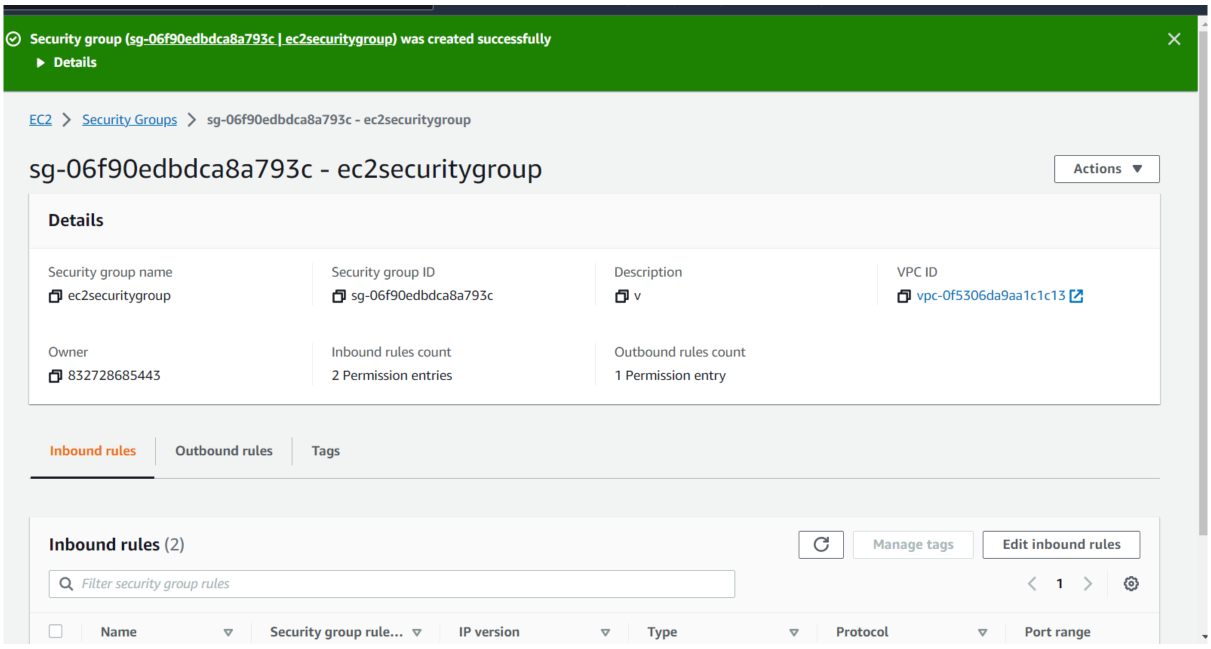
Report submitted by : **Muneeb Ahmed Bali**

TASK 1:

For monolith : 1 EC2 instance, deploy wordpress and MYSQL on the same instances

Creation AWS VPC Security Group

1. **Ec2Securtitygroup:**



Security group (sg-06f90edbdca8a793c | ec2securitygroup) was created successfully

Details

EC2 > Security Groups > sg-06f90edbdca8a793c - ec2securitygroup

sg-06f90edbdca8a793c - ec2securitygroup

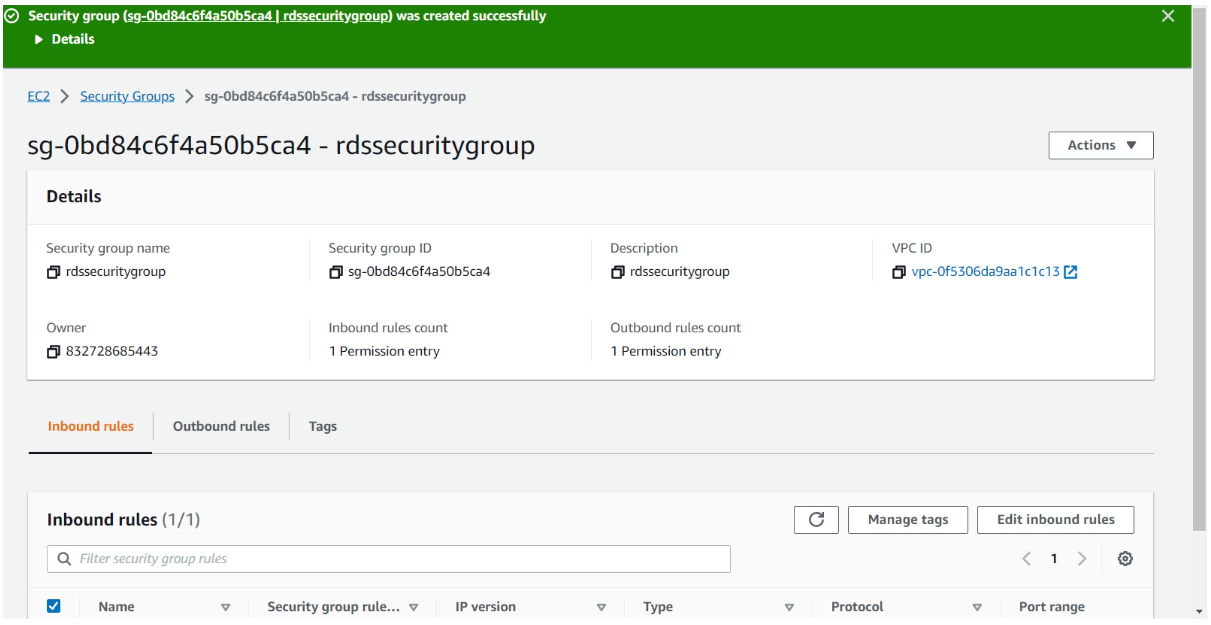
Details

Security group name	Security group ID	Description	VPC ID
ec2securitygroup	sg-06f90edbdca8a793c	v	vpc-0f5306da9aa1c1c13
Owner	**Inbound rules count**	**Outbound rules count**	
832728685443	2 Permission entries	1 Permission entry	

Inbound rules (2)

Name	Security group rule...	IP version	Type	Protocol	Port range

2. **RDS security group**



Security group (sg-0bd84c6f4a50b5ca4 | rdssecuritygroup) was created successfully
Details

EC2 > Security Groups > sg-0bd84c6f4a50b5ca4 - rdssecuritygroup

sg-0bd84c6f4a50b5ca4 - rdssecuritygroup

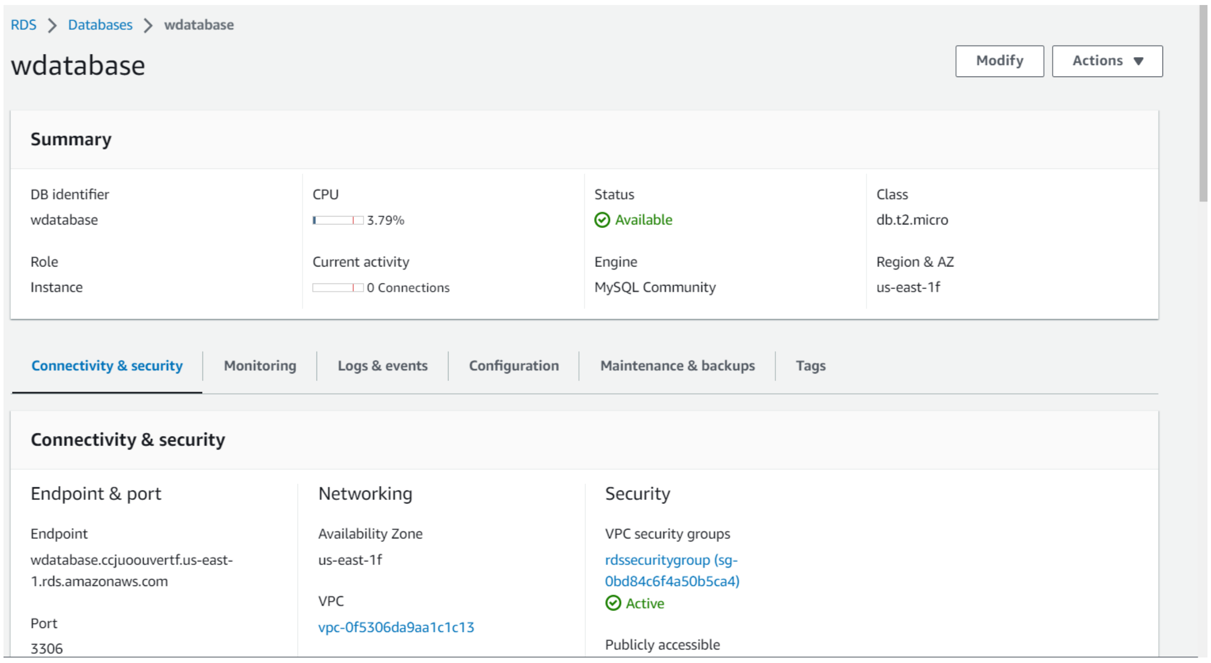
Details

Security group name	Security group ID	Description	VPC ID
rdssecuritygroup	sg-0bd84c6f4a50b5ca4	rdssecuritygroup	vpc-0f5306da9aa1c1c13
Owner	Inbound rules count	Outbound rules count	
832728685443	1 Permission entry	1 Permission entry	

Inbound rules (1/1)

Name	Security group rule...	IP version	Type	Protocol	Port range

Creation of AWS RDS Database



RDS > Databases > wdatabase

wdatabase

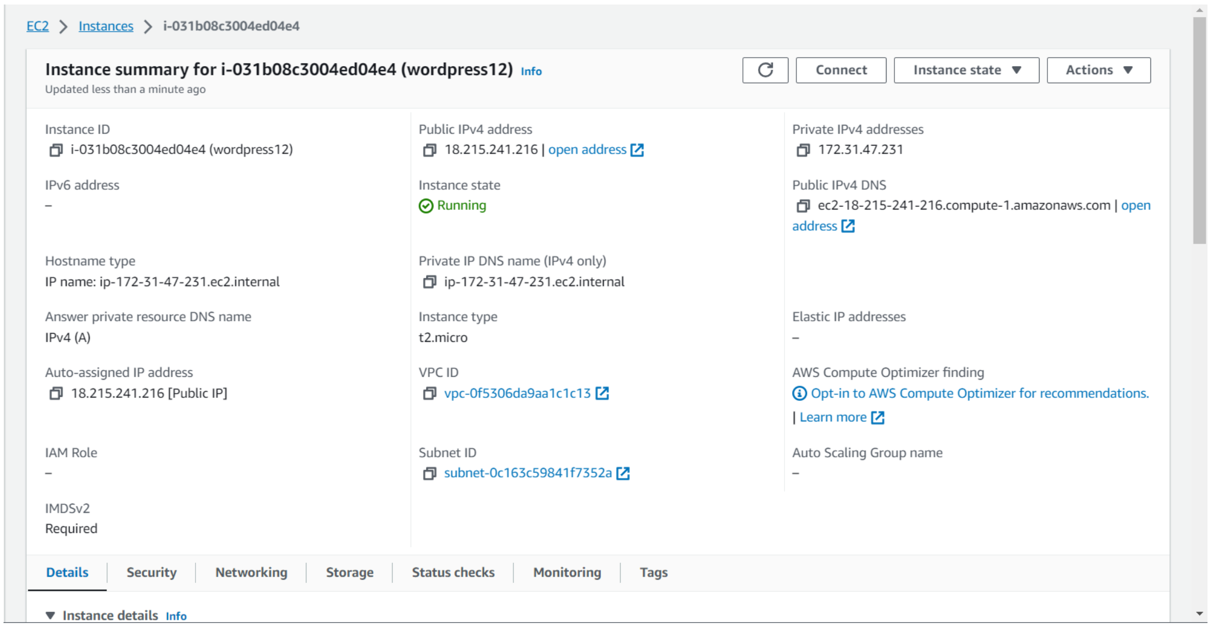
Summary

DB identifier	CPU	Status	Class
wdatabase	3.79%	Available	db.t2.micro
Role	Current activity	Engine	Region & AZ
Instance	0 Connections	MySQL Community	us-east-1f

Connectivity & security

Endpoint & port	Networking	Security
Endpoint	Availability Zone	VPC security groups
wdatabase.ccjuoouvertf.us-east-1.rds.amazonaws.com	us-east-1f	rdssecuritygroup (sg-0bd84c6f4a50b5ca4) Active
Port	VPC	Publicly accessible
3306	vpc-0f5306da9aa1c1c13	

Creation of AWS EC2 Instance –



EC2 > Instances > i-031b08c3004ed04e4

Instance summary for i-031b08c3004ed04e4 (wordpress12) Info

Updated less than a minute ago

Instance ID
i-031b08c3004ed04e4 (wordpress12)

Public IPv4 address
18.215.241.216 | open address

Private IPv4 addresses
172.31.47.231

IPv6 address
–

Instance state
Running

Public IPv4 DNS
ec2-18-215-241-216.compute-1.amazonaws.com | open address

Hostname type
IP name: ip-172-31-47-231.ec2.internal

Private IP DNS name (IPv4 only)
ip-172-31-47-231.ec2.internal

Answer private resource DNS name
IPv4 (A)

Instance type
t2.micro

Elastic IP addresses
–

Auto-assigned IP address
18.215.241.216 [Public IP]

VPC ID
vpc-0f5306da9aa1c1c13

AWS Compute Optimizer finding
Opt-in to AWS Compute Optimizer for recommendations. | Learn more

IAM Role
–

Subnet ID
subnet-0c163c59841f7352a

Auto Scaling Group name
–

IMDSv2
Required

Installing php, mysql, wordpress –

Configure wordpress with AWS RDS Database

Step 1: Prepare the LAMP server

Prerequisites

- This tutorial assumes that you have already launched a new instance using Amazon Linux 2023, with a public DNS name that is reachable from the internet. For more information, see Step 1: Launch an instance. You must also have configured your security group to allow SSH (port 22), HTTP (port 80), and HTTPS (port 443) connections. For more information about these prerequisites, see Authorize inbound traffic for your Linux instances.
- The following procedure installs the latest PHP version available on Amazon Linux 2023, currently 8.1. If you plan to use PHP applications other than those described in this tutorial, you should check their compatibility with 8.1.

To prepare the LAMP server

1. Connect to your instance.

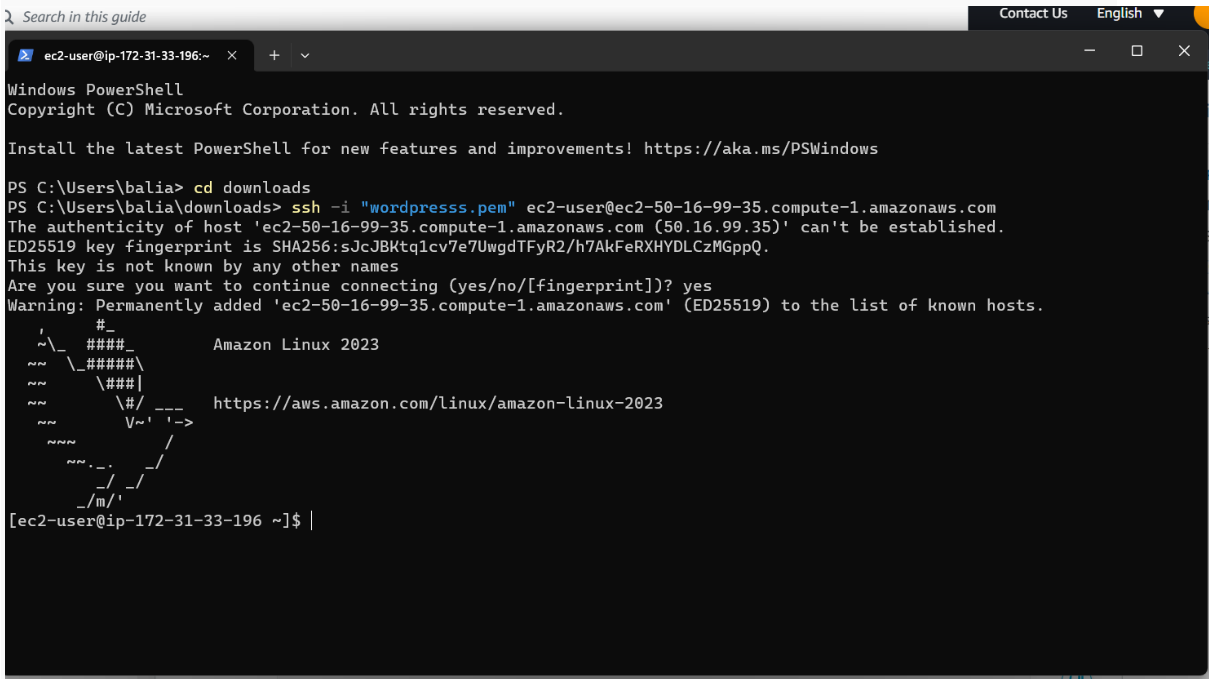
2. To ensure that all of your software packages are up to date, perform a quick software update on your instance. This process might take a few minutes, but it is important to make sure that you have the latest security updates and bug fixes.

   The -y option installs the updates without asking for confirmation. If you would like to examine the updates before installing, you can omit this option.

   ```
   [ec2-user ~]$ sudo dnf update -y
   ```

3. Install the latest versions of Apache web server and PHP packages for Amazon Linux 2023.

   ```
   [ec2-user ~]$ sudo dnf install -y httpd wget php-fpm php-
   mysqli php-json php php-devel
   ```



```
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\balia> cd downloads
PS C:\Users\balia\downloads> ssh -i "wordpresss.pem" ec2-user@ec2-50-16-99-35.compute-1.amazonaws.com
The authenticity of host 'ec2-50-16-99-35.compute-1.amazonaws.com (50.16.99.35)' can't be established.
ED25519 key fingerprint is SHA256:sJcJBKtq1cv7e7UwgdTFyR2/h7AkFeRXHYDLCzMGppQ.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-50-16-99-35.compute-1.amazonaws.com' (ED25519) to the list of known hosts.
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
[ec2-user@ip-172-31-33-196 ~]$ |
```

4. Install the MariaDB software packages. Use the **dnf install** command to install multiple software packages and all related dependencies at the same time.

   ```
   [ec2-user ~]$ sudo dnf install mariadb105-server
   ```

   You can view the current versions of these packages using the following command:

5. Start the Apache web server.

   ```
   [ec2-user ~]$ sudo systemctl start httpd
   ```

6. Use the **systemctl** command to configure the Apache web server to start at each system boot.

```
[ec2-user ~]$ sudo systemctl enable httpd
```

   You can verify that **httpd** is on by running the following command:

```
[ec2-user ~]$ sudo systemctl is-enabled httpd
```

7. Add a security rule to allow inbound HTTP (port 80) connections to your instance if you have not already done so. By default, a **launch-wizard-*N*** security group was created for your instance during launch. If you did not add additional security group rules, this group contains only a single rule to allow SSH connections.
   a. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
   b. In the left navigator, choose **Instances**, and select your instance.
   c. On the **Security** tab, view the inbound rules. You should see the following rule:

```
d. Port range    Protocol      Source

   22            tcp           0.0.0.0/0
```

   Warning

   Using `0.0.0.0/0` allows all IPv4 addresses to access your instance using SSH. This is acceptable for a short time in a test environment, but it's unsafe for production environments. In production, you authorize only a specific IP address or range of addresses to access your instance.
   e. If there is no inbound rule to allow HTTP (port 80) connections, you must the add rule now. Choose the link for the security group. Using the procedures in Add rules to a security group, add a new inbound security rule with the following values:
      - **Type**: HTTP
      - **Protocol**: TCP
      - **Port Range**: 80
      - **Source**: Custom
8. Test your web server. In a web browser, type the public DNS address (or the public IP address) of your instance. If there is no content in `/var/www/html`, you should see the Apache test page, which will display the message "**It works!**".

You can get the public DNS for your instance using the Amazon EC2 console (check the **Public IPv4 DNS** column; if this column is hidden, choose **Preferences** (the gear-shaped icon) and toggle on **Public IPv4 DNS**).

Verify that the security group for the instance contains a rule to allow HTTP traffic on port 80. For more information, see Add rules to a security group.

Important

If you are not using Amazon Linux, you might also need to configure the firewall on your instance to allow these connections. For more information about how to configure the firewall, see the documentation for your specific distribution.

Apache **httpd** serves files that are kept in a directory called the Apache document root. The Amazon Linux Apache document root is `/var/www/html`, which by default is owned by root.

To allow the `ec2-user` account to manipulate files in this directory, you must modify the ownership and permissions of the directory. There are many ways to accomplish this task. In this tutorial, you add `ec2-user` to the `apache` group to give the `apache` group ownership of the `/var/www` directory and assign write permissions to the group.

To set file permissions

1. Add your user (in this case, `ec2-user`) to the `apache` group.

   ```
   [ec2-user ~]$ sudo usermod -a -G apache ec2-user
   ```

2. Log out and then log back in again to pick up the new group, and then verify your membership.
   a. Log out (use the **exit** command or close the terminal window):

      ```
      [ec2-user ~]$ exit
      ```

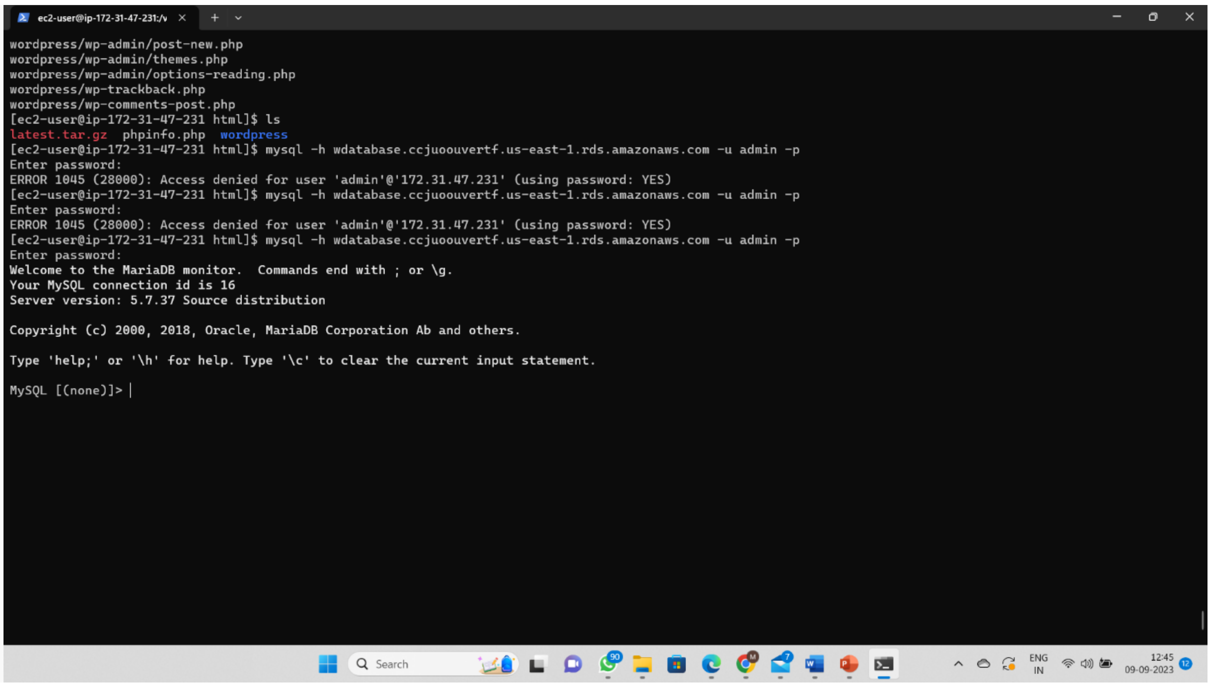
   b. To verify your membership in the `apache` group, reconnect to your instance, and then run the following command:

      ```
   c. [ec2-user ~]$ groups
      ```

      ```
      ec2-user adm wheel apache systemd-journal
      ```

3. Change the group ownership of `/var/www` and its contents to the `apache` group.

```
[ec2-user ~]$ sudo chown -R ec2-user:apache /var/www
```



```
wordpress/wp-admin/post-new.php
wordpress/wp-admin/themes.php
wordpress/wp-admin/options-reading.php
wordpress/wp-trackback.php
wordpress/wp-comments-post.php
[ec2-user@ip-172-31-47-231 html]$ ls
latest.tar.gz  phpinfo.php  wordpress
[ec2-user@ip-172-31-47-231 html]$ mysql -h wdatabase.ccjuoouvertf.us-east-1.rds.amazonaws.com -u admin -p
Enter password:
ERROR 1045 (28000): Access denied for user 'admin'@'172.31.47.231' (using password: YES)
[ec2-user@ip-172-31-47-231 html]$ mysql -h wdatabase.ccjuoouvertf.us-east-1.rds.amazonaws.com -u admin -p
Enter password:
ERROR 1045 (28000): Access denied for user 'admin'@'172.31.47.231' (using password: YES)
[ec2-user@ip-172-31-47-231 html]$ mysql -h wdatabase.ccjuoouvertf.us-east-1.rds.amazonaws.com -u admin -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 16
Server version: 5.7.37 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)]> |
```

4. To add group write permissions and to set the group ID on future subdirectories, change the directory permissions of `/var/www` and its subdirectories.

```
[ec2-user ~]$ sudo chmod 2775 /var/www && find /var/www -type d -exec sudo chmod 2775 {} \;
```

5. To add group write permissions, recursively change the file permissions of `/var/www` and its subdirectories:

```
[ec2-user ~]$ find /var/www -type f -exec sudo chmod 0664 {} \;
```

Now, `ec2-user` (and any future members of the `apache` group) can add, delete, and edit files in the Apache document root, enabling you to add content, such as a static website or a PHP application.

To secure your web server (Optional)

A web server running the HTTP protocol provides no transport security for the data that it sends or receives. When you connect to an HTTP server using a web browser, the URLs that you visit, the content of webpages that you receive, and the contents (including passwords) of any HTML forms that you submit are all visible to eavesdroppers anywhere along the network pathway. The best practice for securing your web server is to install support for HTTPS (HTTP Secure), which protects your data with SSL/TLS encryption.

For information about enabling HTTPS on your server, see Configure SSL/TLS on Amazon Linux 2.

Step 2: Test your LAMP server

If your server is installed and running, and your file permissions are set correctly, your `ec2-user` account should be able to create a PHP file in the `/var/www/html` directory that is available from the internet.

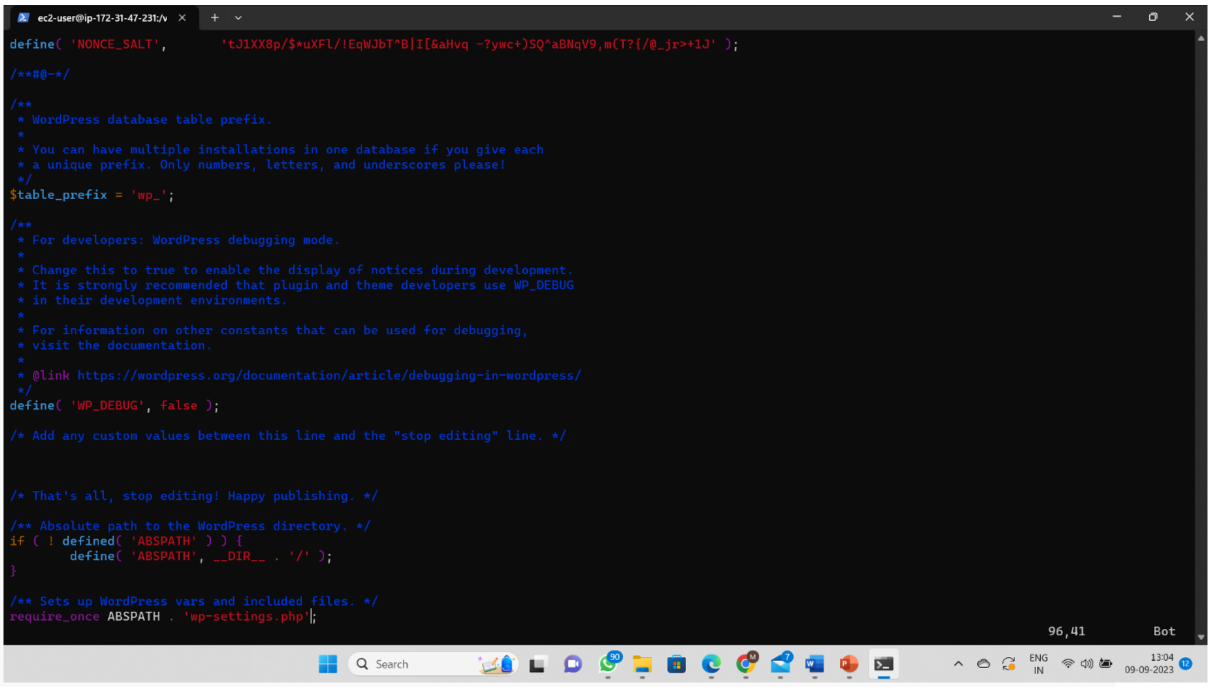
To test your LAMP server

1. Create a PHP file in the Apache document root.

   ```
   [ec2-user ~]$ echo "<?php phpinfo(); ?>" >
   /var/www/html/phpinfo.php
   ```

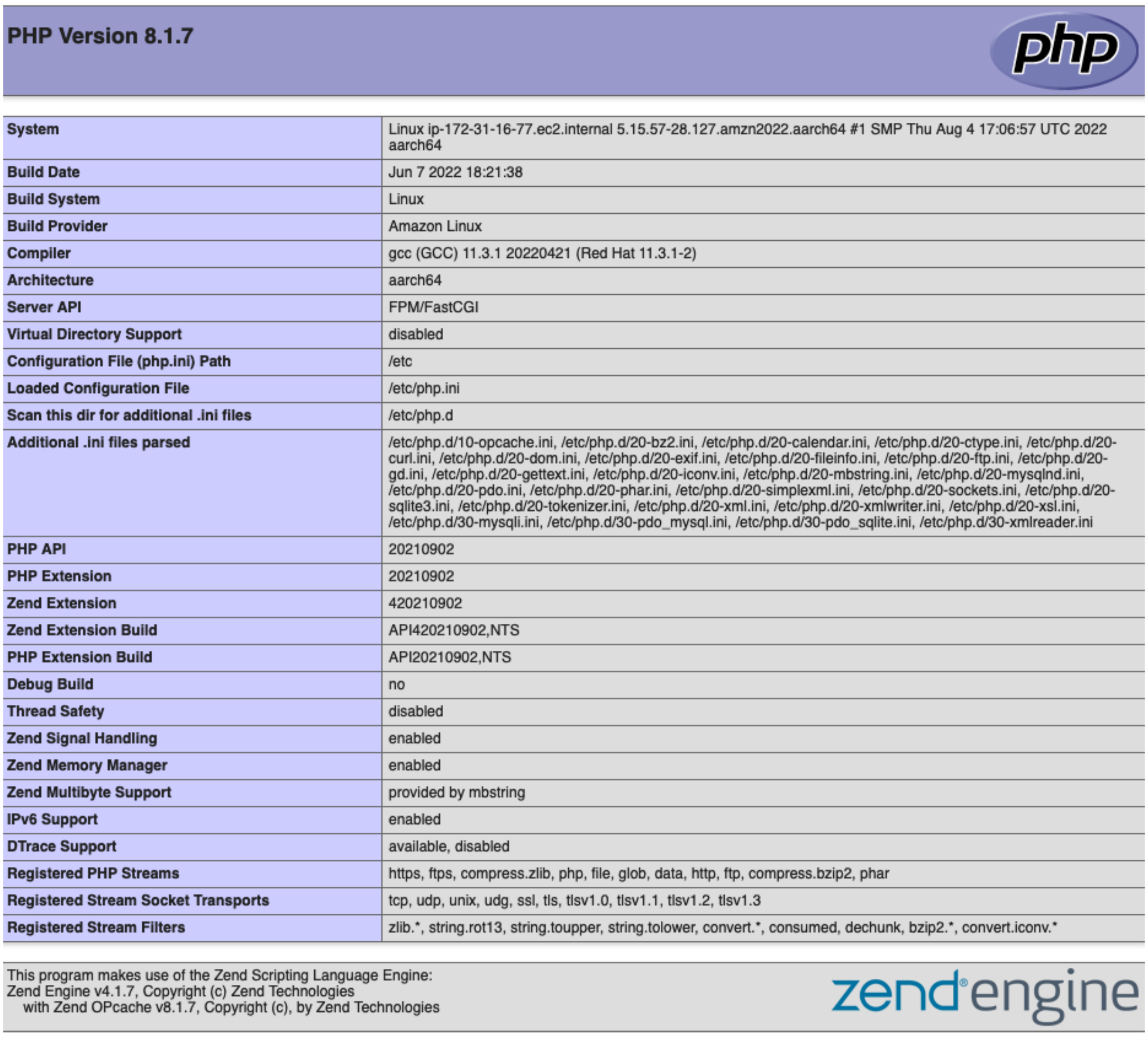
   If you get a "Permission denied" error when trying to run this command, try logging out and logging back in again to pick up the proper group permissions that you configured in To set file permissions.
2. In a web browser, type the URL of the file that you just created. This URL is the public DNS address of your instance followed by a forward slash and the file name. For example:

   ```
   http://my.public.dns.amazonaws.com/phpinfo.php
   ```



```
define( 'NONCE_SALT',       'tJ1XX8p/$*uXFl/!EqWJbT^B|I[&aHvq -?ywc+)SQ^aBNqV9,m(T?{/@_jr>+1J' );

/**#@-*/

/**
 * 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/documentation/article/debugging-in-wordpress/
 */
define( 'WP_DEBUG', false );

/* Add any custom values between this line and the "stop editing" line. */



/* That's all, stop editing! Happy publishing. */

/** Absolute path to the WordPress directory. */
if ( ! defined( 'ABSPATH' ) ) {
        define( 'ABSPATH', __DIR__ . '/' );
}

/** Sets up WordPress vars and included files. */
require_once ABSPATH . 'wp-settings.php';
                                                                    96,41         Bot
```

You should see the PHP information page:



PHP Version 8.1.7

System	Linux ip-172-31-16-77.ec2.internal 5.15.57-28.127.amzn2022.aarch64 #1 SMP Thu Aug 4 17:06:57 UTC 2022 aarch64
Build Date	Jun 7 2022 18:21:38
Build System	Linux
Build Provider	Amazon Linux
Compiler	gcc (GCC) 11.3.1 20220421 (Red Hat 11.3.1-2)
Architecture	aarch64
Server API	FPM/FastCGI
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc
Loaded Configuration File	/etc/php.ini
Scan this dir for additional .ini files	/etc/php.d
Additional .ini files parsed	/etc/php.d/10-opcache.ini, /etc/php.d/20-bz2.ini, /etc/php.d/20-calendar.ini, /etc/php.d/20-ctype.ini, /etc/php.d/20-curl.ini, /etc/php.d/20-dom.ini, /etc/php.d/20-exif.ini, /etc/php.d/20-fileinfo.ini, /etc/php.d/20-ftp.ini, /etc/php.d/20-gd.ini, /etc/php.d/20-gettext.ini, /etc/php.d/20-iconv.ini, /etc/php.d/20-mbstring.ini, /etc/php.d/20-mysqlnd.ini, /etc/php.d/20-pdo.ini, /etc/php.d/20-phar.ini, /etc/php.d/20-simplexml.ini, /etc/php.d/20-sockets.ini, /etc/php.d/20-sqlite3.ini, /etc/php.d/20-tokenizer.ini, /etc/php.d/20-xml.ini, /etc/php.d/20-xmlwriter.ini, /etc/php.d/20-xsl.ini, /etc/php.d/30-mysqli.ini, /etc/php.d/30-pdo_mysql.ini, /etc/php.d/30-pdo_sqlite.ini, /etc/php.d/30-xmlreader.ini
PHP API	20210902
PHP Extension	20210902
Zend Extension	420210902
Zend Extension Build	API420210902,NTS
PHP Extension Build	API20210902,NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	enabled
Zend Memory Manager	enabled
Zend Multibyte Support	provided by mbstring
IPv6 Support	enabled
DTrace Support	available, disabled
Registered PHP Streams	https, ftps, compress.zlib, php, file, glob, data, http, ftp, compress.bzip2, phar
Registered Stream Socket Transports	tcp, udp, unix, udg, ssl, tls, tlsv1.0, tlsv1.1, tlsv1.2, tlsv1.3
Registered Stream Filters	zlib.*, string.rot13, string.toupper, string.tolower, convert.*, consumed, dechunk, bzip2.*, convert.iconv.*

This program makes use of the Zend Scripting Language Engine:
Zend Engine v4.1.7, Copyright (c) Zend Technologies
    with Zend OPcache v8.1.7, Copyright (c), by Zend Technologies

If you do not see this page, verify that the `/var/www/html/phpinfo.php` file was created properly in the previous step. You can also verify that all of the required packages were installed with the following command.

```
[ec2-user ~]$ sudo dnf list installed httpd mariadb-server php-mysqlnd
```

If any of the required packages are not listed in your output, install them with the **sudo yum install *package*** command.

3. Delete the `phpinfo.php` file. Although this can be useful information, it should not be broadcast to the internet for security reasons.

```
[ec2-user ~]$ rm /var/www/html/phpinfo.php
```

You should now have a fully functional LAMP web server. If you add content to the Apache document root at `/var/www/html`, you should be able to view that content at the public DNS address for your instance.

Step 3: Secure the database server

The default installation of the MariaDB server has several features that are great for testing and development, but they should be disabled or removed for production servers. The **mysql_secure_installation** command walks you through the process of setting a root password and removing the insecure features from your installation. Even if you are not planning on using the MariaDB server, we recommend performing this procedure.

To secure the MariaDB server

1. Start the MariaDB server.

   ```
   [ec2-user ~]$ sudo systemctl start mariadb
   ```

2. Run **mysql_secure_installation**.

   ```
   [ec2-user ~]$ sudo mysql_secure_installation
   ```

   a. When prompted, type a password for the root account.
      i. Type the current root password. By default, the root account does not have a password set. Press Enter.
      ii. Type `Y` to set a password, and type a secure password twice. For more information about creating a secure password, see https://identitysafe.norton.com/password-generator/. Make sure to store this password in a safe place.

          Setting a root password for MariaDB is only the most basic measure for securing your database. When you build or install a database-driven application, you typically create a database service user for that application and avoid using the root account for anything but database administration.
   b. Type `Y` to remove the anonymous user accounts.
   c. Type `Y` to disable the remote root login.
   d. Type `Y` to remove the test database.
   e. Type `Y` to reload the privilege tables and save your changes.
3. (Optional) If you do not plan to use the MariaDB server right away, stop it. You can restart it when you need it again.

```
[ec2-user ~]$ sudo systemctl stop mariadb
```

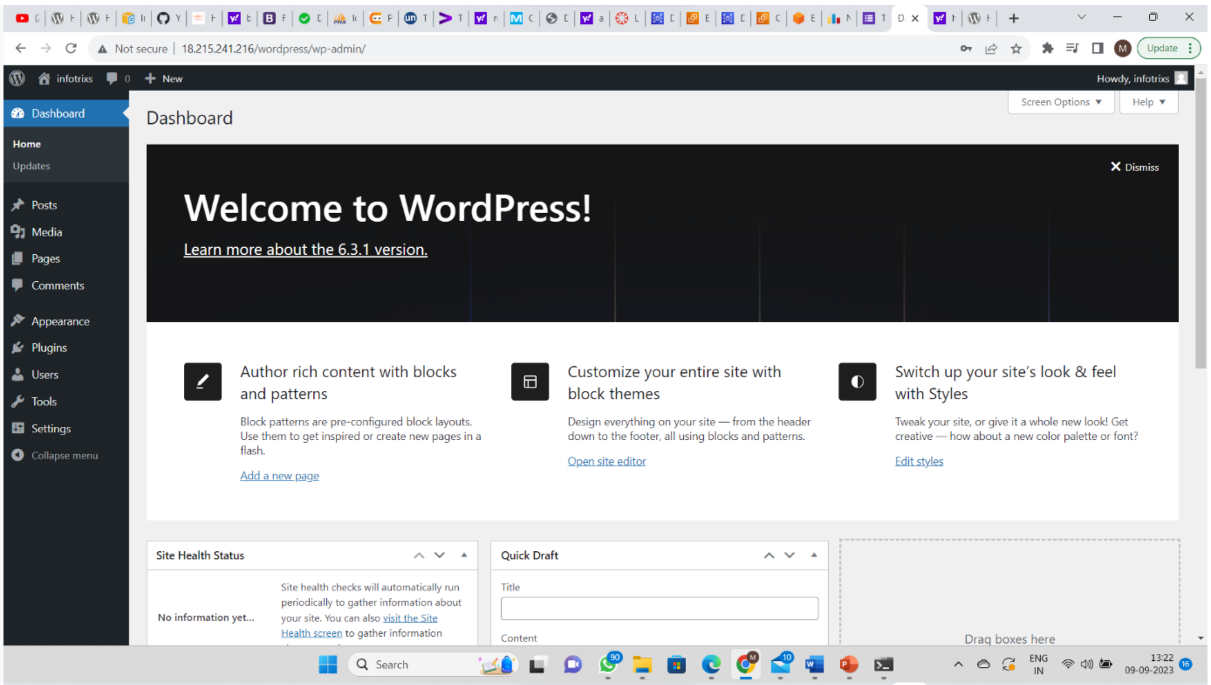
4. (Optional) If you want the MariaDB server to start at every boot, type the following command.

```
[ec2-user ~]$ sudo systemctl enable mariadb
```

Link………….

http://18.215.241.216/wordpress/

output



Not secure | 18.215.241.216/wordpress/wp-admin/

infotrixs 0 New Howdy, infotrixs

Dashboard
Home
Updates
Posts
Media
Pages
Comments
Appearance
Plugins
Users
Tools
Settings
Collapse menu

Dashboard

Screen Options ▾ Help ▾

Dismiss

Welcome to WordPress!

Learn more about the 6.3.1 version.

Author rich content with blocks and patterns

Block patterns are pre-configured block layouts. Use them to get inspired or create new pages in a flash.

Add a new page

Customize your entire site with block themes

Design everything on your site — from the header down to the footer, all using blocks and patterns.

Open site editor

Switch up your site's look & feel with Styles

Tweak your site, or give it a whole new look! Get creative — how about a new color palette or font?

Edit styles

Site Health Status

No information yet...

Site health checks will automatically run periodically to gather information about your site. You can also visit the Site Health screen to gather information

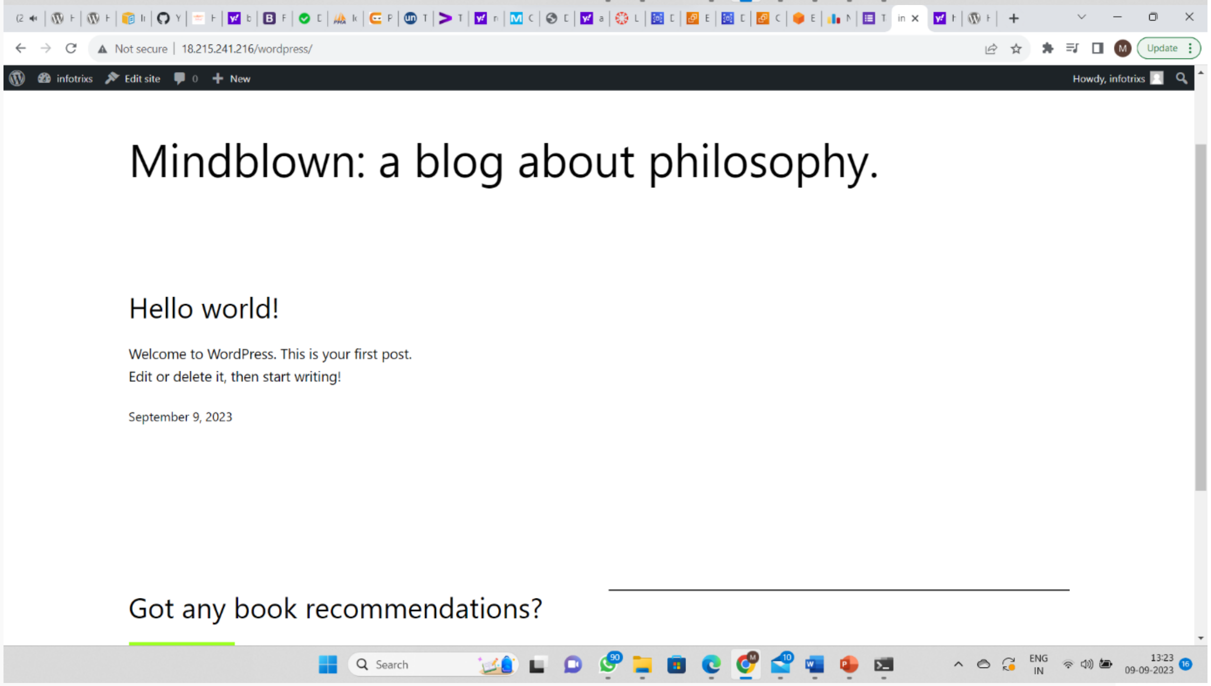
Quick Draft

Title

Content

Drag boxes here

13:22 09-09-2023



Not secure | 18.215.241.216/wordpress/

infotrixs Edit site 0 New Howdy, infotrixs

Mindblown: a blog about philosophy.

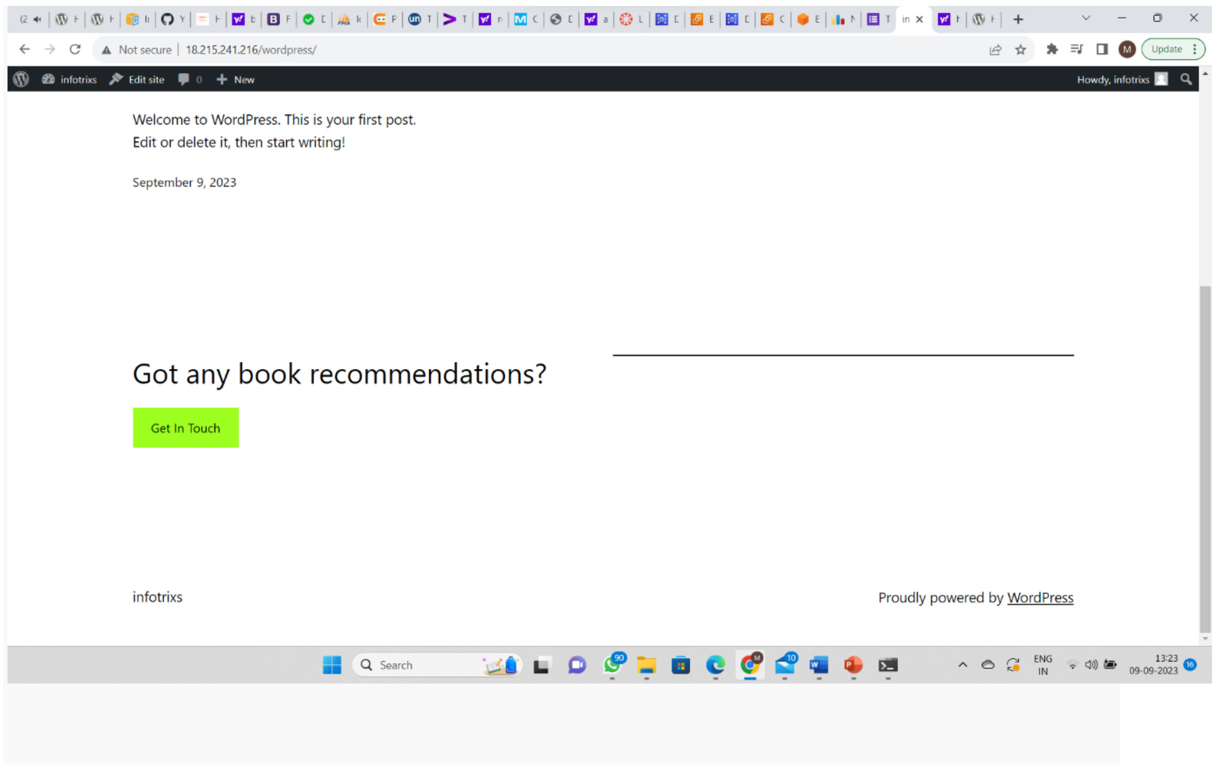
Hello world!

Welcome to WordPress. This is your first post.
Edit or delete it, then start writing!

September 9, 2023

Got any book recommendations?

13:23 09-09-2023



Not secure | 18.215.241.216/wordpress/

infotrixs Edit site 0 New Howdy, infotrixs

Welcome to WordPress. This is your first post.
Edit or delete it, then start writing!

September 9, 2023

Got any book recommendations?

Get In Touch

infotrixs

Proudly powered by WordPress

Task 2:

For microservices: 2 EC2 instance, 1 for wordpress and 1 for MYSQL

EC2 instance, 1 for wordpress

1. Open the Amazon EC2 console.

2. Click on "Launch Instance".

3. Choose an Amazon Machine Image (AMI) that supports Wordpress, such as Wordpress certified ubuntu automation:

....Check volume

4. Select an instance type based on your requirements.

5. Configure the instance details, such as network settings and storage.

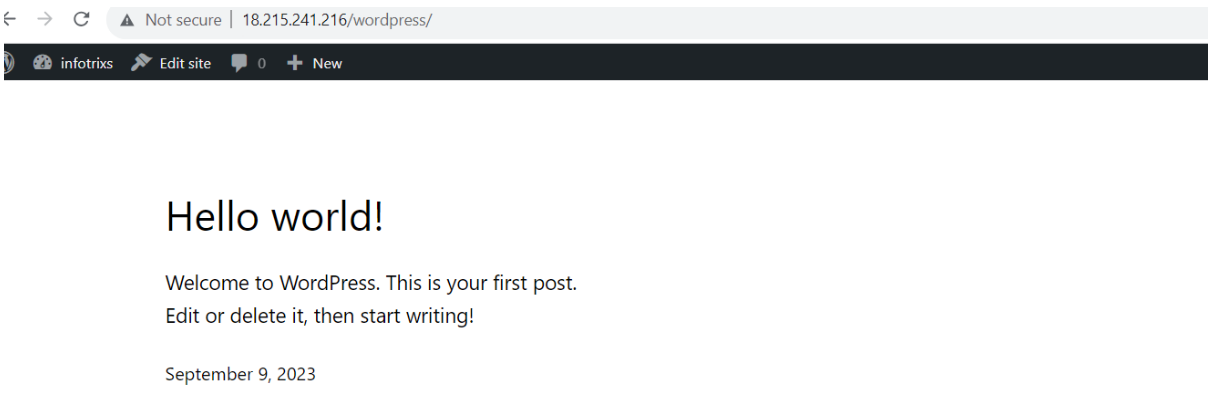
6. Add any additional storage volumes if needed.

8. Review and launch the instance.

9. Create or select an existing key pair for secure access to the instance.

10. Launch the instance and wait for it to start.

Output:



Hello world!

Welcome to WordPress. This is your first post.
Edit or delete it, then start writing!

September 9, 2023

EC2 instance, 1 for MYSQL

1. Open the Amazon EC2 console.

2. Click on "Launch Instance".

3. Choose an Amazon Machine Image (AMI) that supports MySQL, such as Amazon Linux or Ubuntu.

4. Select an instance type based on your requirements.

5. Configure the instance details, such as network settings and storage.

6. Add any additional storage volumes if needed.

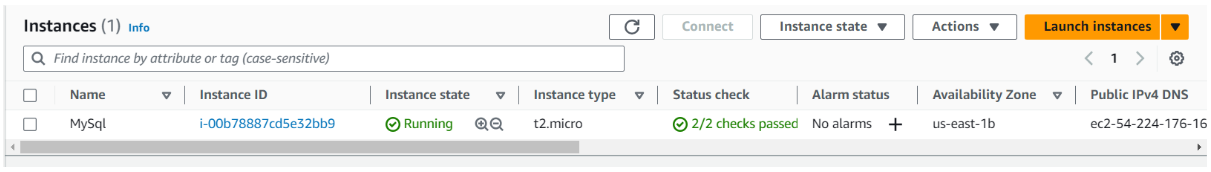
7. Configure security groups to allow access to the MySQL port (default is 3306).

8. Review and launch the instance.

9. Create or select an existing key pair for secure access to the instance.

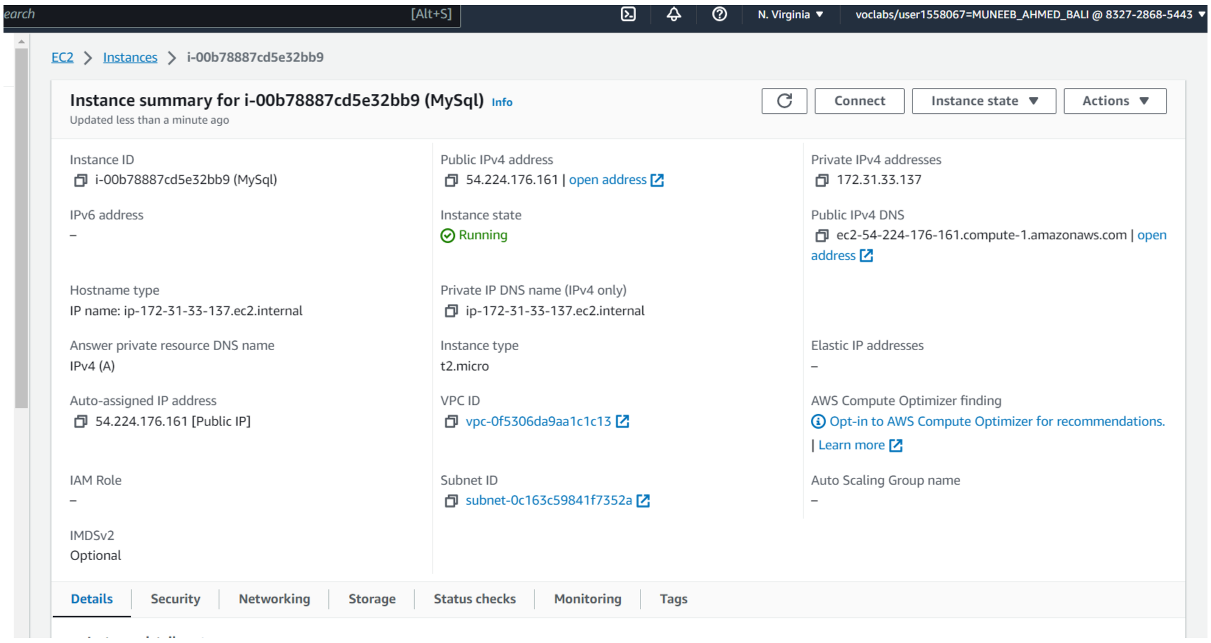
10. Launch the instance and wait for it to start.

Output:



Instances (1) Info

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
☐	MySql	i-00b78887cd5e32bb9	Running	t2.micro	2/2 checks passed	No alarms	us-east-1b	ec2-54-224-176-16



Instance summary for i-00b78887cd5e32bb9 (MySql) Info
Updated less than a minute ago

Instance ID i-00b78887cd5e32bb9 (MySql)	Public IPv4 address 54.224.176.161 | open address	Private IPv4 addresses 172.31.33.137
IPv6 address –	Instance state Running	Public IPv4 DNS ec2-54-224-176-161.compute-1.amazonaws.com | open address
Hostname type IP name: ip-172-31-33-137.ec2.internal	Private IP DNS name (IPv4 only) ip-172-31-33-137.ec2.internal	
Answer private resource DNS name IPv4 (A)	Instance type t2.micro	Elastic IP addresses –
Auto-assigned IP address 54.224.176.161 [Public IP]	VPC ID vpc-0f5306da9aa1c1c13	AWS Compute Optimizer finding Opt-in to AWS Compute Optimizer for recommendations. | Learn more
IAM Role –	Subnet ID subnet-0c163c59841f7352a	Auto Scaling Group name –
IMDSv2 Optional		

Commands used to host MySql Server on AWS EC2 Instance

Step 1: Update the system

```
sudo apt update
```

Step 2: Install MySql

```
sudo apt install mysql-server
```

Step 3: Check the Status of MySql (Active or Inactive)

```
sudo systemctl status mysql
```

Step 4: Login to MySql as a root

```
sudo mysql
```

Step 5: Update the password for the MySql Server

```
ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY 'place-
your-password-here';
```

```
FLUSH PRIVILEGES;
```

Step 6: Test the MySql server if it is working by running sample sql queries

```
CREATE DATABASE mysql_test;
```

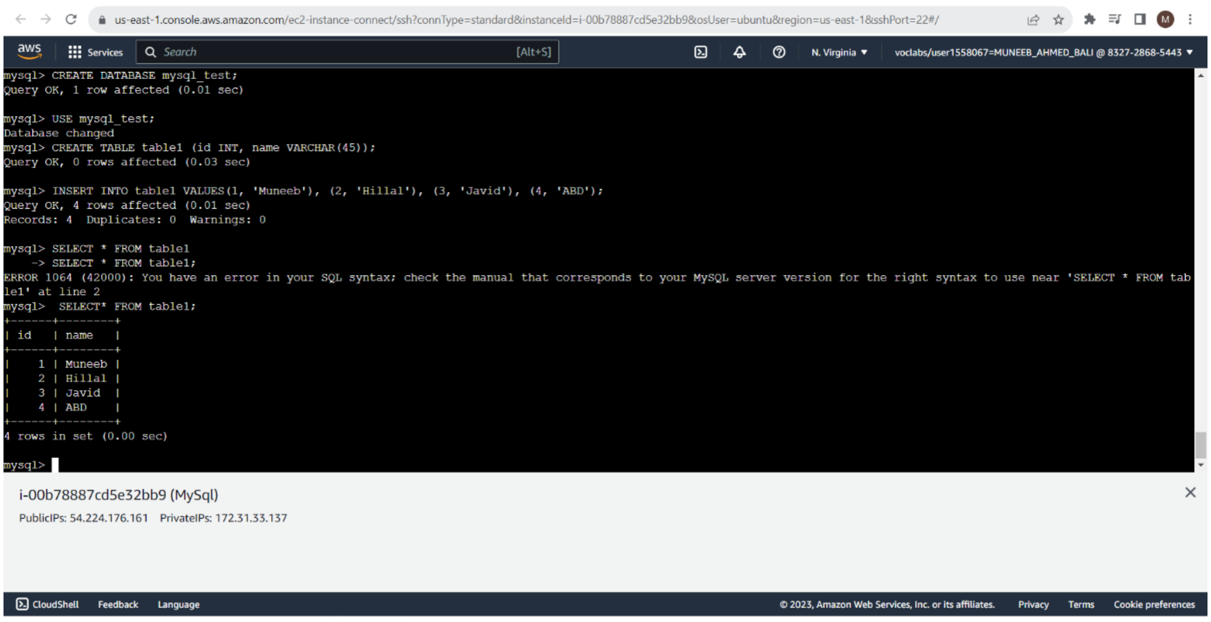
```
USE mysql_test;
```

```
CREATE TABLE table1 (id INT, name VARCHAR (45));
```

```
INSERT INTO table1 VALUES (1, 'Muneeb'), (2, 'Hillal'), (3, 'Javid'), (4, 'ABD');
SELECT * FROM table1;
```

Output:

Link: 18.215.241.216



us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-00b78887cd5e32bb9&osUser=ubuntu&region=us-east-1&sshPort=22#/
aws
Services
Search
[Alt+S]
N. Virginia
voclabs/user1558067=MUNEEB_AHMED_BALI @ 8327-2868-5443

```
mysql> CREATE DATABASE mysql_test;
Query OK, 1 row affected (0.01 sec)

mysql> USE mysql_test;
Database changed
mysql> CREATE TABLE table1 (id INT, name VARCHAR(45));
Query OK, 0 rows affected (0.03 sec)

mysql> INSERT INTO table1 VALUES(1, 'Muneeb'), (2, 'Hillal'), (3, 'Javid'), (4, 'ABD');
Query OK, 4 rows affected (0.01 sec)
Records: 4  Duplicates: 0  Warnings: 0

mysql> SELECT * FROM table1
    -> SELECT * FROM table1;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'SELECT * FROM tab
le1' at line 2
mysql>  SELECT* FROM table1;
+------+--------+
| id   | name   |
+------+--------+
|    1 | Muneeb |
|    2 | Hillal |
|    3 | Javid  |
|    4 | ABD    |
+------+--------+
4 rows in set (0.00 sec)

mysql> 
```

i-00b78887cd5e32bb9 (MySql)

PublicIPs: 54.224.176.161   PrivateIPs: 172.31.33.137