the 'Allocated storage' field, '20' is typed. A note below states: 'After you modify the storage for a DB instance, the status of the DB instance will be in storage-optimization. Your instance will remain available as the storage-optimization operation completes.' Under 'Advanced settings', it says 'Baseline IOPS of 3,000 IOPS and storage throughput of 125 MiBps are included for allocated storage less than 400 GiB.' On the right, the MySQL sidebar remains the same as in the previous step.

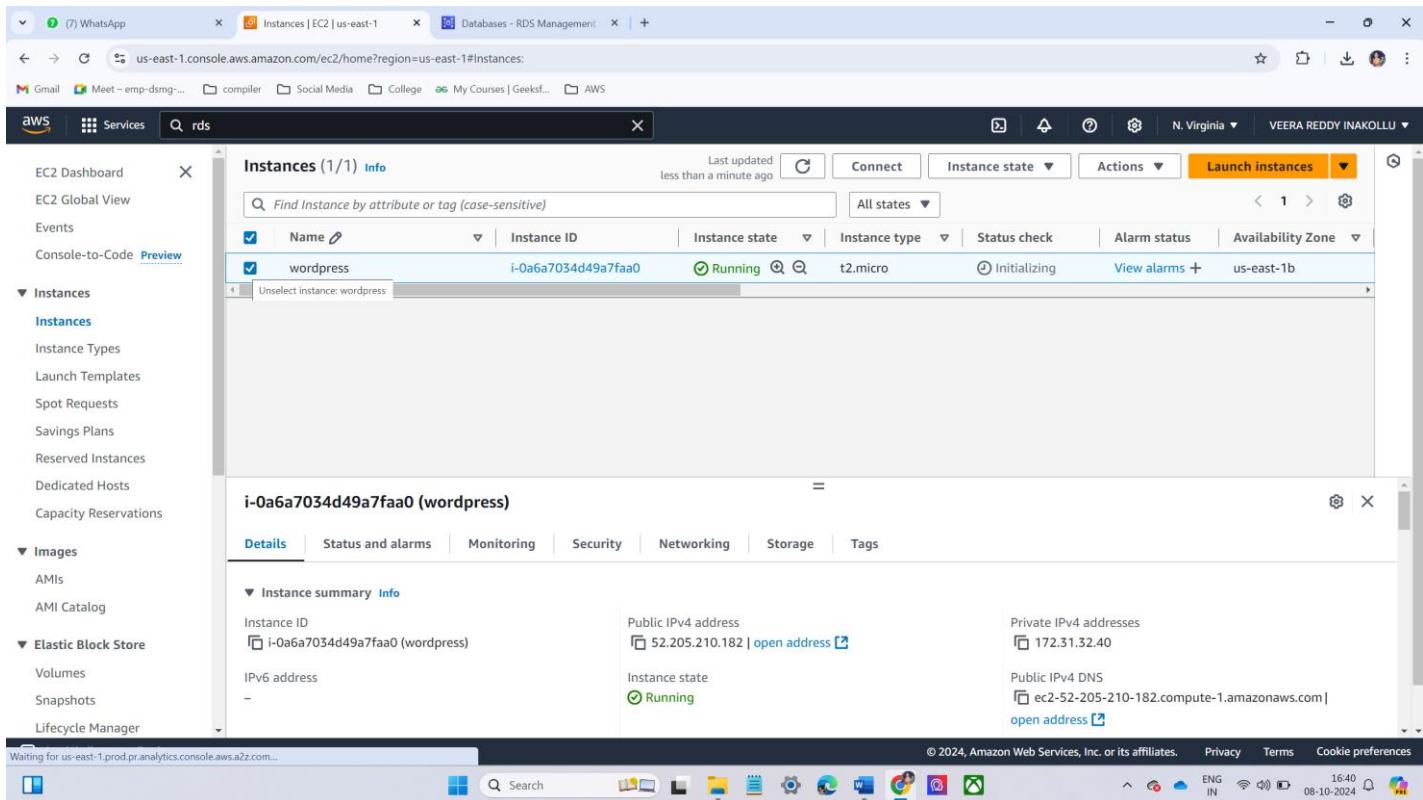
- ◆ Now select your created VPC or select default VPC and it automatically select the database subnet group.

The screenshot shows the AWS RDS Management console with the 'Create database' wizard open. In the 'Virtual private cloud (VPC) Info' section, 'Default VPC (vpc-07c472b50c6ec4ff6)' is selected. Under 'DB subnet group Info', 'default-vpc-07c472b50c6ec4ff6' is chosen. In the 'Public access Info' section, 'No' is selected. For 'VPC security group (firewall) Info', 'Choose existing' is selected. A sidebar on the right titled 'MySQL' lists its features, including support for up to 64 TiB and up to 15 Read Replicas per instance.

- ◆ Now give the name of the database which you give at the stage of DB instance identifier enter same name here.
- ◆ Now click on create database button and will create the MYSQL database.

The screenshot shows the 'Databases' page in the AWS RDS Management console. It lists one database named 'wordpress' with the status 'Available'. The sidebar on the left includes links for Dashboard, Databases, Query Editor, Performance insights, Snapshots, Exports in Amazon S3, Automated backups, Reserved instances, Proxies, Subnet groups, Parameter groups, Option groups, Custom engine versions, Zero-ETL integrations, Events, and Event subscriptions.

- ◆ Now create the ec2 instance by selecting ec2 instance and launch the instance by selecting Amazon Linux -2 version and giving security group with ssh(22) and http(80).



- ◆ Now connect the virtual server through the git bash as shown in fig.

- ◆ Now update the Linux version by using command <sudo yum update -y> and install MySQL by using the command sudo yum install -y mysql2.

The screenshot shows a terminal window titled "ec2-user@ip-172-31-32-40:~" running on Amazon Linux 2. It displays the following output:

```

Amazon Linux 2
AL2 End of Life is 2025-06-30.
A newer version of Amazon Linux is available!
Amazon Linux 2023, GA and supported until 2028-03-15.
https://aws.amazon.com/linux/amazon-linux-2023/

[ec2-user@ip-172-31-32-40 ~]$ sudo yum install mysql
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Resolving Dependencies
--> Running transaction check
--> Package mariadb.x86_64 1:5.5.68-1.amzn2.0.1 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

=====
Package           Arch      Version            Repository      Size
=====
Installing:
mariadb          x86_64   1:5.5.68-1.amzn2.0.1    amzn2-core      8.8 M

Transaction Summary
=====
Install 1 Package

Total download size: 8.8 M
Installed size: 49 M
Is this ok [y/d/N]: y
Downloading packages:
mariadb-5.5.68-1.amzn2.0.1.x86_64.rpm | 8.8 MB 00:00:00
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : 1:mariadb-5.5.68-1.amzn2.0.1.x86_64 1/1
  Verifying  : 1:mariadb-5.5.68-1.amzn2.0.1.x86_64 1/1

Installed:
mariadb.x86_64 1:5.5.68-1.amzn2.0.1

Complete!
[ec2-user@ip-172-31-32-40 ~]$

```

The terminal window also shows the Windows taskbar at the bottom with various icons and the date/time "08-10-2024 16:42".

- ◆ Now to allow create traffic from ec2 instance into rds database for that go to rds service and select your database.
- ◆ Now go inside the created database and go to the security under this option these is security group id click on that.
- ◆ Now access the MySQL database by using the command as “export MYSQL\_HOST <endpoint address>” for that go inside the created database and go to the endpoint address and select and copy the endpoint address
- ◆ Now give the credentials of database by giving the command as “MySQL –user=<username> --password=<password> database=”name”
- ◆ There is another method command to access your database as “mysql -h <endpoint address> -u <user> -p” press enter button it asks the password to enter it, and press enter button
- ◆ Now create a database user for your WordPress application and give it permission to access the WordPress database. By using these commands as
  - CREATE USER wordpress IDENTIFIED BY ‘wordpress-pass’
  - GRANT ALL PRIVILEGE ON wordpress.\* TO wordpress
  - FLUSH PRIVILEGES
  - EXIT

```

ec2-user@ip-172-31-32-40: ~ + - x Transaction Summary
=====
Install 1 Package
Total download size: 8.8 M
Installed size: 49 M
Is this ok [y/d/N]: y
Downloading packages:
mariadb-5.5.68-1.amzn2.0.1.x86_64.rpm | 8.8 MB 00:00:00
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : 1:mariadb-5.5.68-1.amzn2.0.1.x86_64 1/1
  Verifying  : 1:mariadb-5.5.68-1.amzn2.0.1.x86_64 1/1

Installed:
  mariadb.x86_64 1:5.5.68-1.amzn2.0.1

Complete!
[ec2-user@ip-172-31-32-40 ~]$ mysql -h wordpress.cpwlk62gwqgq.us-east-1.rds.amazonaws.com -u wordpress -p
Enter password:
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MySQL connection id is 27
Server version: 8.0.35 Source distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [(none)]> CREATE database wordpress;
Query OK, 1 row affected (0.01 sec)

MySQL [(none)]> CREATE USER 'wordpress'@'localhost' IDENTIFIED BY 'wordpress-pass';
Query OK, 0 rows affected (0.01 sec)

MySQL [(none)]> GRANT ALL PRIVILEGES ON wordpress.* TO 'wordpress'@'localhost';
Query OK, 0 rows affected (0.01 sec)

MySQL [(none)]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.01 sec)

MySQL [(none)]> EXIT;
Bye
[ec2-user@ip-172-31-32-40 ~]$

```



- ◆ To host WordPress application need a Apache web server httpd install that by giving command as <sudo yum install httpd -y> and start and enable the httpd service by giving the command as
  - “sudo service httpd start (or) sudo systemctl start httpd
  - Sudo chkconfig httpd on (or) sudo systemctl enable httpd

```

ec2-user@ip-172-31-32-40: ~ + - x
Installing : apr-util-bdb-1.6.3-1.amzn2.0.1.x86_64 3/9
Installing : httpd-tools-2.4.62-1.amzn2.0.2.x86_64 4/9
Installing : httpd-filesystem-2.4.62-1.amzn2.0.2.noarch 5/9
Installing : generic-logos-httpd-18.0.0-4.amzn2.noarch 6/9
Installing : mailcap-2.1.41-2.amzn2.noarch 7/9
Installing : mod_http2-1.15.19-1.amzn2.0.2.x86_64 8/9
Installing : httpd-2.4.62-1.amzn2.0.2.x86_64 9/9
Verifying : apr-util-bdb-1.6.3-1.amzn2.0.1.x86_64 1/9
Verifying : httpd-2.4.62-1.amzn2.0.2.x86_64 2/9
Verifying : apr-1.7.2-1.amzn2.x86_64 3/9
Verifying : mod_http2-1.15.19-1.amzn2.0.2.x86_64 4/9
Verifying : apr-util-1.6.3-1.amzn2.0.1.x86_64 5/9
Verifying : mailcap-2.1.41-2.amzn2.noarch 6/9
Verifying : generic-logos-httpd-18.0.0-4.amzn2.noarch 7/9
Verifying : httpd-tools-2.4.62-1.amzn2.0.2.x86_64 8/9
Verifying : httpd-filesystem-2.4.62-1.amzn2.0.2.noarch 9/9

Installed:
  httpd.x86_64 0:2.4.62-1.amzn2.0.2

Dependency Installed:
  apr.x86_64 0:1.7.2-1.amzn2.0.1     apr-util.x86_64 0:1.6.3-1.amzn2.0.1     apr-util-bdb.x86_64 0:1.6.3-1.amzn2.0.1     generic-logos-httpd.noarch 0:18.0.0-4.amzn2
  httpd-filesystem.noarch 0:2.4.62-1.amzn2.0.2     httpd-tools.x86_64 0:2.4.62-1.amzn2.0.2     mailcap.noarch 0:2.1.41-2.amzn2     mod_http2.x86_64 0:1.15.19-1.amzn2.0.2

Complete!
[ec2-user@ip-172-31-32-40 ~]$ sudo systemctl start httpd
[ec2-user@ip-172-31-32-40 ~]$ sudo systemctl enable httpd
Created symlink from /etc/systemd/system/multi-user.target.wants/httpd.service to /usr/lib/systemd/system/httpd.service.
[ec2-user@ip-172-31-32-40 ~]$ sudo systemctl status httpd
● httpd.service - The Apache HTTP Server
  Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; vendor preset: disabled)
  Active: active (running) since Tue 2024-10-08 11:16:22 UTC; 18s ago
    Docs: man:httpd.service(8)
  Main PID: 3467 (httpd)
    Status: "Total requests: 0; Idle/Busy workers 100/0;Requests/sec: 0; Bytes served/sec: 0 B/sec"
   CGroup: /system.slice/httpd.service
           ├─3467 /usr/sbin/httpd -DFOREGROUND
           ├─3468 /usr/sbin/httpd -DFOREGROUND
           ├─3469 /usr/sbin/httpd -DFOREGROUND
           ├─3470 /usr/sbin/httpd -DFOREGROUND
           ├─3471 /usr/sbin/httpd -DFOREGROUND
           └─3472 /usr/sbin/httpd -DFOREGROUND

Oct 08 11:16:22 ip-172-31-32-40.ec2.internal systemd[1]: Starting The Apache HTTP Server...
Oct 08 11:16:22 ip-172-31-32-40.ec2.internal systemd[1]: Started The Apache HTTP Server.
[ec2-user@ip-172-31-32-40 ~]$

```



- ◆ Now go to ec2 instance and copy the public ipv4 it on Google. Browse it and check the official page of httpd is displayed

This screenshot shows a browser window with multiple tabs open. The active tab is titled "Test Page for the Apache HTTP" and displays the Apache test page content. The browser's address bar shows the URL "52.205.210.182". The status bar at the bottom indicates the date and time as "08-10-2024 16:47".

This page is used to test the proper operation of the Apache HTTP server after it has been installed. If you can read this page, it means that the Apache HTTP server installed at this site is working properly.

**If you are a member of the general public:**

The fact that you are seeing this page indicates that the website you just visited is either experiencing problems, or is undergoing routine maintenance.

If you would like to let the administrators of this website know that you've seen this page instead of the page you expected, you should send them e-mail. In general, mail sent to the name "webmaster" and directed to the website's domain should reach the appropriate person.

For example, if you experienced problems while visiting [www.example.com](http://www.example.com), you should send e-mail to "webmaster@example.com".

**If you are the website administrator:**

You may now add content to the directory `/var/www/html/`. Note that until you do so, people visiting your website will see this page, and not your content. To prevent this page from ever being used, follow the instructions in the file `/etc/httpd/conf.d/welcome.conf`.

You are free to use the image below on web sites powered by the Apache HTTP Server:



This screenshot shows a Windows desktop environment. The taskbar includes icons for breaking news, Vinesh Phogat, and various system applications. The system tray shows the date and time as "08-10-2024 16:47".

- ◆ Now go to browse and select an download WordPress and click on proper link and select and copy the address link of WordPress. Download file and past that along with wget command in gitbash as “wget <address. Link>”
- ◆ It gives the zip file to unzip that file by using a command as “unzip <zip file>”

```
ec2-user@ip-172-31-32-40: ~ % wget https://wordpress.org/latest.zip
--2024-10-08 16:47:21-- https://wordpress.org/latest.zip
Resolving wordpress.org (wordpress.org)... 151.101.94.142
Connecting to wordpress.org (wordpress.org)|151.101.94.142|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1039911 [application/zip]
Saving to: ‘latest.zip’
latest.zip          100%[=====]  1.04M/s   0:00:01
2024-10-08 16:47:22 (1.04 MB/s) - ‘latest.zip’ saved [1039911/1039911]

[ec2-user@ip-172-31-32-40 ~]$ ls
latest.zip  wordpress
[ec2-user@ip-172-31-32-40 ~]$ |
```

This screenshot shows a terminal window on an Amazon Linux instance. The user has downloaded the latest version of WordPress as a zip file. The terminal prompt is "[ec2-user@ip-172-31-32-40 ~]\$". The desktop environment at the bottom shows the date and time as "08-10-2024 16:47".

- ◆ To run WordPress web application, you must install of WordPress web application as php language with the following commands `<sudo Amazon-linux-extras install -y lamp-mariadb10.2-php7.2.php7.2>` otherwise update ec2 instance with the following commands `<sudo yum update -y>`
- ◆ Now go inside the unzip directory by using command as “`cd <unzip directory>` and change the WordPress configuration file by giving command as “`sudo mv wp-config-sample.php wp-config.php`”

```

[ec2-user@ip-172-31-32-40 ~] ls
inflating: wordpress/wp-admin/js/inline-edit-post.min.js
inflating: wordpress/wp-admin/js/customize-widgets.min.js
inflating: wordpress/wp-admin/js/inline-edit-post.js
inflating: wordpress/wp-admin/js/uploads.js
inflating: wordpress/wp-admin/js/media-upload.js
inflating: wordpress/wp-admin/js/media.js
inflating: wordpress/wp-admin/js/editor-expand.min.js
inflating: wordpress/wp-admin/js/media-gallery.min.js
inflating: wordpress/wp-admin/js/common.min.js
inflating: wordpress/wp-admin/js/tags-box.min.js
inflating: wordpress/wp-admin/js/svg-painter.min.js
inflating: wordpress/wp-admin/js/custom-background.js
inflating: wordpress/wp-admin/js/color-picker.min.js
inflating: wordpress/wp-admin/js/site-icon.min.js
inflating: wordpress/wp-admin/js/auth-app.js
inflating: wordpress/wp-admin/js/code-editor.js
inflating: wordpress/wp-admin/js/common.js
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
[ec2-user@ip-172-31-32-40 ~]$ ls
latest.zip wordpress
[ec2-user@ip-172-31-32-40 ~]$ cd wordpress/
[ec2-user@ip-172-31-32-40 wordpress]$ 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
[ec2-user@ip-172-31-32-40 wordpress]$ sudo mv wp wp-config-sample.php wp-config.php
[ec2-user@ip-172-31-32-40 wordpress]$ 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.php  wp-cron.php  wp-links-opml.php  wp-login.php  wp-settings.php  wp-trackback.php
[ec2-user@ip-172-31-32-40 wordpress]$ 

```

IRE - SA Game score 16:48 IN 08-10-2024

- Now do some configurations in the WordPress configuration file as by giving database name, username, password and hostname for that execute as sudo vi wp-config.php" along with these change the WordPress secret key. To get the secret key browse the Google generate WordPress secret key.

```

[ec2-user@ip-172-31-32-40 ~] ls
/*
** Database settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define( 'DB_NAME', 'wordpress' );

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

/** Database password */
define( 'DB_PASSWORD', 'wordpress-pass' );

/** Database hostname */
define( 'DB_HOST', 'wordpress.cpwk62gwgqsg.us-east-1.rds.amazonaws.com' );

/** 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', '' );

/**#@+
 * Authentication unique keys and salts.
 *
 * Change these to different unique phrases! You can generate these using
 * the {@link https://api.wordpress.org/secret-key/1.1/salt/ WordPress.org secret-key service}.
 *
 * You can change these at any point in time to invalidate all existing cookies.
 * This will force all users to have to log in again.
 *
 * @since 2.6.0
 */
define('AUTH_KEY',         '7jEENyij#t=KtD-Uo,-w=q!AB>y@-pC3j]/_BwFn[Lf=-bMyvl+e%1x-$Z');
define('SECURE_AUTH_KEY',  'ZrM)b=t@-M<`l>x+oxf{l|wyKTd&IyJMK`&[k5cvStAy/]jNUKAlog2s+K');
define('LOGGED_IN_KEY',    'SPtGc?0..HRRF&Di,01[&TXD(jx)-h%Z4X[2?e),j-CxK54g[P]jelQap=C-');
define('NONCE_KEY',        'XA..r>Dyxz'HEPC[lcm[Ikm:QF-D-i7]>bftBDAC274DBL->6MjlxT7+-iOr*89');
define('AUTH_SALT',        'F-J-#tEal-pz:uoBdwOL6)'^+ZVEh3PxYID0Knmgl|Wwp1#H6a-5fY-0<ZdY');
define('SECURE_AUTH_SALT', '@EisM G8yQaAcw$5CH$!IRm=E:f1H0%ox,;aaA?-~y(Cp,--2n;WbSaLRN0t');
define('LOGGED_IN_SALT',   'ulcGR8SwKm@tWL|ctBFxWLEHR0+oImdqF7-Genj+C,6|-7b154'U03e810e$1');
define('NONCE_SALT',       'fglfk;-d>rev-[u$MHIY(GwP[O)RF3Him<rx|[&|wl'V70x|uII>t8CEssPU:<');

/**#@-*/
/**
 * WordPress database table prefix.

```

58, 95 35%  
31°C Sunny 16:51 IN 08-10-2024

- Now copy the WordPress directory to the document root directory to host the web application of WordPress by giving as command as
- Sudo cp -I <unzip file>/\* /var/www/html/
- To restart the httpd server by using command sudo systemctl restart the httpd

```

[ec2-user@ip-172-31-32-40 ~]$ ls
latest.zip wordpress
[ec2-user@ip-172-31-32-40 ~]$ cd wordpress/
[ec2-user@ip-172-31-32-40 wordpress]$ 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
[ec2-user@ip-172-31-32-40 wordpress]$ sudo mv wp-config-sample.php wp-config.php
[ec2-user@ip-172-31-32-40 wordpress]$ 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.php  wp-cron.php  wp-links-opml.php  wp-login.php  wp-settings.php  wp-trackback.php
[ec2-user@ip-172-31-32-40 wordpress]$ 
[ec2-user@ip-172-31-32-40 wordpress]$ sudo vi wp-config.php
"wp-config.php" [dos] 96L, 3342B written
[ec2-user@ip-172-31-32-40 wordpress]$ cd
[ec2-user@ip-172-31-32-40 ~]$ sudo cp -r wordpress/* /var/www/html/
[ec2-user@ip-172-31-32-40 ~]$ cd /var/www/html/
[ec2-user@ip-172-31-32-40 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.php  wp-cron.php  wp-links-opml.php  wp-login.php  wp-settings.php  wp-trackback.php
[ec2-user@ip-172-31-32-40 html]$ sudo systemctl restart httpd
[ec2-user@ip-172-31-32-40 html]$ 

```

- ◆ Now go to ec2 instance and copy public ipv4 and paste it on google browse it and check the official page of WordPress it displays

The screenshot shows a Windows desktop environment with a terminal window and a web browser. The terminal window displays the command-line steps taken to set up WordPress on an AWS EC2 instance. The web browser is open to the WordPress dashboard, which includes a welcome message, three main features (blocks, themes, styles), and a PHP update notice.