

# AWS-DEPLOY WORDPRESS WITH AMAZON RDS

## Lab-1: Creating a MySQL Database with RDS:

1. go to Amazon RDS in the AWS console. Click on the Create database button to get started.

The screenshot shows the AWS RDS Management Console. In the top navigation bar, the 'RDS Management Console' tab is selected. On the left sidebar, under the 'Amazon RDS' heading, the 'Databases' option is selected. A central panel displays information about Amazon Aurora, stating it's a MySQL- and PostgreSQL-compatible enterprise-class database. It includes a 'Create database' button and a note about restoring from S3. Below this, the 'Resources' section lists various Amazon RDS resources: DB Instances (0/40), DB Clusters (0/40), Reserved instances (0/40), Snapshots (10), and Subnet groups (0). The 'Additional information' sidebar on the right provides links to getting started with RDS, documentation, and forums.

2. WordPress uses MySQL, so select that engine now.

The screenshot shows the 'Engine options' selection screen in the AWS RDS Management Console. Under the 'Engine type' heading, the 'MySQL' option is selected. Other options shown include 'Amazon Aurora', 'MariaDB', 'PostgreSQL', 'Oracle', and 'Microsoft SQL Server'. Below the engine selection, the 'Edition' is set to 'MySQL Community'. A 'Known issues/limitations' section at the bottom provides a link to review compatibility issues. The right side of the screen displays a detailed description of Amazon Aurora, highlighting its features like up to 5 times throughput, 128TB storage, and 6-way replication.

3. In the Templates section of the creation wizard, there is an option to only show options that are available in the AWS Free Tier. Select this option now if you would like to use this lab for learning without spending any money.

The screenshot shows the AWS RDS Management Console interface. In the top navigation bar, the URL is <https://us-west-1.console.aws.amazon.com/rds/home?region=us-west-1#launch-dbinstancegdb=false&s3-import=false>. The left sidebar lists services: EC2, VPC, CloudFormation, and RDS. The main area shows the 'Engine Version' set to MySQL 8.0.28. Under 'Templates', the 'Free tier' option is selected, described as being intended for development and testing. The 'Settings' section includes fields for the DB instance identifier ('database-1') and the master username ('admin'). On the right, the 'Amazon Aurora' sidebar provides a detailed description of the service, mentioning its MySQL and PostgreSQL compatibility, pricing starting at \$1/day, and various performance and replication features.

4. Next, you will specify the authentication settings for your MySQL deployment. This includes the database name and the master username and password.

This screenshot continues the process of setting up a MySQL instance. The 'Master password' field has been populated with the value '\*\*\*\*\*'. The rest of the interface and sidebar are identical to the previous screenshot, showing the 'Settings' section and the 'Amazon Aurora' sidebar.

- After setting your username and password, you can select key details about your MySQL deployment. This includes the instance class and storage details.

**Instance configuration**

The DB instance configuration options below are limited to those supported by the engine that you selected above.

**DB instance class** [Info](#)

- Standard classes (includes m classes)
- Memory optimized classes (includes r and x classes)
- Burstable classes (includes t classes)**

db.t2.micro  
1 vCPU 1 GiB RAM Not EBS Optimized

Include previous generation classes

**Storage**

**Storage type** [Info](#)  
General Purpose SSD (gp2)  
Baseline performance determined by volume size

Allocated storage  
20 GiB  
The minimum value is 20 GiB and the maximum value is 6,144 GiB

**Storage autoscaling** [Info](#)

Amazon Aurora

Amazon Aurora is a MySQL and PostgreSQL compatible enterprise-class database, starting at <\$1/day.

- Up to 5 times the throughput of MySQL and 3 times the throughput of PostgreSQL.
- Up to 128TB of auto-scaling SSD storage
- 6-way replication across three Availability Zones
- Up to 15 Read Replicas with sub-10ms replica lag
- Automatic monitoring and failover in less than 30 seconds

- Click on the Additional configuration line to expand the options.
- Set the initial database name to WordPress. This will ensure RDS creates the database in your MySQL instance upon initialization. We will use this database name when connecting to your database

**Additional configuration**

Database options

**Initial database name** [Info](#)  
wordpress  
If you do not specify a database name, Amazon RDS does not create a database.

**DB parameter group** [Info](#)  
default.mysql8.0

**Option group** [Info](#)  
default.mysql8-0

**Backup**

Enable automated backups  
Creates a point-in-time snapshot of your database

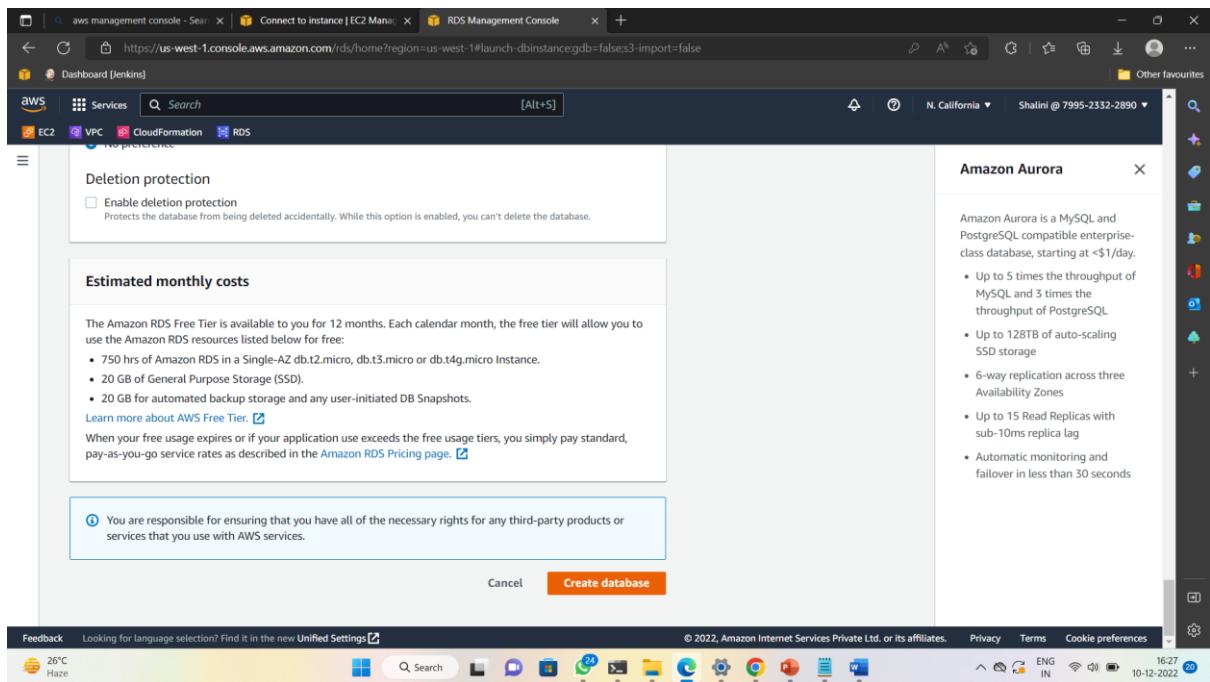
**Backup retention period** [Info](#)  
The number of days (1-35) for which automatic backups are kept.

**Amazon Aurora**

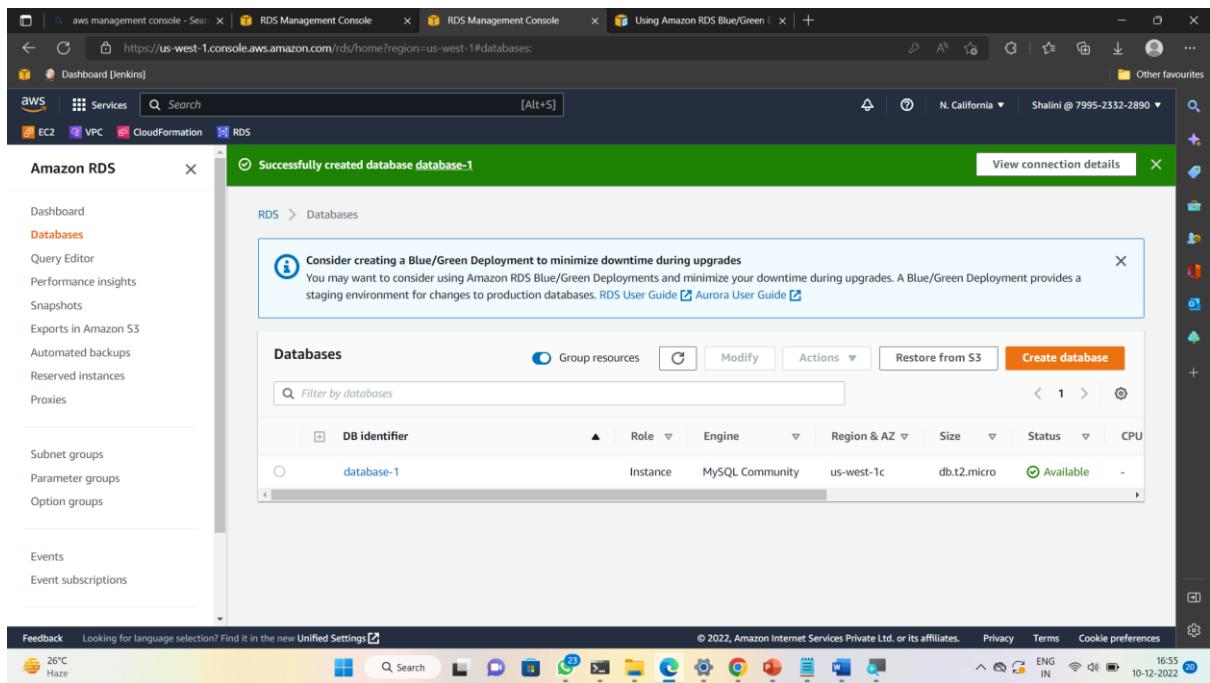
Amazon Aurora is a MySQL and PostgreSQL compatible enterprise-class database, starting at <\$1/day.

- Up to 5 times the throughput of MySQL and 3 times the throughput of PostgreSQL.
- Up to 128TB of auto-scaling SSD storage
- 6-way replication across three Availability Zones
- Up to 15 Read Replicas with sub-10ms replica lag
- Automatic monitoring and failover in less than 30 seconds

- Click on the Create database button to create your database.

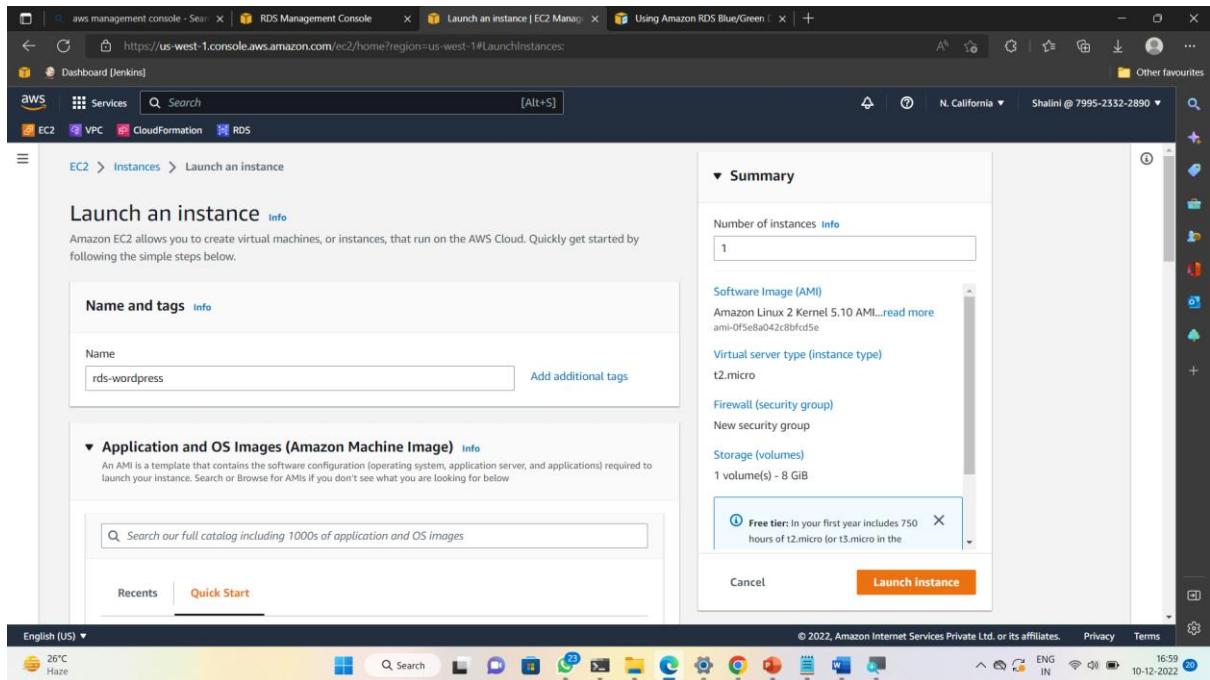


9. You should see a success notice indicating that your database is being created.

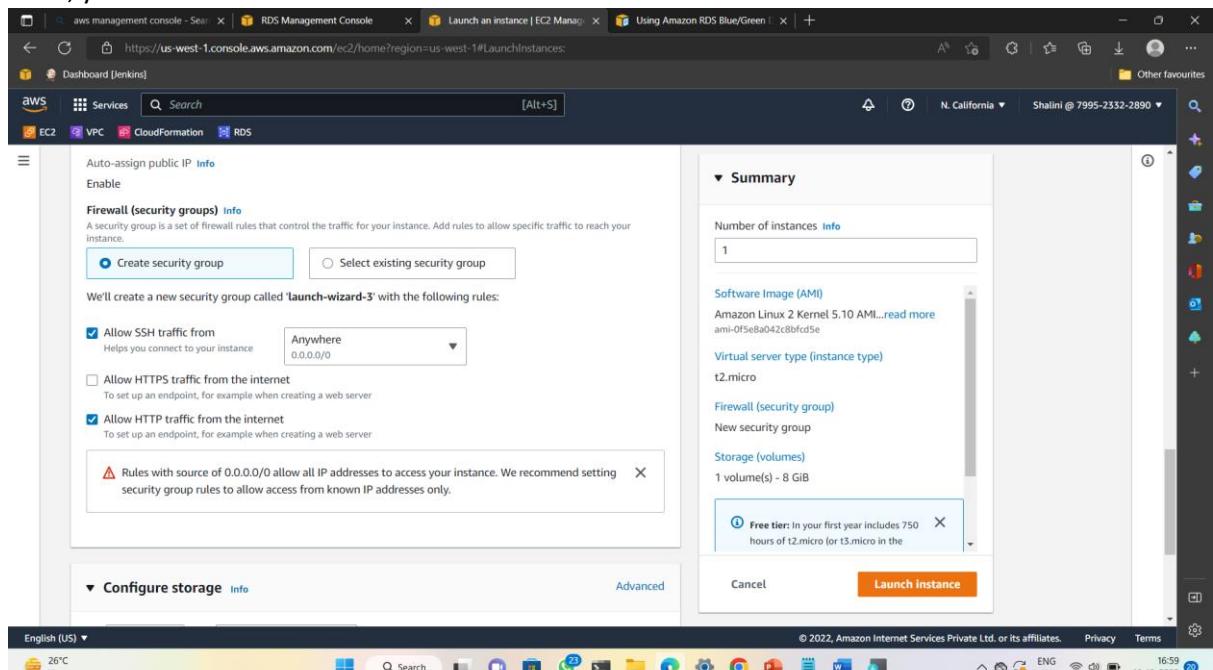


## Lab – 2: Creating an EC2 Instance to run WordPress site.

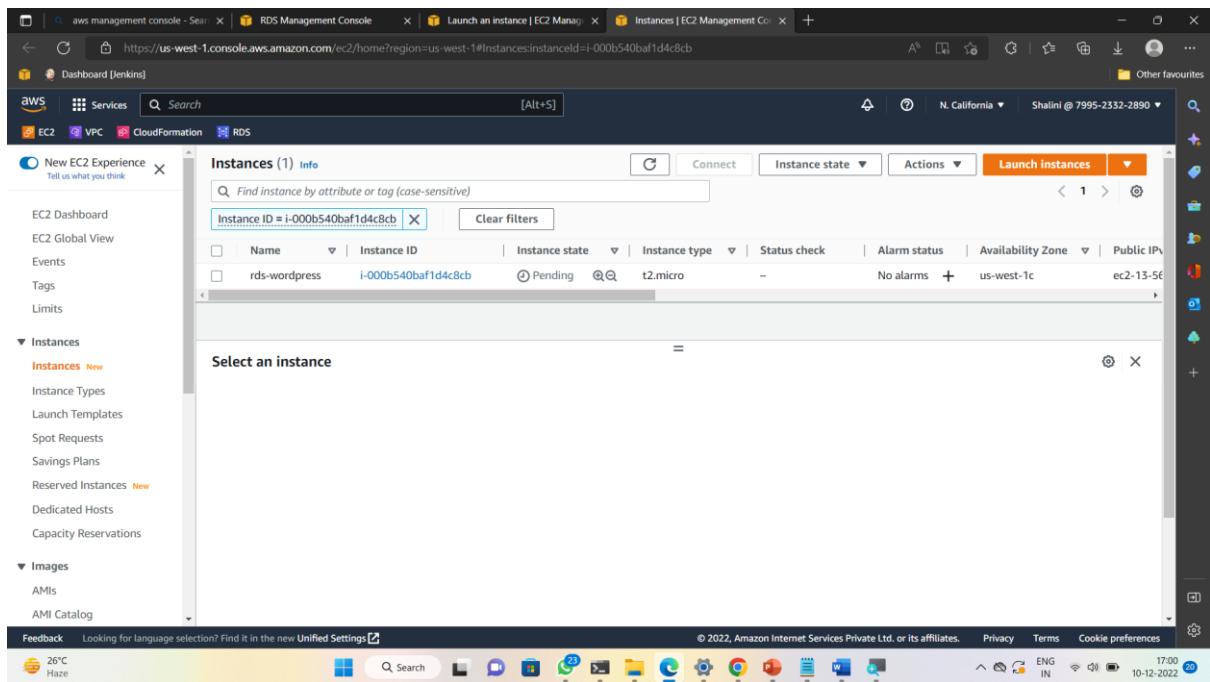
1. To create your EC2 instance, go to Amazon EC2 in the AWS console. Click the blue button that says Launch instance to open the instance creation wizard.



2. There is an SSH rule configured, but it allows SSH access from any IP address. Click under Source to restrict it to your current IP address.
3. Then, you need to add a new rule to allow HTTP traffic. Click Add Rule.

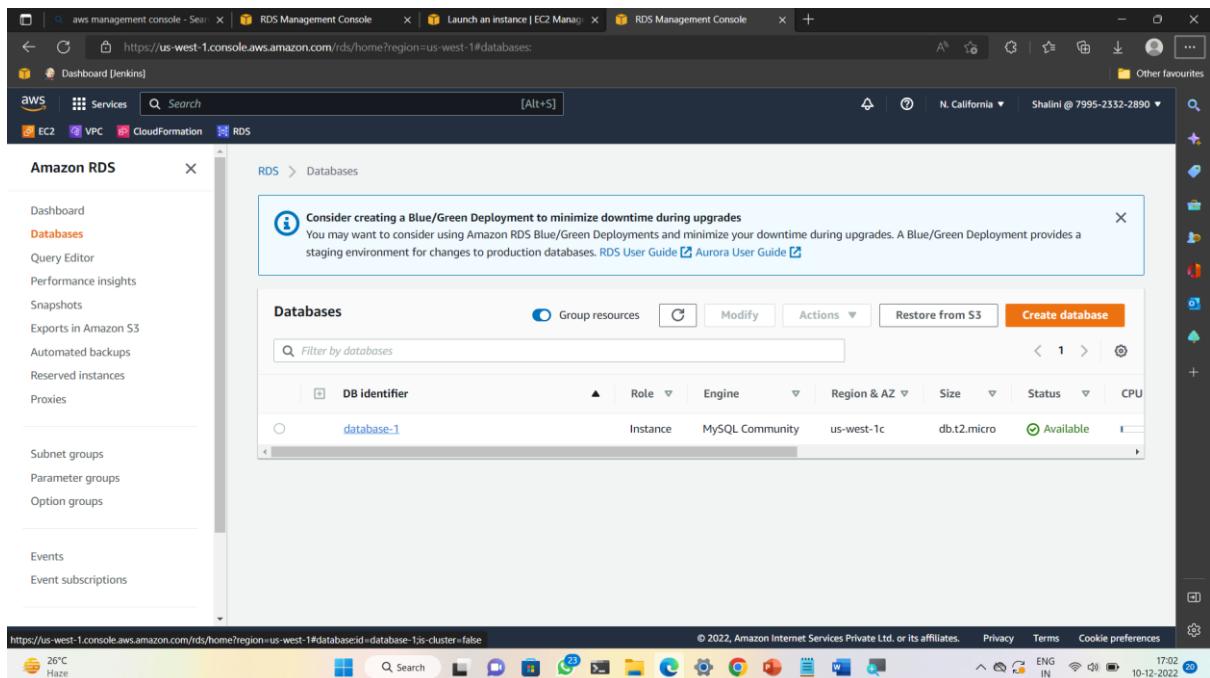


4. You have successfully launched your EC2 instance.



### Lab – 3: Configuring your RDS database.

1. First, you will modify your RDS database to allow network access from your EC2 instance.
2. This time, you want to allow certain traffic from your EC2 instance into your RDS database
3. To configure this, go to the RDS databases in the AWS console. Click on the MySQL database you created in an earlier module in this lab.



4. Scroll to the Connectivity & security tab in the display, and click on the security group listed in VPC security groups.

The screenshot shows the AWS RDS Management Console. On the left, there's a sidebar titled 'Amazon RDS' with various options like Dashboard, Databases, Query Editor, etc. The main area is titled 'Connectivity & security'. It displays information about the endpoint and port, networking details (VPC, subnet group, subnets), and security settings (VPC security groups, certificate authority). The URL in the browser is <https://us-west-1.console.aws.amazon.com/rds/home?region=us-west-1#databaseId=database-1&is-cluster=false>.

5. Click the Inbound tab, then click the Edit button to change the rules for your security group.

The screenshot shows the AWS EC2 Management Console. On the left, there's a sidebar with options like New EC2 Experience, EC2 Dashboard, Instances, and Images. The main area is titled 'Security Groups (1/1) Info'. It shows a table with one row for a security group named 'wordpress'. The 'Actions' menu is open, showing options like 'Edit inbound rules', 'Edit outbound rules', and 'Manage tags'. The URL in the browser is <https://us-west-1.console.aws.amazon.com/ec2/v2/home?region=us-west-1#SecurityGroupssearch=sg-08de57148cf444a4a>.

6. Change the Type property to MySQL/Aurora, which will update the Protocol and Port Range to the proper values.

7. Search the security group that you used for your EC2 instance.
8. hit the Save button to save your changes.

## SSH into your EC2 Instance

1. We are accessing the command line interface with ssh command

```

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\shalini sunny> cd .\downloads
PS C:\Users\shalini sunny\downloads> ssh -i "mykeypair.pem" ec2-user@ec2-13-56-234-193.us-west-1.compute.amazonaws.com
The authenticity of host 'ec2-13-56-234-193.us-west-1.compute.amazonaws.com (13.56.234.193)' can't be established.
ED25519 key fingerprint is SHA256:9/eY34lrlr0cxoteSmkFsizDW7QKuya8/qCLKUiFDK/g8.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-13-56-234-193.us-west-1.compute.amazonaws.com' (ED25519) to the list of known hosts.

[ec2-user@ip-172-31-3-7 ~]$ 

```

2. First, run the following command in your terminal to install a MySQL client to interact with the database.
- **Sudo yum -y install mysql**

3. Next, find the hostname for your RDS database in the AWS console. In the details of your RDS database, the hostname will be shown as the Endpoint in the Connectivity & security section.
4. In your terminal, enter the following command to set an environment variable for your MySQL host. Be sure to replace “” with the hostname of your RDS instance.

➤ **export MYSQL\_HOST=database-1.cbqfqdgifpko.us-west-1.rds.amazonaws.com**

The screenshot shows the AWS RDS Management Console. On the left, there's a sidebar with options like Dashboard, Databases (which is selected), Query Editor, Performance insights, Snapshots, Exports in Amazon S3, Automated backups, Reserved instances, Proxies, Subnet groups, Parameter groups, Option groups, Events, and Event subscriptions. The main area has tabs for Connectivity & security, Monitoring, Logs & events, Configuration, Maintenance & backups, and Tags. Under Connectivity & security, it shows the Endpoint & port (Endpoint: database-1.cbqfqdgifpko.us-west-1.rds.amazonaws.com, Port: 3306), Networking (Availability Zone: us-west-1c, VPC: vpc-06c4b647f3f90a398, Subnet group: default-vpc-06c4b647f3f90a398, Subnets: subnet-0e654ade0b9c67218, subnet-028c7b0166e1a069e, Network type: IPv4), and Security (VPC security groups: default (sg-08de57148cf444a4a) - Active). At the bottom, there are links for Feedback, Looking for language selection? Find it in the new Unified Settings, Privacy, Terms, Cookie preferences, and a weather widget showing 25°C Cloudy.

5. Next, run the following command in your terminal to connect to your MySQL database. Replace “” and “” with the master username and password you configured when creating your RDS database.

➤ **mysql --user= admin --password= 123455678 wordpress**

```

Installing : 1:mariadb-5.5.68-1.amzn2.x86_64
Verifying : 1:mariadb-5.5.68-1.amzn2.x86_64

1/1
1/1

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

Complete!
[ec2-user@ip-172-31-3-7 ~]$ [ec2-user@ip-172-31-3-7 ~]$ [ec2-user@ip-172-31-3-7 ~]$ ssh -i "mykeypair.pem" ec2-user@ec2-13-56-234-193.us-west-1.compute.amazonaws.com
Warning: Identity file mykeypair.pem not accessible: No such file or directory.
The authenticity of host 'ec2-13-56-234-193.us-west-1.compute.amazonaws.com (172.31.3.7)' can't be established.
ECDSA key fingerprint is SHA256:G+iJXJuyQh4FbAoAY3PMjZjZE045Jej5f0H0Cn27hc.
ECDSA key fingerprint is MD5:49:23:c4:61:2c:ed:15:d6:9a:87:51:8c:7d:b9:ca:f4.
Are you sure you want to continue connecting (yes/no)?
Host key verification failed.
[ec2-user@ip-172-31-3-7 ~]$ [ec2-user@ip-172-31-3-7 ~]$ [ec2-user@ip-172-31-3-7 ~]$ [ec2-user@ip-172-31-3-7 ~]$ export MYSQL_HOST=database-1.cbqfqdgifpk0.us-west-1.rds.amazonaws.com
[ec2-user@ip-172-31-3-7 ~]$ mysql --user=admin --password=12345678 wordpress
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MySQL connection id is 29
Server version: 8.0.28 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 [wordpress]> CREATE USER 'wordpress' IDENTIFIED BY 'wordpress-pass';
Query OK, 0 rows affected (0.01 sec)

MySQL [wordpress]> CREATE USER 'wordpress' IDENTIFIED BY 'wordpress-pass';
ERROR 1396 (HY000): Operation CREATE USER failed for 'wordpress'@'%'
MySQL [wordpress]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.00 sec)

MySQL [wordpress]> EXIT
Bye
[ec2-user@ip-172-31-3-7 ~]$
```

- Finally, create a database user for to my WordPress application and give it permission to access the “WordPress” database.

Run the following commands in your terminal:

- **CREATE USER 'wordpress' IDENTIFIED BY 'wordpress-pass';**
- **GRANT ALL PRIVILEGES ON wordpress.\* TO wordpress;**
- **FLUSH PRIVILEGES;**
- **EXIT**

#### **Lab – 4: Configuring WordPress on EC2.**

- Installing the Apache Web Server To run WordPress, i need to run a web server on your EC2 instance.

To install Apache on your EC2 instance,

- run the following command in your terminal:

- **sudo yum install -y httpd**

```

[ec2-user@ip-172-31-3-7 ~]$ export MYSQL_HOST=database-1.cbqfqdg1fpko.us-west-1.rds.amazonaws.com
[ec2-user@ip-172-31-3-7 ~]$ mysql --user=admin --password=12345678 wordpress
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MySQL connection id is 29
Server version: 8.0.28 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 [wordpress]> CREATE USER 'wordpress' IDENTIFIED BY 'wordpress-pass';
Query OK, 0 rows affected (0.01 sec)

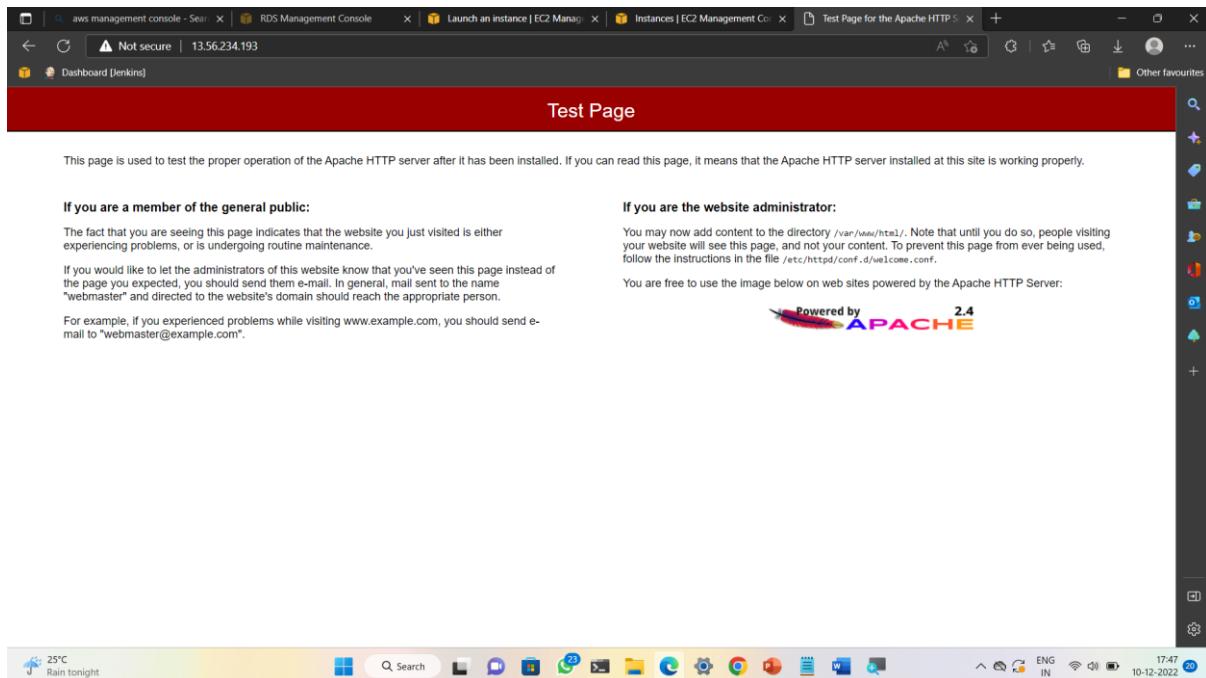
MySQL [wordpress]> CREATE USER 'wordpress' IDENTIFIED BY 'wordpress-pass';
ERROR 1396 (HY000): Operation CREATE USER failed for 'wordpress'@'%'
MySQL [wordpress]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.00 sec)

MySQL [wordpress]> EXIT
Bye
[ec2-user@ip-172-31-3-7 ~]$ client_loop: send disconnect: Connection reset
PS C:\Users\shalini\sunny\downloads> ssh -i "mykeypair.pem" ec2-user@ec2-13-56-234-193.us-west-1.compute.amazonaws.com
Last login: Sat Dec 10 11:56:17 2022 from 103.160.27.7
--| --|- )
_ | ( _ /   Amazon Linux 2 AMI
---\---|---|_|

https://aws.amazon.com/amazon-linux-2/
19 package(s) needed for security, out of 31 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-3-7 ~]$ 
[ec2-user@ip-172-31-3-7 ~]$ 
[ec2-user@ip-172-31-3-7 ~]$ sudo yum -y install httpd
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core
Resolving Dependencies
--> Running transaction check
--> Package httpd.x86_64 0:2.4.54-1.amzn2 will be installed
--> Processing Dependency: httpd-tools = 2.4.54-1.amzn2 for package: httpd-2.4.54-1.amzn2.x86_64
| 3.7 kB  00:00:00

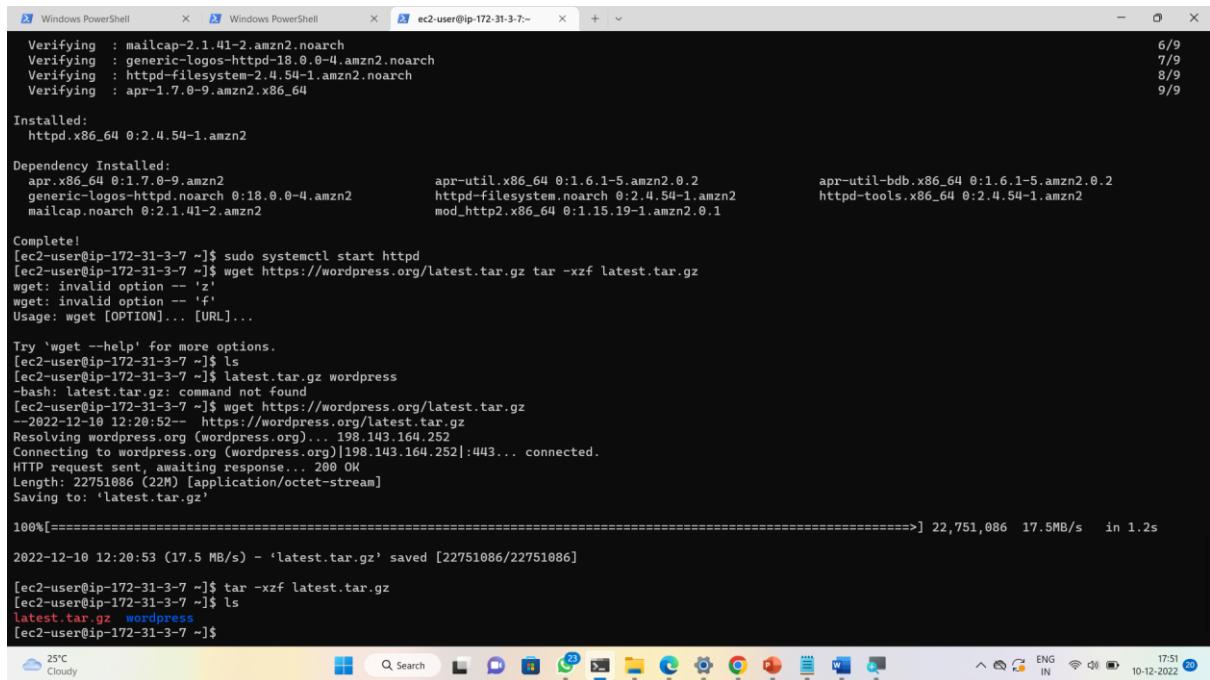
```

- To start the Apache web server, run the following command in your terminal:
  - **sudo service httpd start**
- You can see that your Apache web server is working and that your security groups are configured correctly by visiting the public DNS of your EC2 instance in your browser.



- Now that your Apache web server is working, it's time to download and configure WordPress
- ❖ **Download and Configure WordPress:**
  1. In this step, you will download the WordPress software and set up the configuration.
  2. First, download and uncompressed the software by running the following commands in your terminal:

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



```

Windows PowerShell          Windows PowerShell          ec2-user@ip-172-31-3-7:~ + 
Verifying : mailcap-2.1.41-2.amzn2.noarch          6/9
Verifying : generic-logos-httdp-18.0-4.amzn2.noarch 7/9
Verifying : httpd-filesystem-2.4.54-1.amzn2.noarch 8/9
Verifying : apr-1.7.0-9.amzn2.x86_64                9/9

Installed:
httpd.x86_64 0:2.4.54-1.amzn2

Dependency Installed:
apr.x86_64 0:1.7.0-9.amzn2
generic-logos-httdp.noarch 0:18.0.0-4.amzn2
mailcap.noarch 0:2.1.41-2.amzn2

Complete!
[ec2-user@ip-172-31-3-7 ~]$ sudo systemctl start httpd
[ec2-user@ip-172-31-3-7 ~]$ wget https://wordpress.org/latest.tar.gz tar -xzf latest.tar.gz
wget: invalid option -- 'z'
wget: invalid option -- 'f'
Usage: wget [OPTION]... [URL]...
Try 'wget --help' for more options.
[ec2-user@ip-172-31-3-7 ~]$ ls
[ec2-user@ip-172-31-3-7 ~]$ latest.tar.gz wordpress
-bash: latest.tar.gz: command not found
[ec2-user@ip-172-31-3-7 ~]$ wget https://wordpress.org/latest.tar.gz
--2022-12-10 12:20:52-- https://wordpress.org/latest.tar.gz
Resolving wordpress.org (wordpress.org)... 198.143.164.252
Connecting to wordpress.org (wordpress.org)|198.143.164.252|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 22751086 (22M) [application/octet-stream]
Saving to: 'latest.tar.gz'

100%[=====] 22,751,086 17.5MB/s in 1.2s
2022-12-10 12:20:53 (17.5 MB/s) - 'latest.tar.gz' saved [22751086/22751086]

[ec2-user@ip-172-31-3-7 ~]$ tar -xzf latest.tar.gz
[ec2-user@ip-172-31-3-7 ~]$ ls
latest.tar.gz wordpress
[ec2-user@ip-172-31-3-7 ~]$ 

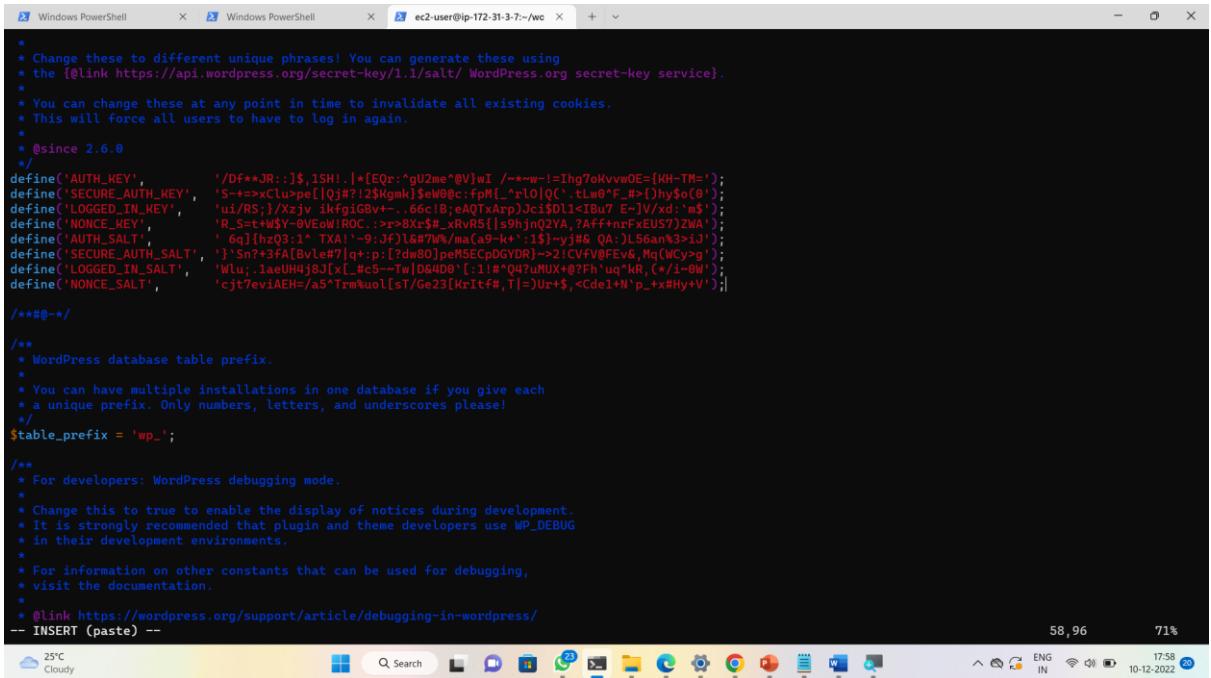
```

The screenshot shows a Windows PowerShell window with three tabs open. The active tab displays the command-line session for downloading and extracting the WordPress latest tarball. The taskbar at the bottom shows various application icons, and the system tray indicates it's 17:51 on 10-12-2022.

3. If you run “ls” to view the contents of your directory, you will see a tar file and a directory called wordpress with the uncompressed contents.
  - ls
  - latest.tar.gz wordpress
4. Change into the wordpress directory and create a copy of the default config file using the following commands:
  - cd wordpress
  - cp wp-config-sample.php wp-config.php
5. Then, open the wp-config.php file using the vi editor by running the following command.
6. You need to edit two areas of configuration.
7. First, edit the database configuration by changing the following lines:

```
// ** MySQL settings - You can get this info from your web host ** // /**
 * The name of the
database for WordPress */
define( 'DB_NAME', 'database_name_here' );
/** MySQL database username */
define( 'DB_USER', 'username_here' );
/** MySQL database password */
define( 'DB_PASSWORD', 'password_here' );
/** MySQL hostname */
define( 'DB_HOST', 'localhost' );
```

- The second configuration section you need to configure is the Authentication Unique Keys and Salts. It looks as follows in the configuration



```

/*
 * 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', '/Df**JR::]$._lSH!.|*[EO:~^qU2me@VwI /~~w-!=Ihg7oKvvOE={WH-TM='); define('SECURE_AUTH_KEY', 'S-+>xClu>|[Oj#?2$kgmk$lew@0c:fpMl_>r10|Q`_tLw@F_#>()hySo(0'); define('LOGGED_IN_KEY', 'ui@RS:}/Xzjv ikFpiGBv--_66c1B:eAQTxArp)JcisDL1:tBu7 E-Jv/xd:_s$'); define('NONCE_KEY', 'R_Sst+w$Y-6VEoW!ROC_>r>8Xr$_.xRvRS5{[s9hjnQ2YA_7Aff+nxFxeUS7)ZWA'); define('AUTH_SALT', '_6q]{hzQ2:|^_TXA!`-9:jF]lg#7W%ma(a9-k+`1$>yjR6 QA:JL56an%3>i3'); define('SECURE_AUTH_SALT', ']`Sn?+3FA[BvLe#7|q+:p:[?dw80]peMSECp0GyDR)-->2!CVFV@FEvG_Mq(WCyg'); define('LOGGED_IN_SALT', 'Wlu_iaeUHujbjJ[x_#c5->lw|Dw4D0[::!#^Q4?uMUX+@?Fn[uq"KR_(</i-0W'); define('NONCE_SALT', 'cj7eviAEH=a5'Trm@uol[ST/Ge23(krit#,T|=Ur+s,<de1+N'p_++#Hyv');
/**#@*/
/**
 * WordPress database table prefix.
 *
 * You can have multiple installations in one database if you give each
 * a unique prefix. Only numbers, letters, and underscores please!
 */
$table_prefix = 'wp_';

/**
 * For developers: WordPress debugging mode.
 *
 * Change this to true to enable the display of notices during development.
 * It is strongly recommended that plugin and theme developers use WP_DEBUG
 * in their development environments.
 *
 * For information on other constants that can be used for debugging,
 * visit the documentation.
 *
 * [link https://wordpress.org/support/article/debugging-in-wordpress/
-- INSERT (paste) --

```

- You can save and exit from vi editor by entering Esc then type wq and hit Enter.

#### ❖ Deploying WordPress Site

1. In this step, you will make your Apache web server handle requests for WordPress.
2. First, install the application dependencies you need for WordPress. In your terminal, run the following command.
3. sudo amazon-linux-extras install -y lamp-mariadb10.2-php7.2 php7.2
4. Second, change to the proper directory by running the following command:

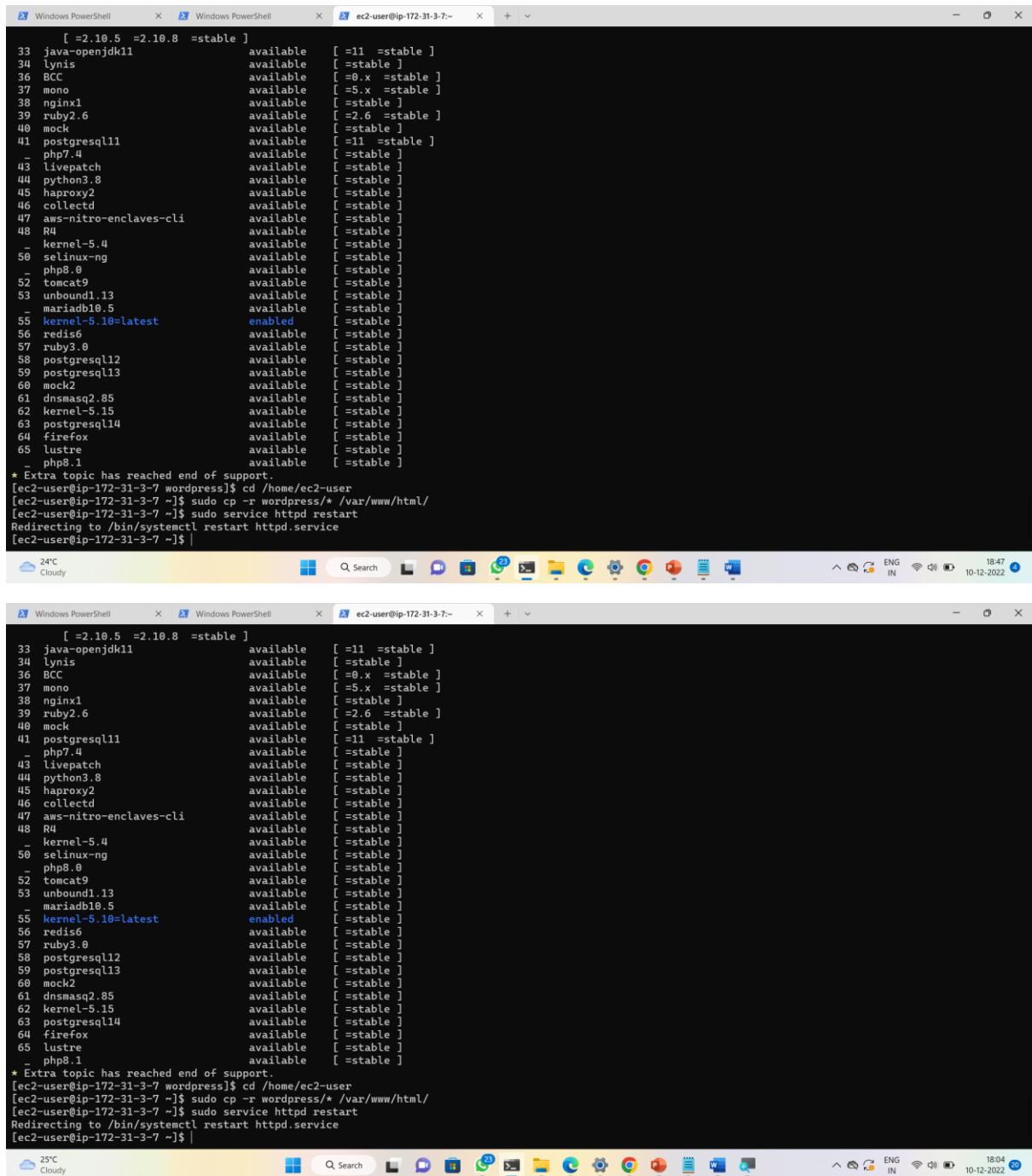
➤ **cd /home/ec2-user**

Then, copy your WordPress application files into the /var/www/html directory used by Apache.

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

Finally, restart the Apache web server to pick up the changes.

- **sudo service httpd restart**



```
[ =2.10.5  =2.10.8  =stable ]
33 java-openjdk11      available  [ =11  =stable ]
34 lynis               available  [ =stable ]
36 BCC                 available  [ =0.x  =stable ]
37 mono                available  [ =5.x  =stable ]
38 nginx1x             available  [ =stable ]
39 ruby2.6             available  [ =2.6  =stable ]
40 mock                available  [ =stable ]
41 postgresql11        available  [ =11  =stable ]
42 php7.4              available  [ =stable ]
43 livepatch           available  [ =stable ]
44 python3.8            available  [ =stable ]
45 haproxy2             available  [ =stable ]
46 collectd             available  [ =stable ]
47 aws-nitro-enclaves-cl
48 R4                  available  [ =stable ]
49 kernel-5.4            available  [ =stable ]
50 selinux-ng           available  [ =stable ]
51 php8.0               available  [ =stable ]
52 tomcat9              available  [ =stable ]
53 unbound1.13          available  [ =stable ]
54 mariadb10.5          available  [ =stable ]
55 kernel-5.10=latest   enabled    [ =stable ]
56 redis6               available  [ =stable ]
57 ruby3.0              available  [ =stable ]
58 postgresql12          available  [ =stable ]
59 postgresql13          available  [ =stable ]
60 mock2                available  [ =stable ]
61 dnsmasq2.85          available  [ =stable ]
62 kernel-5.15            available  [ =stable ]
63 postgresql14          available  [ =stable ]
64 firefox              available  [ =stable ]
65 lustre               available  [ =stable ]
66 php8.1               available  [ =stable ]
* Extra topic has reached end of support.
[ec2-user@ip-172-31-3-7 ~]$ cd /home/ec2-user
[ec2-user@ip-172-31-3-7 ~]$ sudo cp -r wordpress/* /var/www/html/
[ec2-user@ip-172-31-3-7 ~]$ sudo service httpd restart
Redirecting to /bin/systemctl restart httpd.service
[ec2-user@ip-172-31-3-7 ~]$ |
```

24°C Cloudy 18:47 ENG IN 10-12-2022

```
[ =2.10.5  =2.10.8  =stable ]
33 java-openjdk11      available  [ =11  =stable ]
34 lynis               available  [ =stable ]
36 BCC                 available  [ =0.x  =stable ]
37 mono                available  [ =5.x  =stable ]
38 nginx1x             available  [ =stable ]
39 ruby2.6             available  [ =2.6  =stable ]
40 mock                available  [ =stable ]
41 postgresql11        available  [ =11  =stable ]
42 php7.4              available  [ =stable ]
43 livepatch           available  [ =stable ]
44 python3.8            available  [ =stable ]
45 haproxy2             available  [ =stable ]
46 collectd             available  [ =stable ]
47 aws-nitro-enclaves-cl
48 R4                  available  [ =stable ]
49 kernel-5.4            available  [ =stable ]
50 selinux-ng           available  [ =stable ]
51 php8.0               available  [ =stable ]
52 tomcat9              available  [ =stable ]
53 unbound1.13          available  [ =stable ]
54 mariadb10.5          available  [ =stable ]
55 kernel-5.10=latest   enabled    [ =stable ]
56 redis6               available  [ =stable ]
57 ruby3.0              available  [ =stable ]
58 postgresql12          available  [ =stable ]
59 postgresql13          available  [ =stable ]
60 mock2                available  [ =stable ]
61 dnsmasq2.85          available  [ =stable ]
62 kernel-5.15            available  [ =stable ]
63 postgresql14          available  [ =stable ]
64 firefox              available  [ =stable ]
65 lustre               available  [ =stable ]
66 php8.1               available  [ =stable ]
* Extra topic has reached end of support.
[ec2-user@ip-172-31-3-7 ~]$ cd /home/ec2-user
[ec2-user@ip-172-31-3-7 ~]$ sudo cp -r wordpress/* /var/www/html/
[ec2-user@ip-172-31-3-7 ~]$ sudo service httpd restart
Redirecting to /bin/systemctl restart httpd.service
[ec2-user@ip-172-31-3-7 ~]$ |
```

25°C Cloudy 18:04 ENG IN 10-12-2022

You should see the WordPress welcome page and the five-minute installation process

