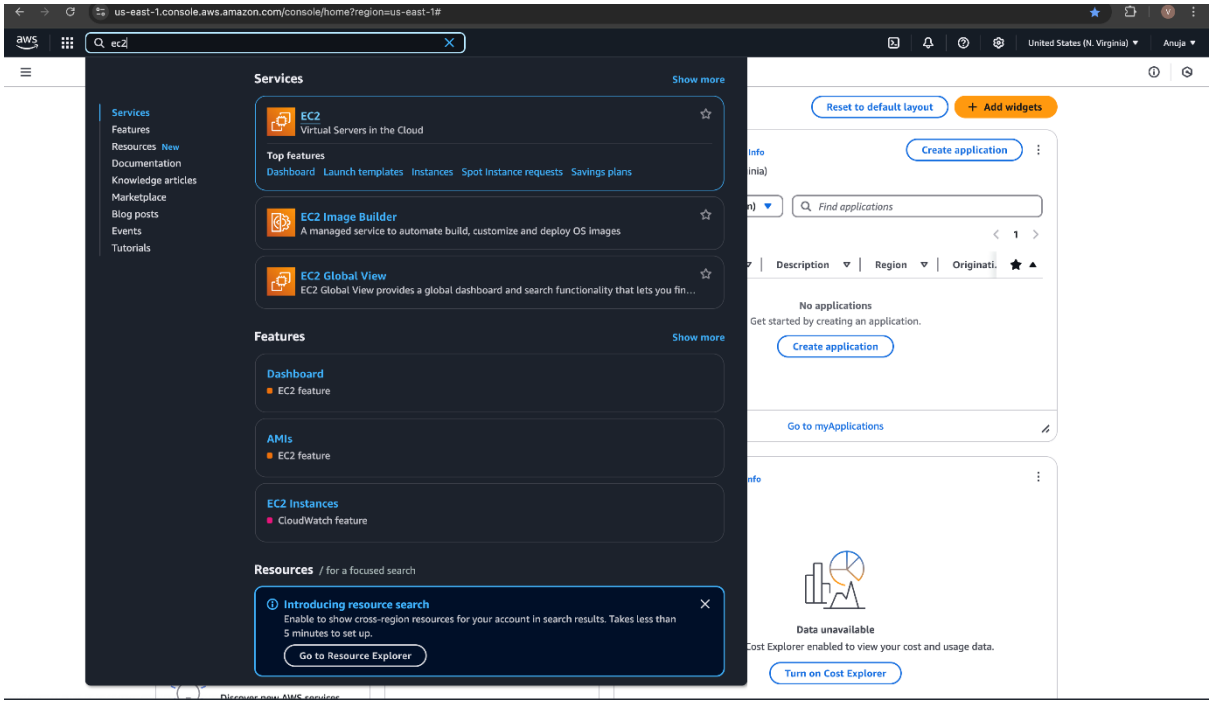
Word Press Website Creation

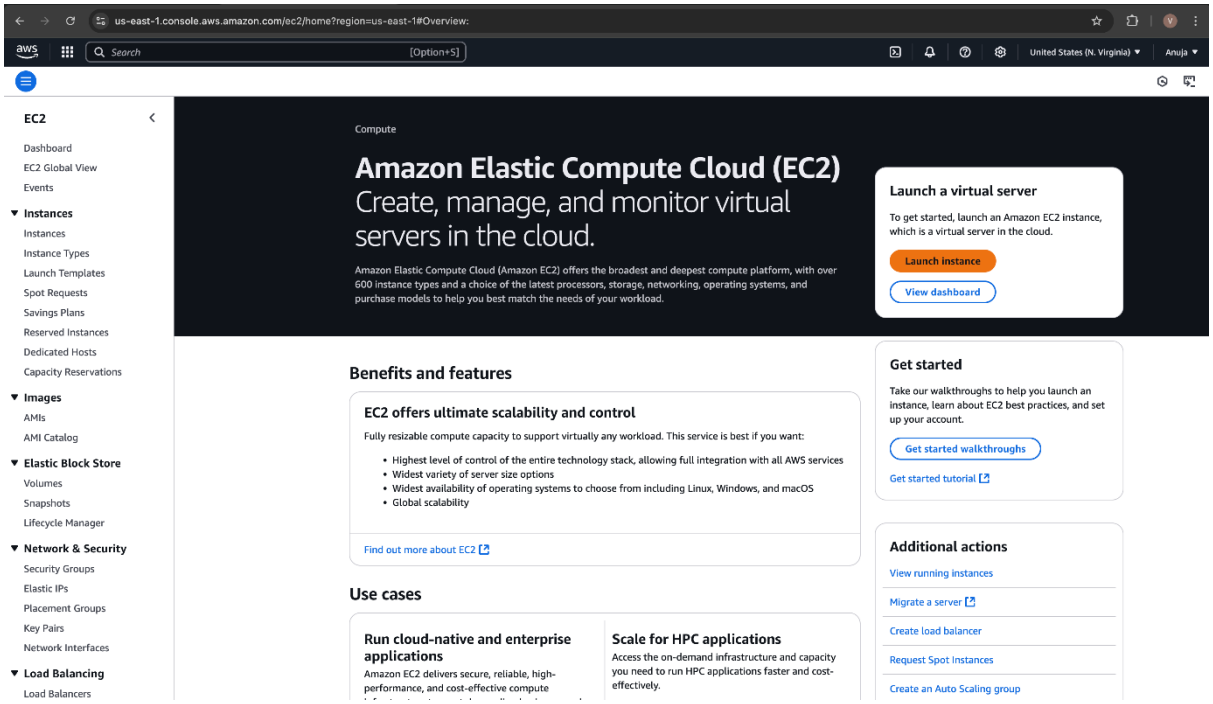
Using EC2 Creating WordPress:

First we need to create one Ec2 Linux instance using AWS console.

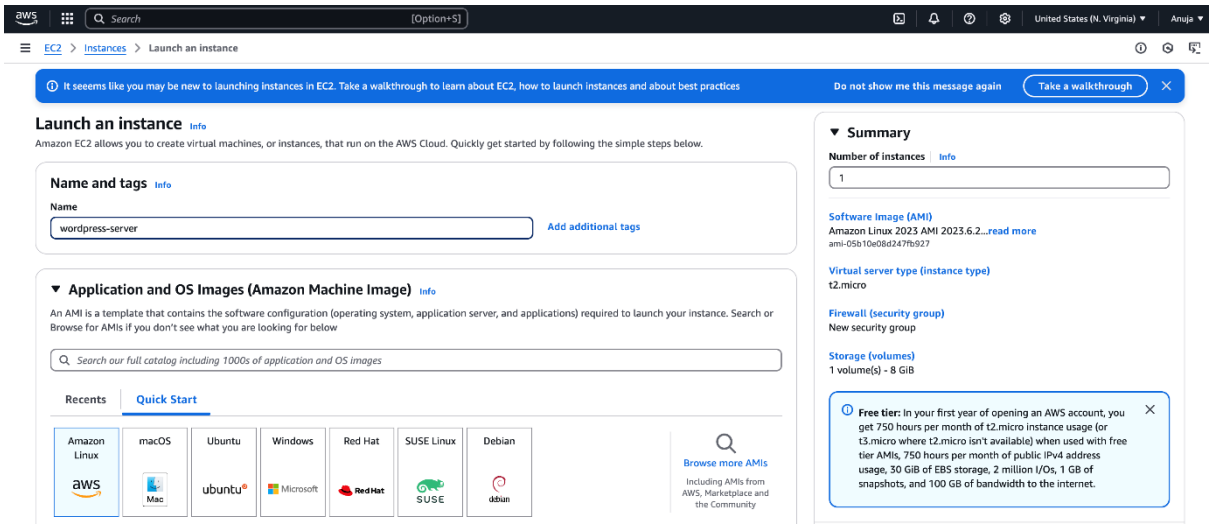
Go to AWS and search for EC2.



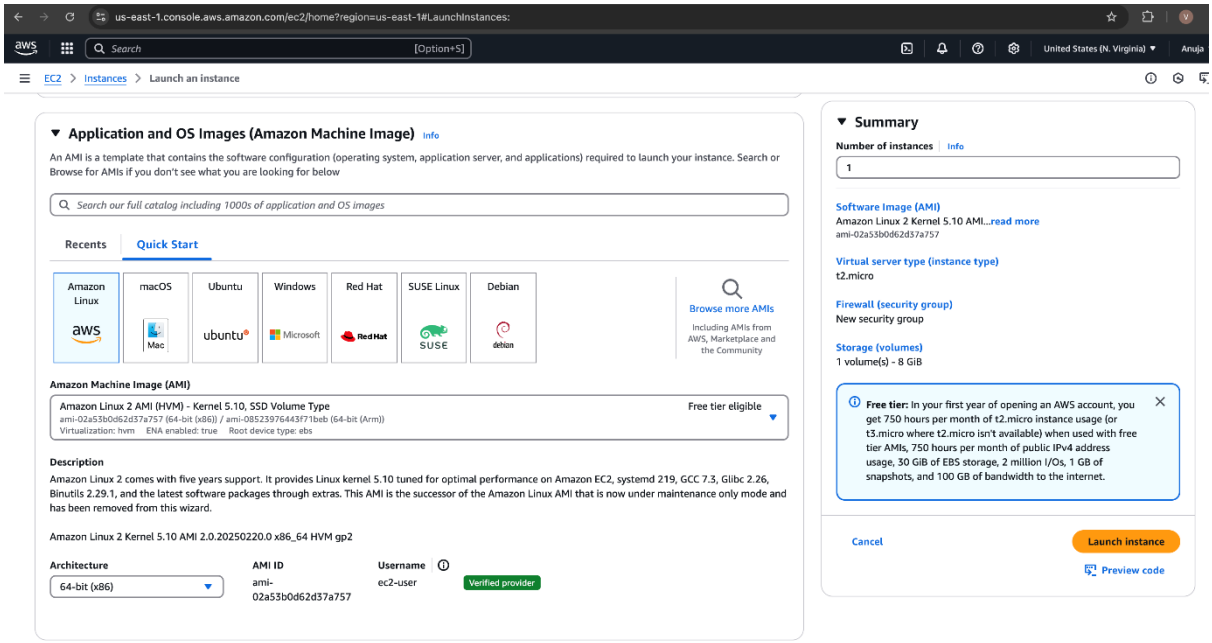
And then press on launch instance.



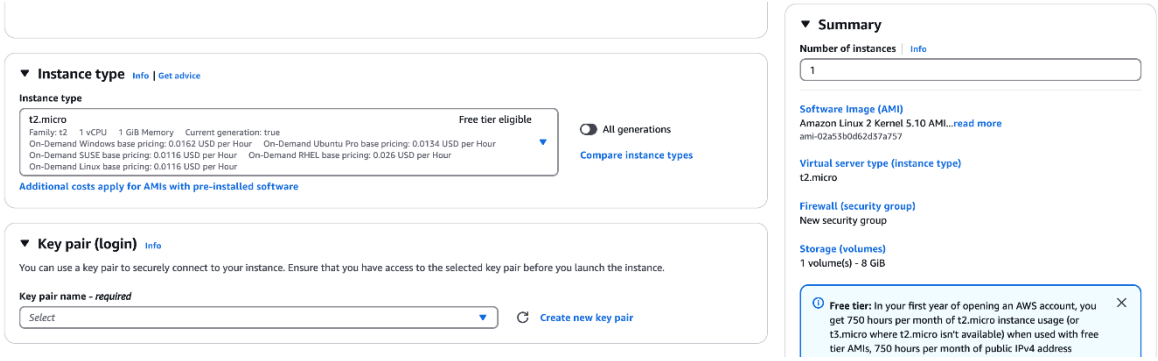
Give name of your server as wordpress-server



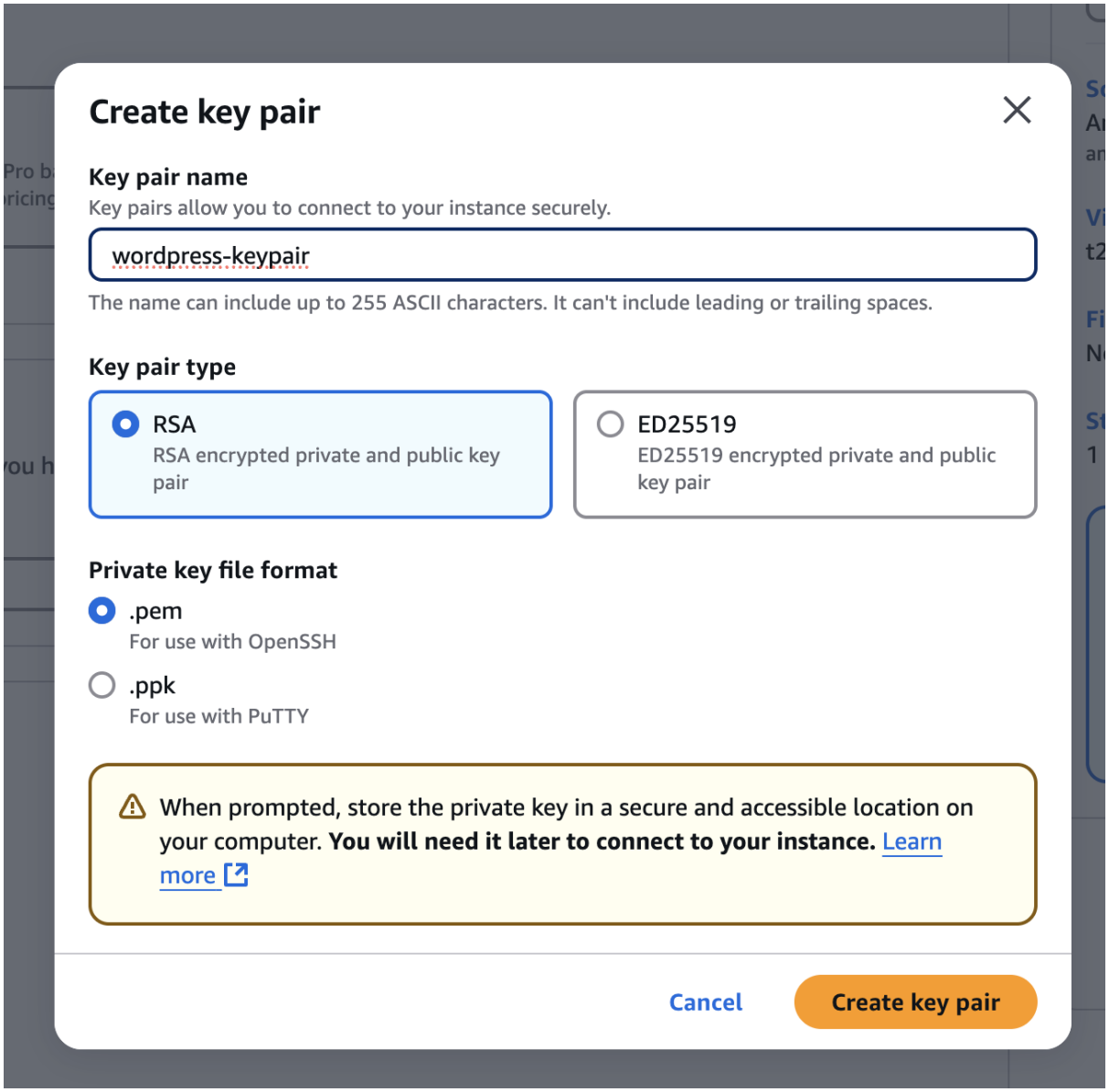
And give application and OS images here I am selecting Image as Amazon Linux 2 AMI 64 bit.



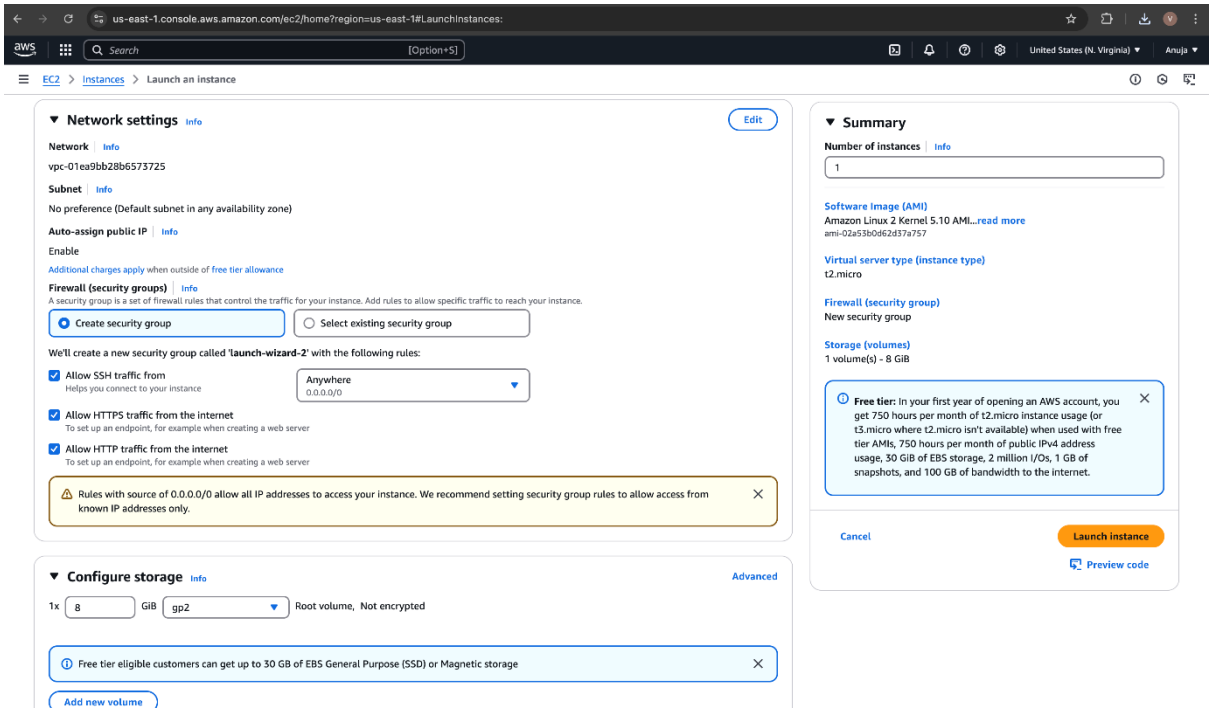
Select instance type as T2. micro and in keypair section press on create new keypair.



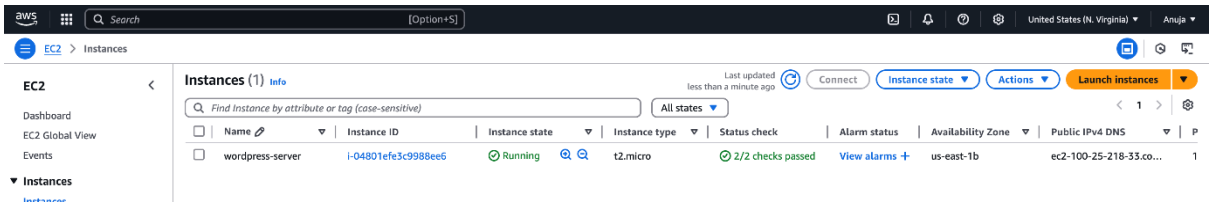
Give keypair name as wordpress-keypair keypair type as RSA and private key file format as .pem to work with open ssh. And press create keypair.



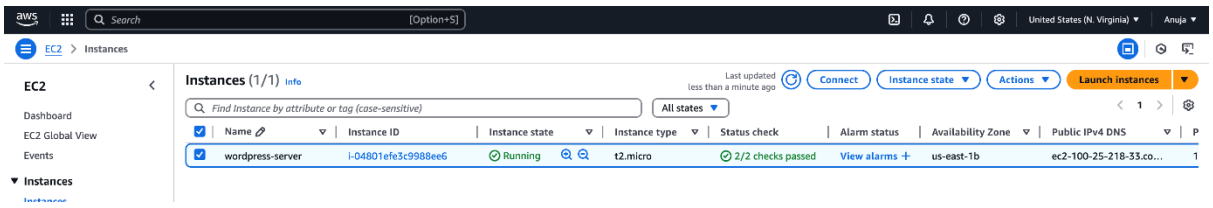
In network setting give security group as create security group and allow ssh, http and https traffic. And press on launch instance.



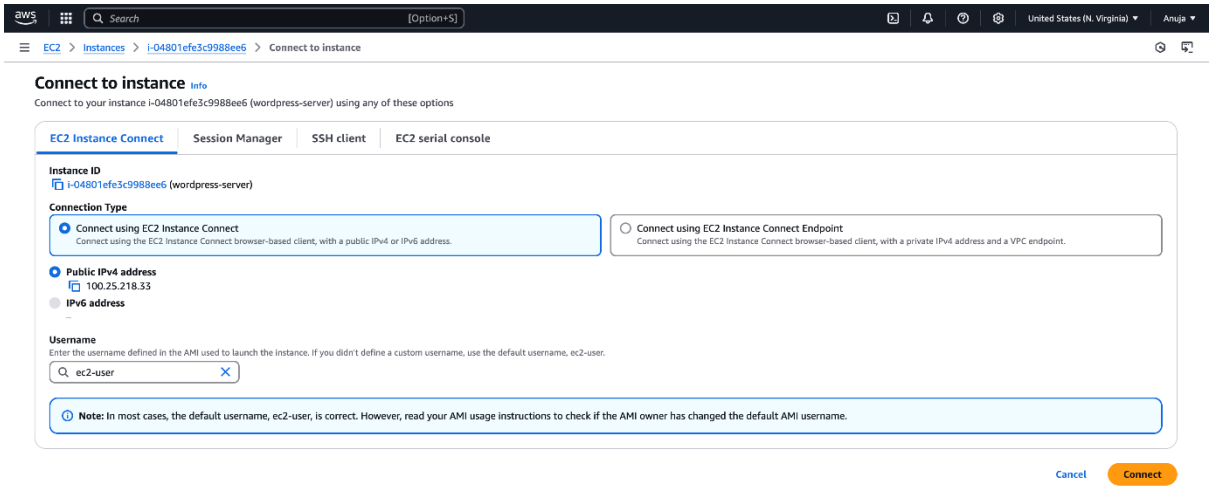
Your instance will launch and it will take some time to get running.



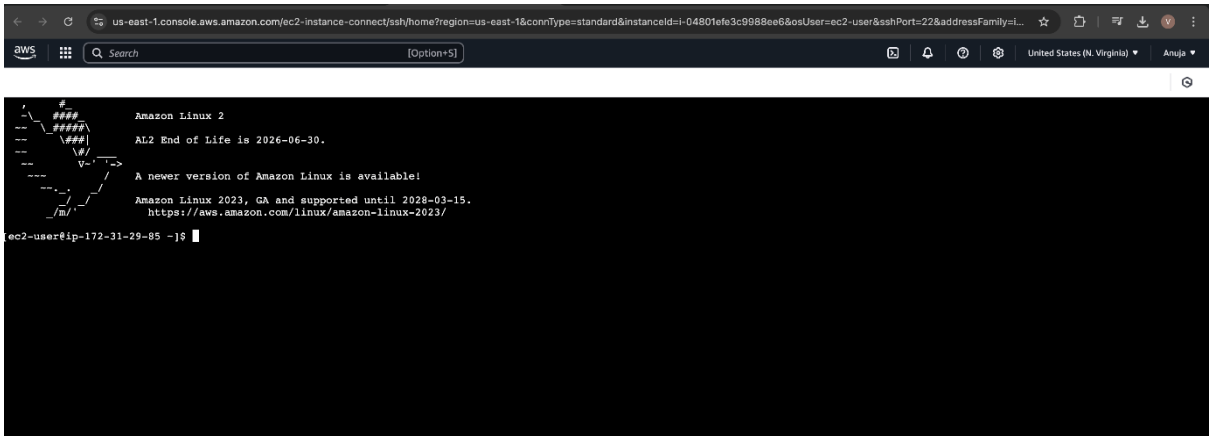
And press the instance and click on connect button.



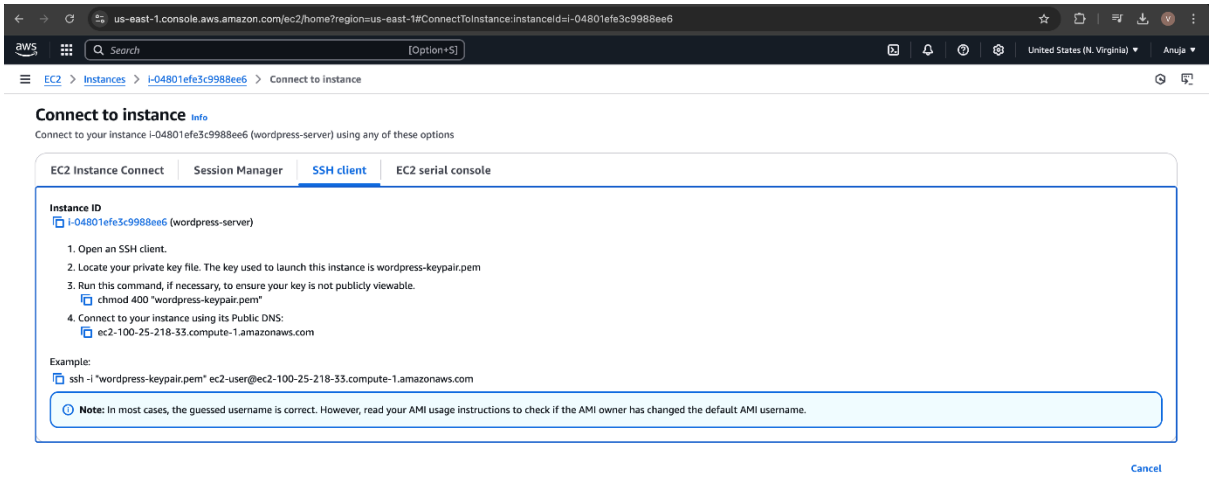
And then connect to Ec2 instance connect it will look like this.



And then press on connect then we will get this prompt.



Another way to create EC2 instance is using SSH client.



Then press on connect then you can have another way to create your instance.

```
Last login: Wed Feb 26 11:44:09 on console
raj@Rajs-Laptop ~ % cd desktop
raj@Rajs-Laptop desktop % chmod 400 "wordpress-keypair.pem"
raj@Rajs-Laptop desktop %
```

After login in you can change the permission of your keypair using this command.
chmod 400 ‘’wordpress-keypair.pem’’

Then connect your instance using this command.
ssh -i “wordpress-keypair.pem” ec2-user@ec2-100-25-218-33.compute-1.amazonaws.com

You can get this command on your AWS EC2 instance connect.

```
Last login: Wed Feb 26 11:44:09 on console
raj@Rajs-Laptop ~ % cd desktop
raj@Rajs-Laptop desktop % chmod 400 "wordpress-keypair.pem"
raj@Rajs-Laptop desktop % ssh -i "wordpress-keypair.pem" ec2-user@ec2-100-25-218-33.compute-1.amazonaws.com
The authenticity of host 'ec2-100-25-218-33.compute-1.amazonaws.com (100.25.218.33)' can't be established.
ED25519 key fingerprint is SHA256:sBoRkbRW2hEKs4WGsqLyQwriDwJs/4txm+PTKj/+Fuw.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-100-25-218-33.compute-1.amazonaws.com' (ED25519) to the list of known hosts.
Last login: Wed Feb 26 06:22:56 2025 from ec2-18-206-107-28.compute-1.amazonaws.com
   ,     #_
   ~\_  ####_        Amazon Linux 2
  ~~  \_#####\
  ~~     \###|       AL2 End of Life is 2026-06-30.
  ~~       \#/ ___
   ~~       V~' '->
    ~~~         /    A newer version of Amazon Linux is available!
      ~~._.   _/
         _/ _/       Amazon Linux 2023, GA and supported until 2028-03-15.
       _/m/'           https://aws.amazon.com/linux/amazon-linux-2023/

[ec2-user@ip-172-31-29-85 ~]$
```

Till now we have EC2 machine ready with us now we have to install Apache webserver in this EC2 Linux machine.

First we have to update the yum package using sudo yum update -y .

```
[ec2-user@ip-172-31-29-85 ~]$ sudo yum update -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core                                                               | 3.6 kB  00:00:00
No packages marked for update
[ec2-user@ip-172-31-29-85 ~]$
```

After that start installing Apache server in your ec2 machine. Using the following command.
sudo yum install -y httpd

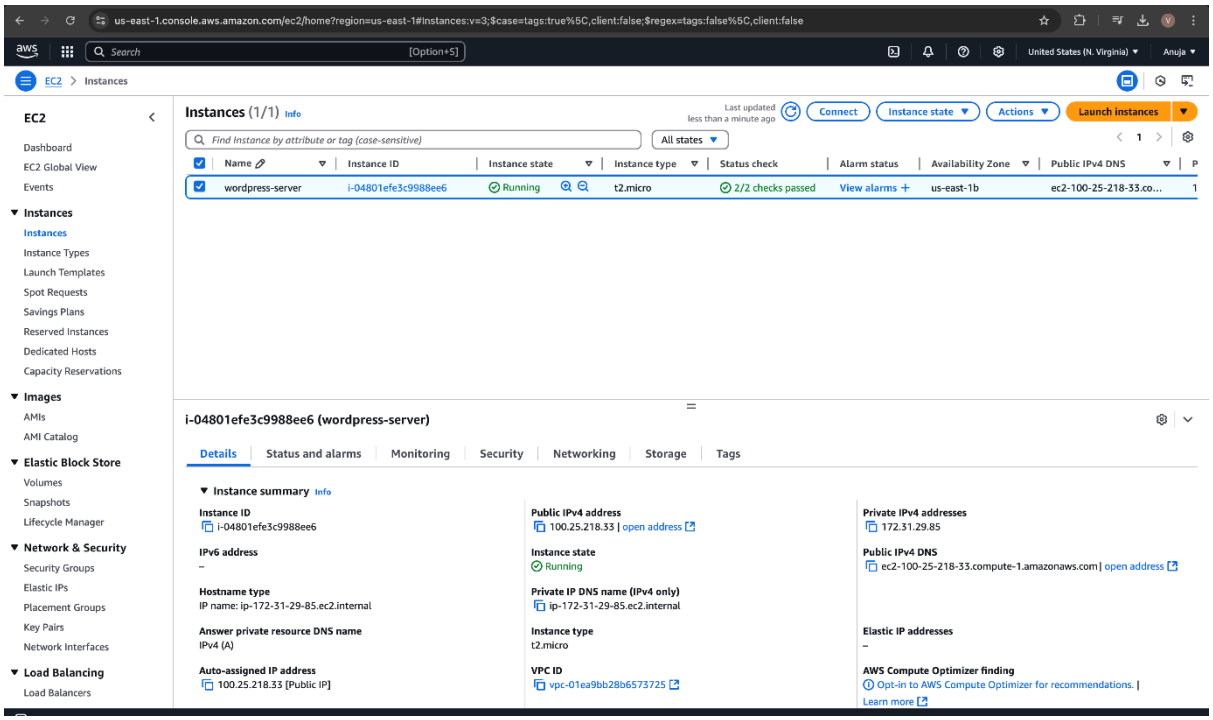
```
[ec2-user@ip-172-31-29-85 ~]$ sudo yum install -y httpd
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Resolving Dependencies
--> Running transaction check
---> Package httpd.x86_64 0:2.4.62-1.amzn2.0.2 will be installed
--> Processing Dependency: httpd-filesystem = 2.4.62-1.amzn2.0.2 for package: httpd-2.4.62-1.amzn2.0.2.x86_64
--> Processing Dependency: httpd-tools = 2.4.62-1.amzn2.0.2 for package: httpd-2.4.62-1.amzn2.0.2.x86_64
--> Processing Dependency: /etc/mime.types for package: httpd-2.4.62-1.amzn2.0.2.x86_64
--> Processing Dependency: httpd-filesystem for package: httpd-2.4.62-1.amzn2.0.2.x86_64
--> Processing Dependency: mod_http2 for package: httpd-2.4.62-1.amzn2.0.2.x86_64
--> Processing Dependency: system-logos-httpd for package: httpd-2.4.62-1.amzn2.0.2.x86_64
--> Processing Dependency: libapr-1.so.0()(64bit) for package: httpd-2.4.62-1.amzn2.0.2.x86_64
--> Processing Dependency: libaprutil-1.so.0()(64bit) for package: httpd-2.4.62-1.amzn2.0.2.x86_64
--> Running transaction check
---> Package apr.x86_64 0:1.7.2-1.amzn2.0.1 will be installed
---> Package apr-util.x86_64 0:1.6.3-1.amzn2.0.1 will be installed
--> Processing Dependency: apr-util-bdb(x86-64) = 1.6.3-1.amzn2.0.1 for package: apr-util-1.6.3-1.amzn2.0.1.x86_64
---> Package generic-logos-httpd.noarch 0:18.0.0-4.amzn2 will be installed
---> Package httpd-filesystem.noarch 0:2.4.62-1.amzn2.0.2 will be installed
---> Package httpd-tools.x86_64 0:2.4.62-1.amzn2.0.2 will be installed
---> Package mailcap.noarch 0:2.1.41-2.amzn2 will be installed
---> Package mod_http2.x86_64 0:1.15.19-1.amzn2.0.2 will be installed
--> Running transaction check
---> Package apr-util-bdb.x86_64 0:1.6.3-1.amzn2.0.1 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================
 Package                Arch      Version                   Repository     Size
================================================================================
Installing:
 httpd                  x86_64    2.4.62-1.amzn2.0.2        amzn2-core    1.4 M
Installing for dependencies:
 apr                    x86_64    1.7.2-1.amzn2.0.1         amzn2-core    130 k
 apr-util               x86_64    1.6.3-1.amzn2.0.1         amzn2-core    101 k
 apr-util-bdb           x86_64    1.6.3-1.amzn2.0.1         amzn2-core     22 k
 generic-logos-httpd    noarch    18.0.0-4.amzn2            amzn2-core     19 k
 httpd-filesystem       noarch    2.4.62-1.amzn2.0.2        amzn2-core     25 k
 httpd-tools            x86_64    2.4.62-1.amzn2.0.2        amzn2-core     89 k
 mailcap                noarch    2.1.41-2.amzn2            amzn2-core     31 k
 mod_http2              x86_64    1.15.19-1.amzn2.0.2       amzn2-core    149 k
```

```
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-filesys
tem.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-172-31-29-85 ~]$
```

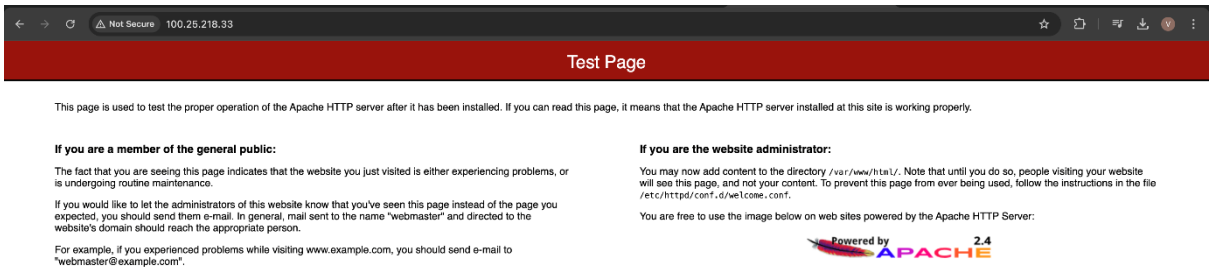
After that start the service using this command sudo service httpd start

```
[ec2-user@ip-172-31-29-85 ~]$ sudo service httpd start
Redirecting to /bin/systemctl start httpd.service
[ec2-user@ip-172-31-29-85 ~]$
```

Then open your EC2 at down you will find public IP copy that Ip.



Open that public IP in a browser. If you get Apache test page then your Apache server is ready.



And next step is to install php in the ec2 server. Using this command.

sudo amazon-linux-extras install php8.2 -y

```
[ec2-user@ip-172-31-29-85 ~]$ sudo amazon-linux-extras install php8.2 -y
```

After installation it will show as installation complete.

```
Installed:
  php-cli.x86_64 0:8.2.27-1.amzn2.0.5      php-common.x86_64 0:8.2.27-1.amzn2.0.5  php-fpm.x86_64 0:8.2.27-1.amzn2.0.5
  php-mysqlnd.x86_64 0:8.2.27-1.amzn2.0.5  php-pdo.x86_64 0:8.2.27-1.amzn2.0.5

Dependency Installed:
  libzip.x86_64 0:1.3.2-1.amzn2.0.1

Complete!
  2  httpd_modules            available    [ =1.0  =stable ]
  3  memcached1.5             available    \
       [ =1.5.1  =1.5.16  =1.5.17 ]
  9  R3.4                     available    [ =3.4.3  =stable ]
 10  rust1                    available    \
       [ =1.22.1  =1.26.0  =1.26.1  =1.27.2  =1.31.0  =1.38.0
         =stable ]
 18  libreoffice              available    \
       [ =5.0.6.2_15  =5.3.6.1  =stable ]
 19  gimp                     available    [ =2.8.22 ]
 20 +docker=latest            enabled      \
```

You can see the installation status using the php -v command.

```
[ec2-user@ip-172-31-29-85 ~]$ php -v
PHP 8.2.27 (cli) (built: Feb  6 2025 17:35:56) (NTS)
Copyright (c) The PHP Group
Zend Engine v4.2.27, Copyright (c) Zend Technologies
[ec2-user@ip-172-31-29-85 ~]$
```

And next we need to install MariaDB SQL server using the following command.
sudo amazon-linux-extras install mariadb10.5 -y

```
[ec2-user@ip-172-31-29-85 ~]$ sudo amazon-linux-extras install mariadb10.5 -y
Topic mariadb10.5 has end-of-support date of 2025-06-24
Installing mariadb
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Cleaning repos: amzn2-core amzn2extra-docker amzn2extra-kernel-5.10 amzn2extra-mariadb10.5 amzn2extra-php8.2
22 metadata files removed
8 sqlite files removed
0 metadata files removed
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core                                                                | 3.6 kB  00:00:00
amzn2extra-docker                                                         | 2.9 kB  00:00:00
amzn2extra-kernel-5.10                                                    | 3.0 kB  00:00:00
amzn2extra-mariadb10.5                                                    | 3.0 kB  00:00:00
amzn2extra-php8.2                                                         | 2.9 kB  00:00:00
(1/11): amzn2-core/2/x86_64/group_gz                                      | 2.7 kB  00:00:00
(2/11): amzn2-core/2/x86_64/updateinfo                                    | 1.0 MB  00:00:00
(3/11): amzn2extra-docker/2/x86_64/updateinfo                             |  22 kB  00:00:00
(4/11): amzn2extra-mariadb10.5/2/x86_64/updateinfo                        | 7.8 kB  00:00:00
(5/11): amzn2extra-kernel-5.10/2/x86_64/updateinfo                        | 104 kB  00:00:00
(6/11): amzn2extra-docker/2/x86_64/primary_db                             | 120 kB  00:00:00
(7/11): amzn2extra-php8.2/2/x86_64/updateinfo                             | 6.6 kB  00:00:00
(8/11): amzn2extra-php8.2/2/x86_64/primary_db                             | 111 kB  00:00:00
(9/11): amzn2extra-mariadb10.5/2/x86_64/primary_db                        | 129 kB  00:00:00
(10/11): amzn2extra-kernel-5.10/2/x86_64/primary_db                       |  34 MB  00:00:01
(11/11): amzn2-core/2/x86_64/primary_db                                   |  74 MB  00:00:01
Resolving Dependencies
--> Running transaction check
```

After installation we can see the below installation completed status.

```
Installed:
  mariadb.x86_64 3:10.5.25-1.amzn2

Dependency Installed:
  Judy.x86_64 0:1.0.5-8.amzn2.0.1                          keyutils-libs-devel.x86_64 0:1.5.8-3.amzn2.0.2
  krb5-devel.x86_64 0:1.15.1-55.amzn2.2.8                  libcom_err-devel.x86_64 0:1.42.9-19.amzn2.0.1
  libkadm5.x86_64 0:1.15.1-55.amzn2.2.8                    libselinux-devel.x86_64 0:2.5-12.amzn2.0.2
  libsepol-devel.x86_64 0:2.5-10.amzn2.0.1                 libsphinxclient.x86_64 0:2.2.11-5.amzn2.0.1
  libverto-devel.x86_64 0:0.2.5-4.amzn2.0.2                mariadb-backup.x86_64 3:10.5.25-1.amzn2
  mariadb-common.x86_64 3:10.5.25-1.amzn2                  mariadb-config.x86_64 3:10.5.25-1.amzn2
  mariadb-connect-engine.x86_64 3:10.5.25-1.amzn2          mariadb-cracklib-password-check.x86_64 3:10.5.25-1.amzn2
  mariadb-devel.x86_64 3:10.5.25-1.amzn2                   mariadb-errmsg.x86_64 3:10.5.25-1.amzn2
  mariadb-gssapi-server.x86_64 3:10.5.25-1.amzn2           mariadb-oqgraph-engine.x86_64 3:10.5.25-1.amzn2
  mariadb-pam.x86_64 3:10.5.25-1.amzn2                     mariadb-rocksdb-engine.x86_64 3:10.5.25-1.amzn2
  mariadb-server.x86_64 3:10.5.25-1.amzn2                  mariadb-server-utils.x86_64 3:10.5.25-1.amzn2
  mariadb-sphinx-engine.x86_64 3:10.5.25-1.amzn2           mytop.noarch 0:1.7-20.b737f60.amzn2
  openssl-devel.x86_64 1:1.0.2k-24.amzn2.0.14              pcre-devel.x86_64 0:8.32-17.amzn2.0.3
  perl-Compress-Raw-Bzip2.x86_64 0:2.061-3.amzn2.0.2       perl-Compress-Raw-Zlib.x86_64 1:2.061-4.amzn2.0.2
  perl-DBD-MySQL.x86_64 0:4.023-6.amzn2                    perl-DBI.x86_64 0:1.627-4.amzn2.0.2
  perl-Data-Dumper.x86_64 0:2.145-3.amzn2.0.2              perl-IO-Compress.noarch 0:2.061-2.amzn2
  perl-Net-Daemon.noarch 0:0.48-5.amzn2                    perl-PlRPC.noarch 0:0.2020-14.amzn2
  perl-TermReadKey.x86_64 0:2.30-20.amzn2.0.2              postgresql-libs.x86_64 0:9.2.24-8.amzn2.0.5
  sphinx.x86_64 0:2.2.11-5.amzn2.0.1                       zlib-devel.x86_64 0:1.2.7-19.amzn2.0.3

Dependency Updated:
  mariadb-libs.x86_64 3:10.5.25-1.amzn2

Complete!
  2  httpd_modules            available    [ =1.0  =stable ]
  3  memcached1.5             available    \
       [ =1.5.1  =1.5.16  =1.5.17 ]
  9  R3.4                     available    [ =3.4.3  =stable ]
 10  rust1                    available    \
       [ =1.22.1  =1.26.0  =1.26.1  =1.27.2  =1.31.0  =1.38.0
```

start MariaDB using the following command.

sudo systemctl start mariadb

```
[ec2-user@ip-172-31-29-85 ~]$ sudo systemctl start mariadb
[ec2-user@ip-172-31-29-85 ~]$
```

you can see the completion status as below.

```
[ec2-user@ip-172-31-29-85 ~]$ sudo service mariadb status
Redirecting to /bin/systemctl status mariadb.service
● mariadb.service - MariaDB 10.5 database server
   Loaded: loaded (/usr/lib/systemd/system/mariadb.service; disabled; vendor preset: disabled)
   Active: active (running) since Wed 2025-02-26 06:59:16 UTC; 53s ago
     Docs: man:mariadbd(8)
           https://mariadb.com/kb/en/library/systemd/
  Process: 5027 ExecStartPost=/usr/libexec/mariadb-check-upgrade (code=exited, status=0/SUCCESS)
  Process: 4863 ExecStartPre=/usr/libexec/mariadb-prepare-db-dir %n (code=exited, status=0/SUCCESS)
  Process: 4839 ExecStartPre=/usr/libexec/mariadb-check-socket (code=exited, status=0/SUCCESS)
 Main PID: 4973 (mariadbd)
   Status: "Taking your SQL requests now..."
   CGroup: /system.slice/mariadb.service
           └─4973 /usr/libexec/mariadbd --basedir=/usr

Feb 26 06:59:16 ip-172-31-29-85.ec2.internal mariadb-prepare-db-dir[4863]: The second is mysql@localhost, it has no ...ut
Feb 26 06:59:16 ip-172-31-29-85.ec2.internal mariadb-prepare-db-dir[4863]: you need to be the system 'mysql' user to...t.
Feb 26 06:59:16 ip-172-31-29-85.ec2.internal mariadb-prepare-db-dir[4863]: After connecting you can set the password...be
Feb 26 06:59:16 ip-172-31-29-85.ec2.internal mariadb-prepare-db-dir[4863]: able to connect as any of these users wit...do
Feb 26 06:59:16 ip-172-31-29-85.ec2.internal mariadb-prepare-db-dir[4863]: See the MariaDB Knowledgebase at https://...kb
Feb 26 06:59:16 ip-172-31-29-85.ec2.internal mariadb-prepare-db-dir[4863]: Please report any problems at https://mar...ra
Feb 26 06:59:16 ip-172-31-29-85.ec2.internal mariadb-prepare-db-dir[4863]: The latest information about MariaDB is a.../.
Feb 26 06:59:16 ip-172-31-29-85.ec2.internal mariadb-prepare-db-dir[4863]: Consider joining MariaDB's strong and vib...y:
Feb 26 06:59:16 ip-172-31-29-85.ec2.internal mariadb-prepare-db-dir[4863]: https://mariadb.org/get-involved/
Feb 26 06:59:16 ip-172-31-29-85.ec2.internal systemd[1]: Started MariaDB 10.5 database server.
Hint: Some lines were ellipsized, use -l to show in full.
[ec2-user@ip-172-31-29-85 ~]$
```

Now we are all set to install our wordpress server in our ec2 server. First, we need to change our path and make sure we are installing all in correct place.

```
[ec2-user@ip-172-31-29-85 ~]$ cd ~
[ec2-user@ip-172-31-29-85 ~]$ pwd
/home/ec2-user
[ec2-user@ip-172-31-29-85 ~]$
```

Download all wordpress files using the following command.

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

```
[ec2-user@ip-172-31-29-85 ~]$ wget https://wordpress.org/latest.tar.gz
--2025-02-26 07:02:31--  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: 26780969 (26M) [application/octet-stream]
Saving to: 'latest.tar.gz'

 0%  0           --.-K/s                5%  1,510,991   7.20MB/s              55%  14,831,183  35.2MB/s
00% 26,780,969  43.6MB/s   in 0.6s

2025-02-26 07:02:31 (43.6 MB/s) - 'latest.tar.gz' saved [26780969/26780969]

[ec2-user@ip-172-31-29-85 ~]$ ls
latest.tar.gz
[ec2-user@ip-172-31-29-85 ~]$
```

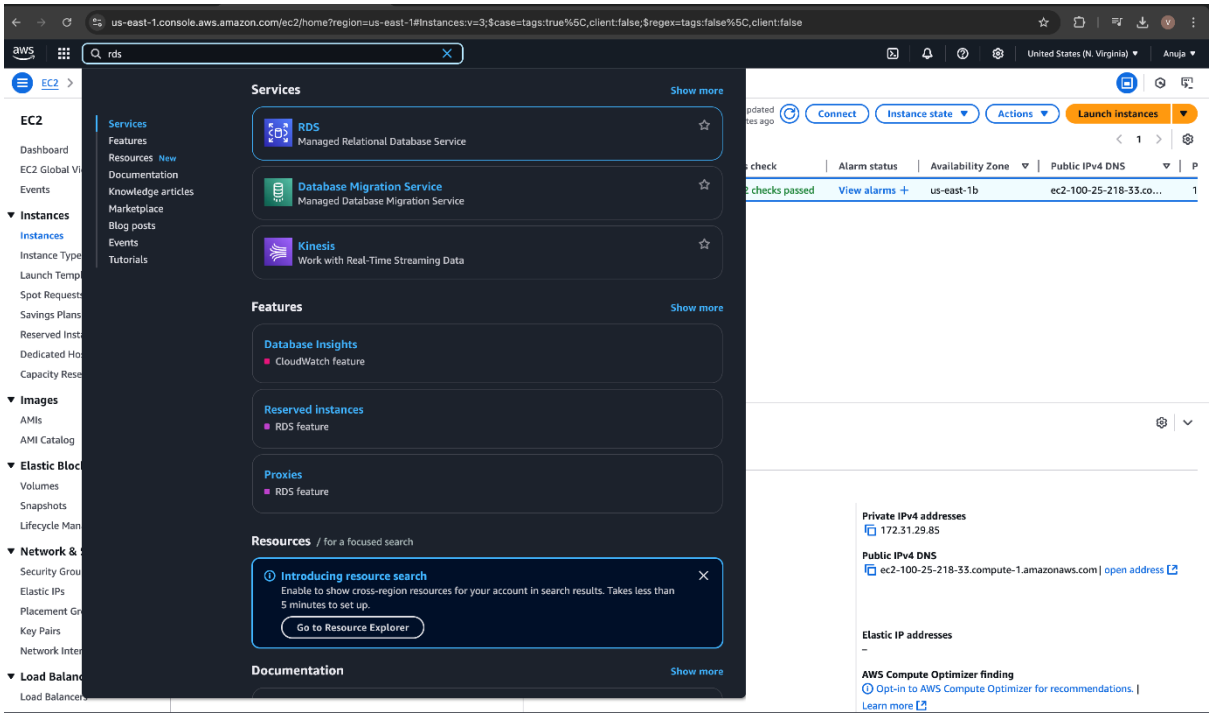
unzip the tar file using the following command.

tar -xzf latest.tar.gz after installation you can find the wordpress file.

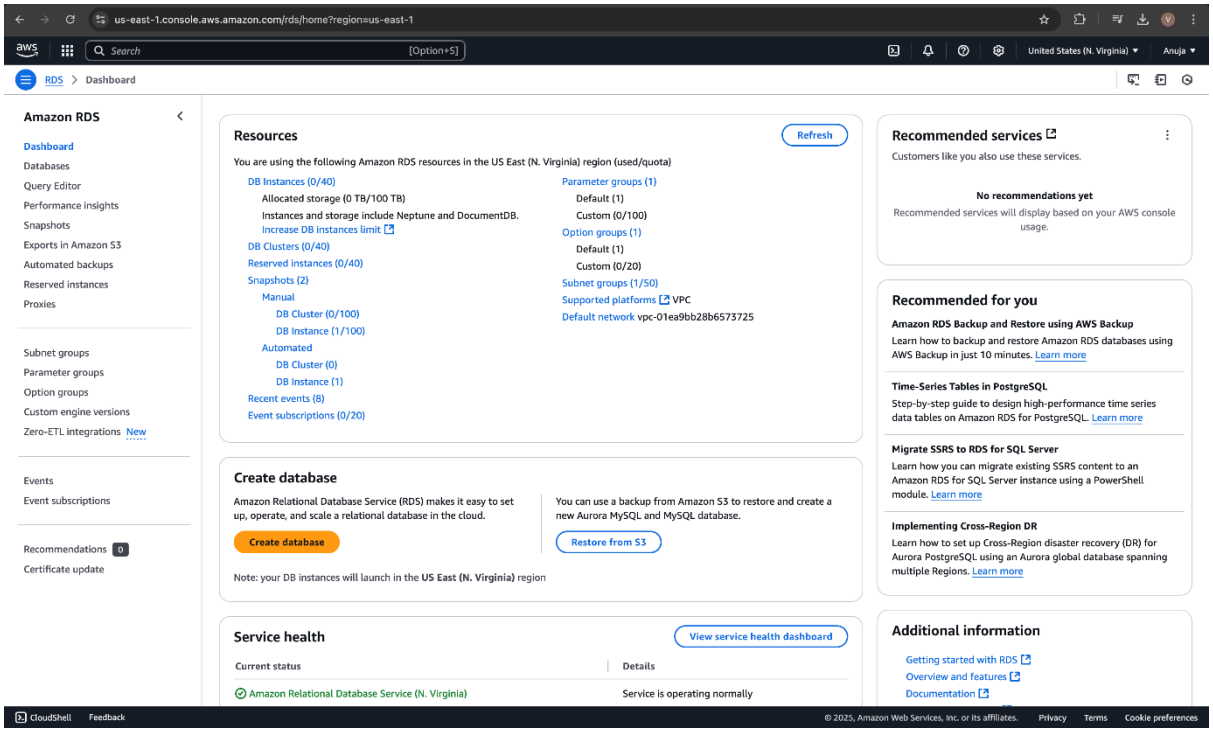
```
[ec2-user@ip-172-31-29-85 ~]$ ls
latest.tar.gz
[ec2-user@ip-172-31-29-85 ~]$ tar -xzf latest.tar.gz
[ec2-user@ip-172-31-29-85 ~]$ ls
latest.tar.gz  wordpress
[ec2-user@ip-172-31-29-85 ~]$
```

Till now we installed all servers like php, mariadb and wordpress for our website now we will create a database to connect our server.

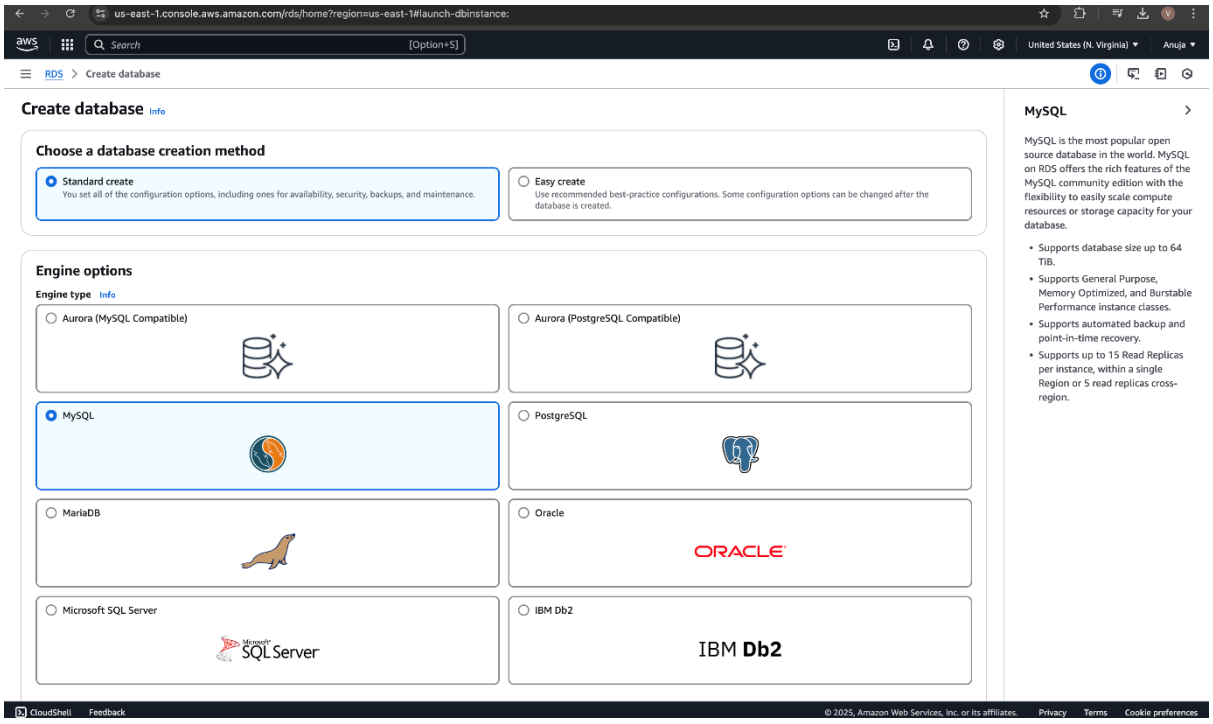
Go to aws console and search for rds.



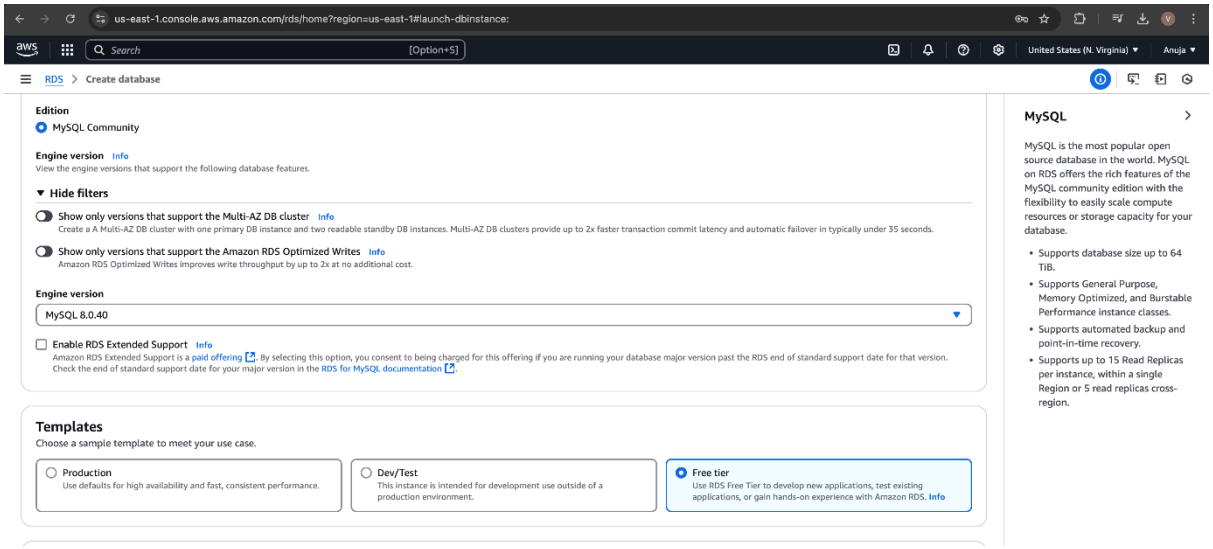
click on create database.



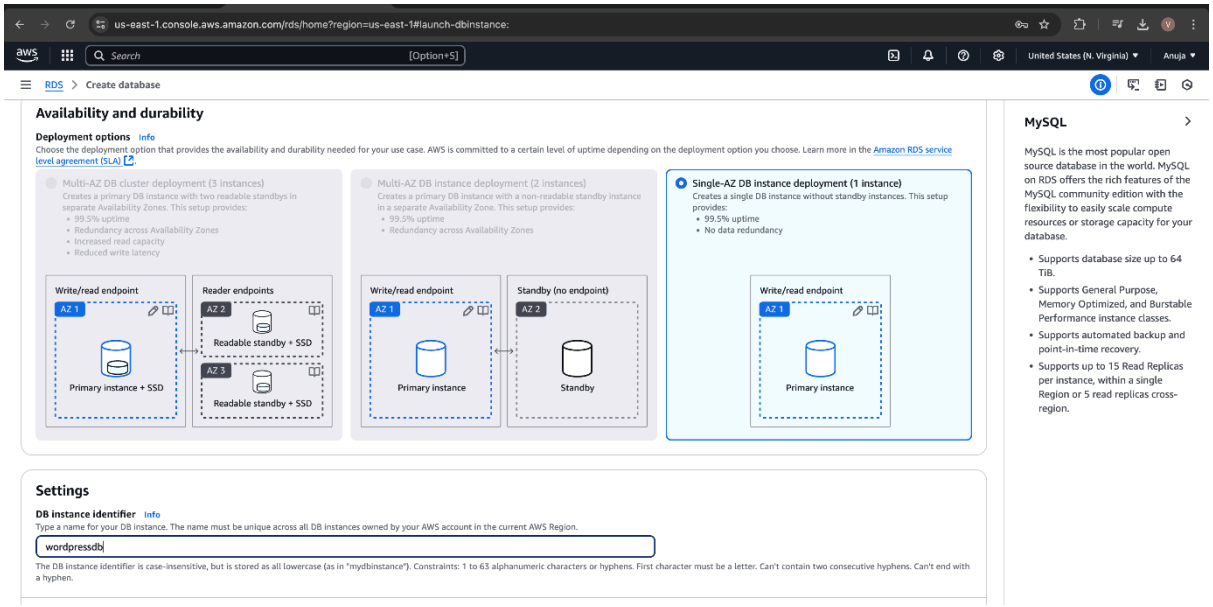
choose database creation method as standard create and engine option as MYSQL.



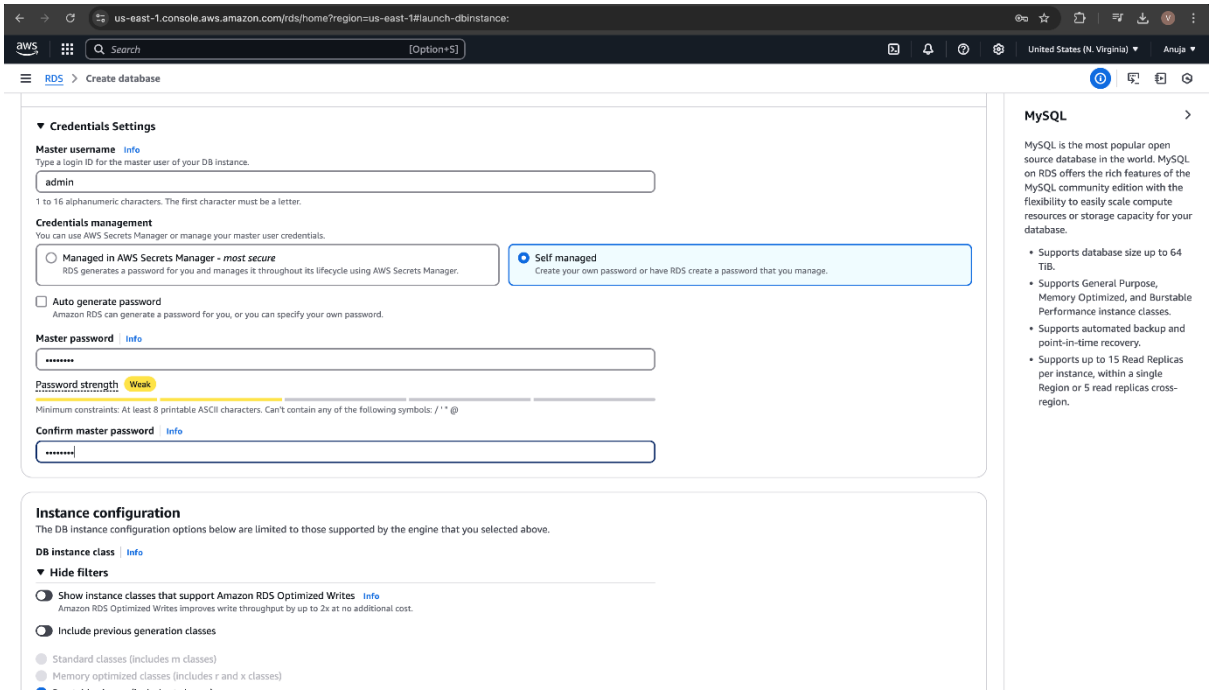
Select engine version as MySQL 8.0.40 and template as free tier.



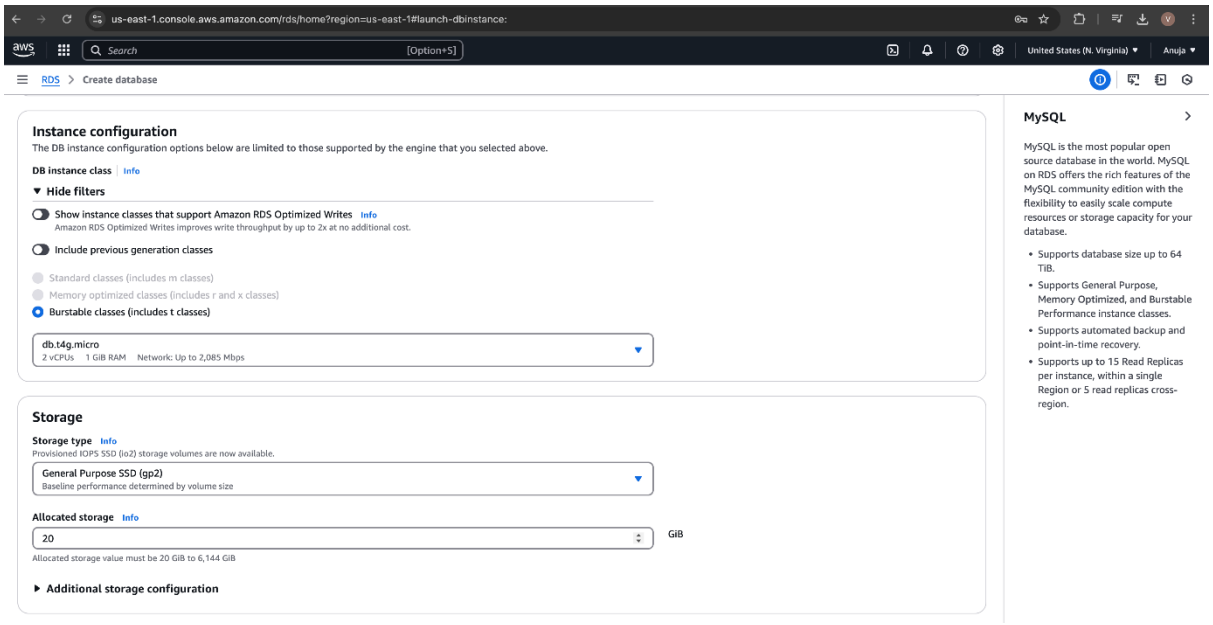
Availability and durability as single AZ-db instance give your database a name as wordpressdb.



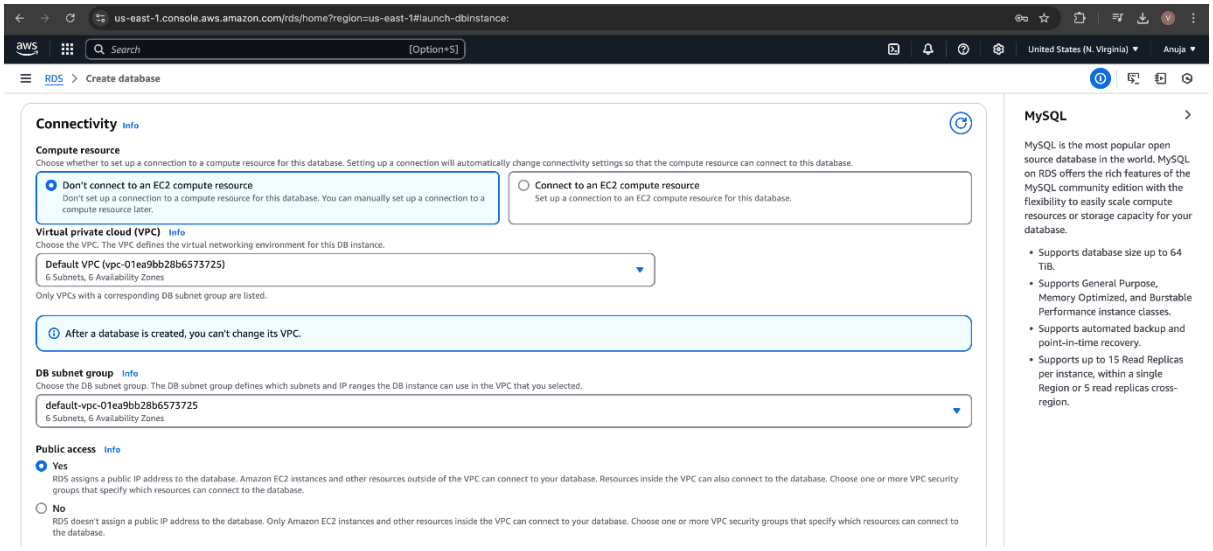
set credentials and give a password for your database make sure to remember your credentials.



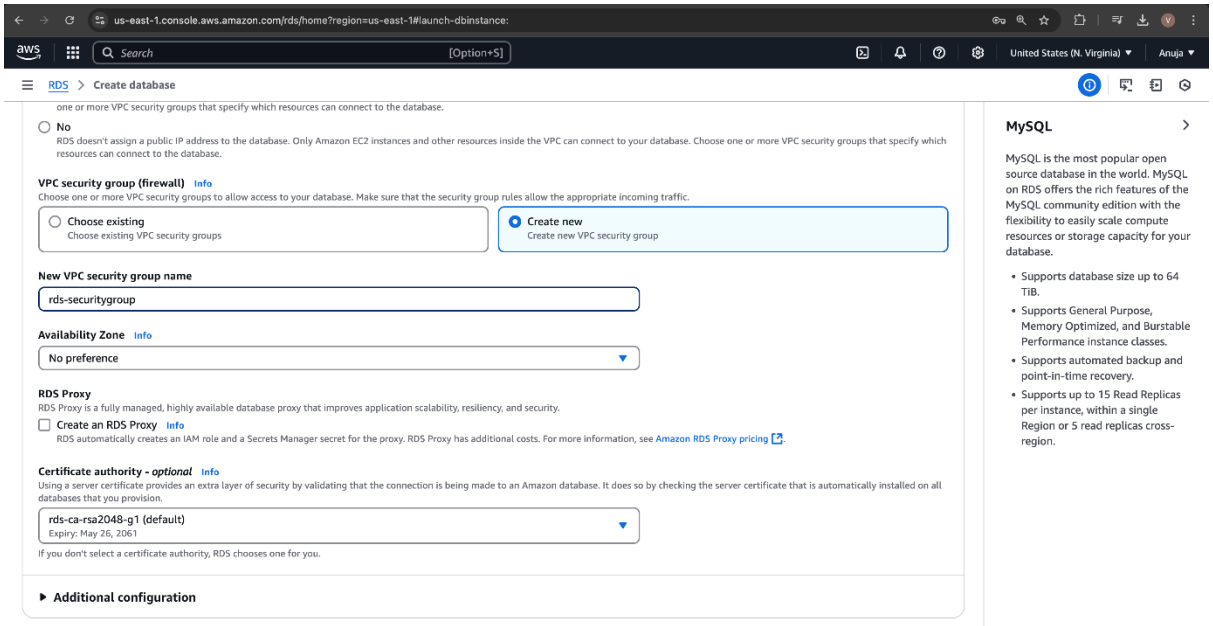
select instance configuration as burstable classes t4g.micro and storage as general purpose SSD (gp2)



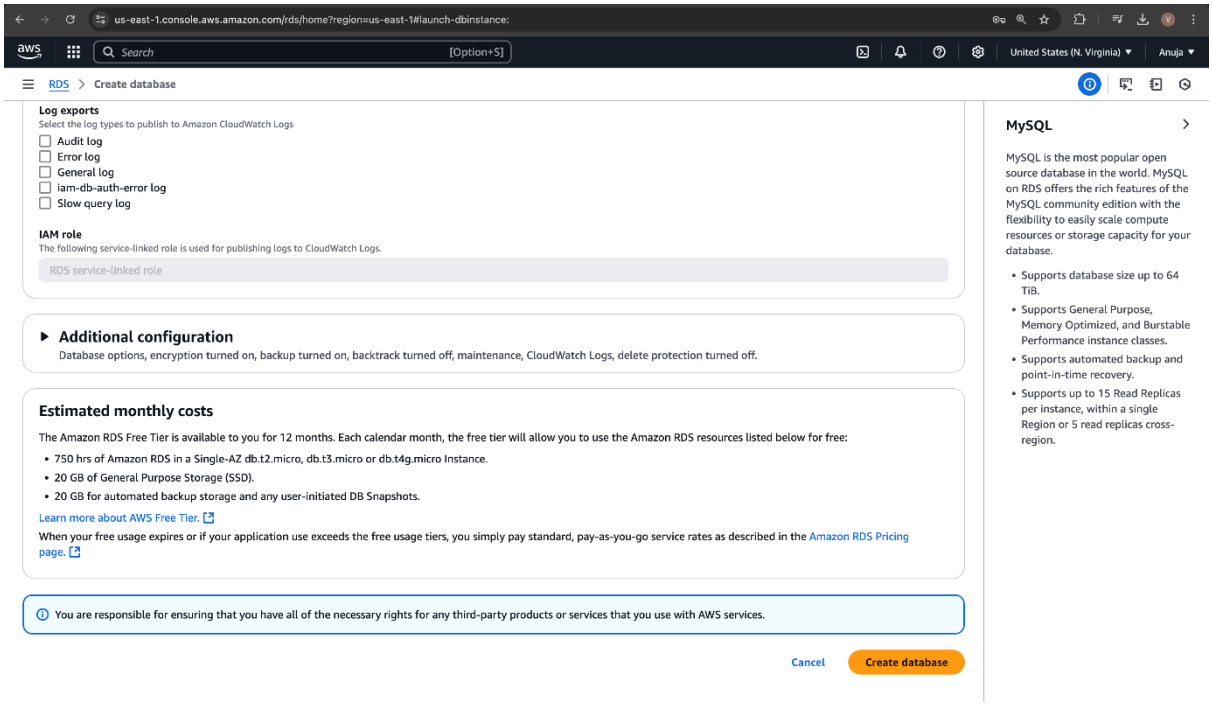
select connectivity as don’t connect to ec2 instance. And select VPC and give a subnet group. You need to give public access as YES to accessible over internet.



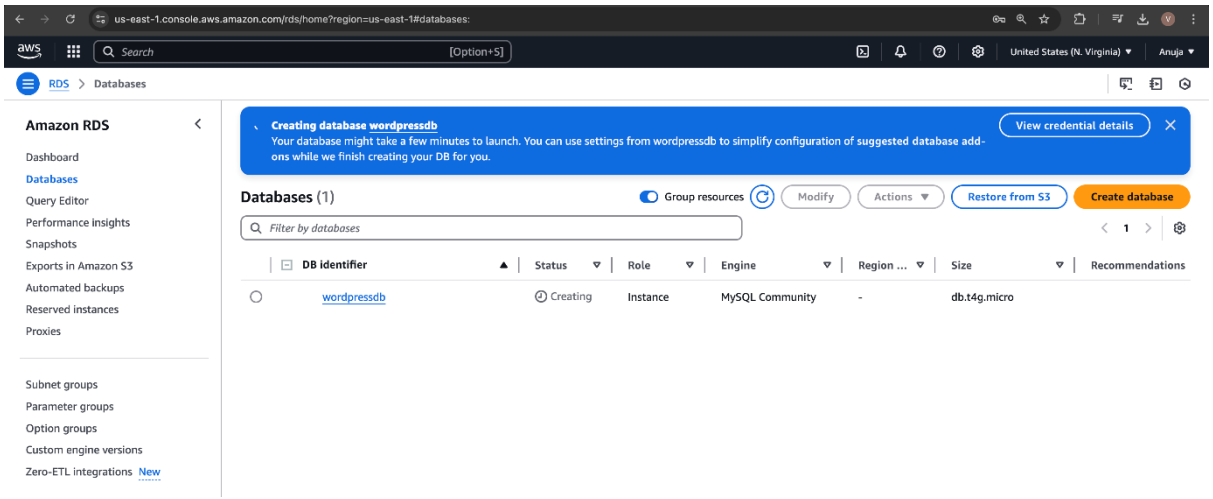
create a security group name as rds-securitygroup.



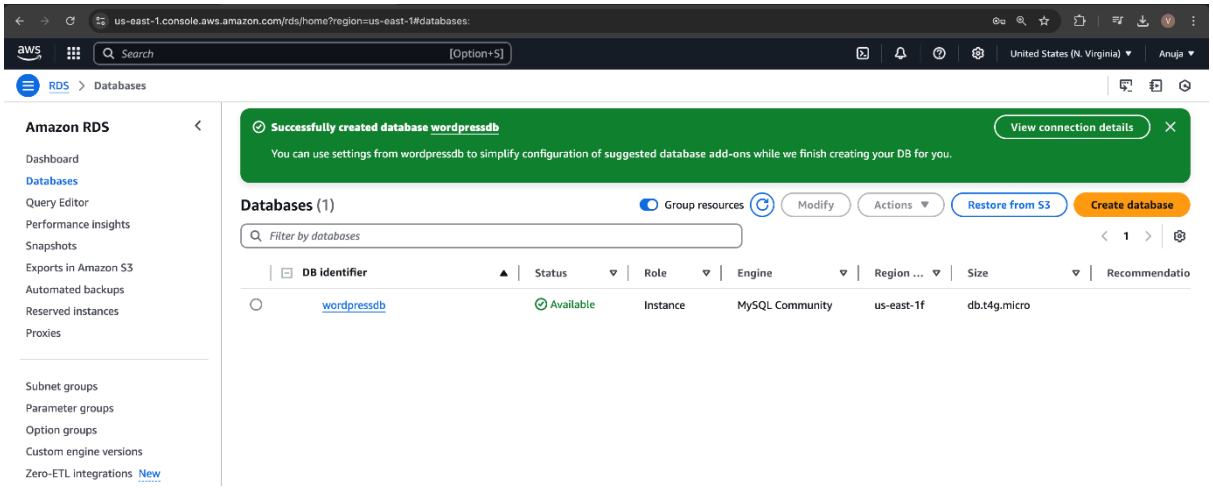
You can leave remaining setting as it is and click on create database.



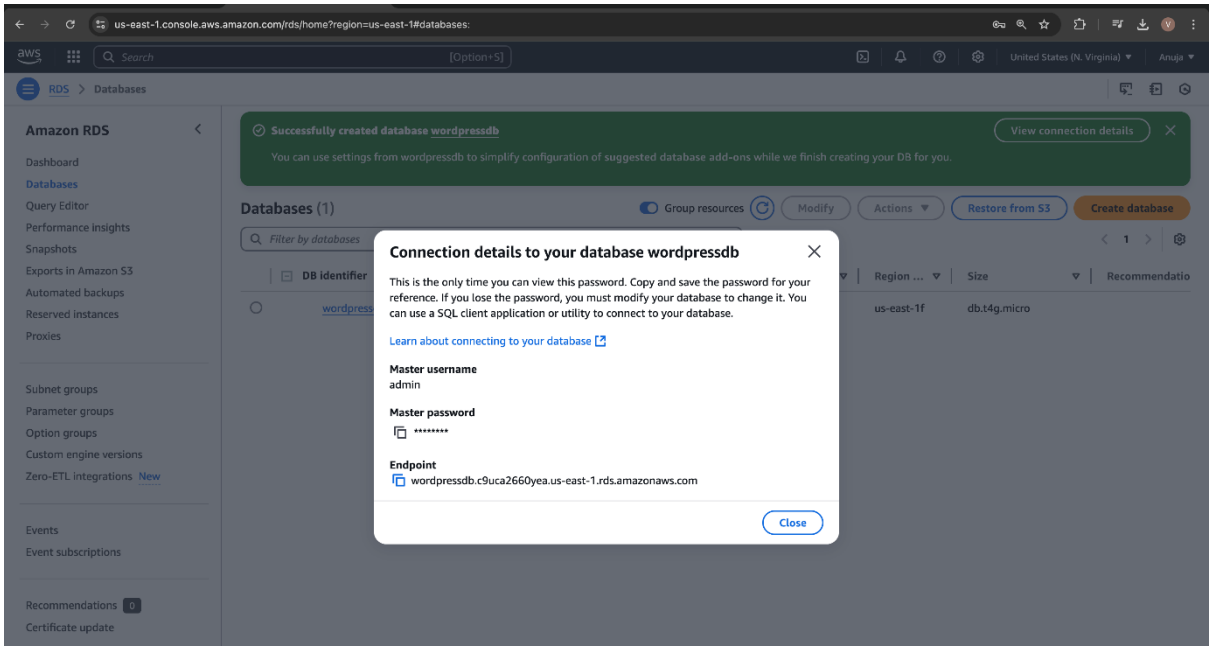
You can see the wordpressdb is in creating status it will take 5 to 10 mins to get ready our db instance.



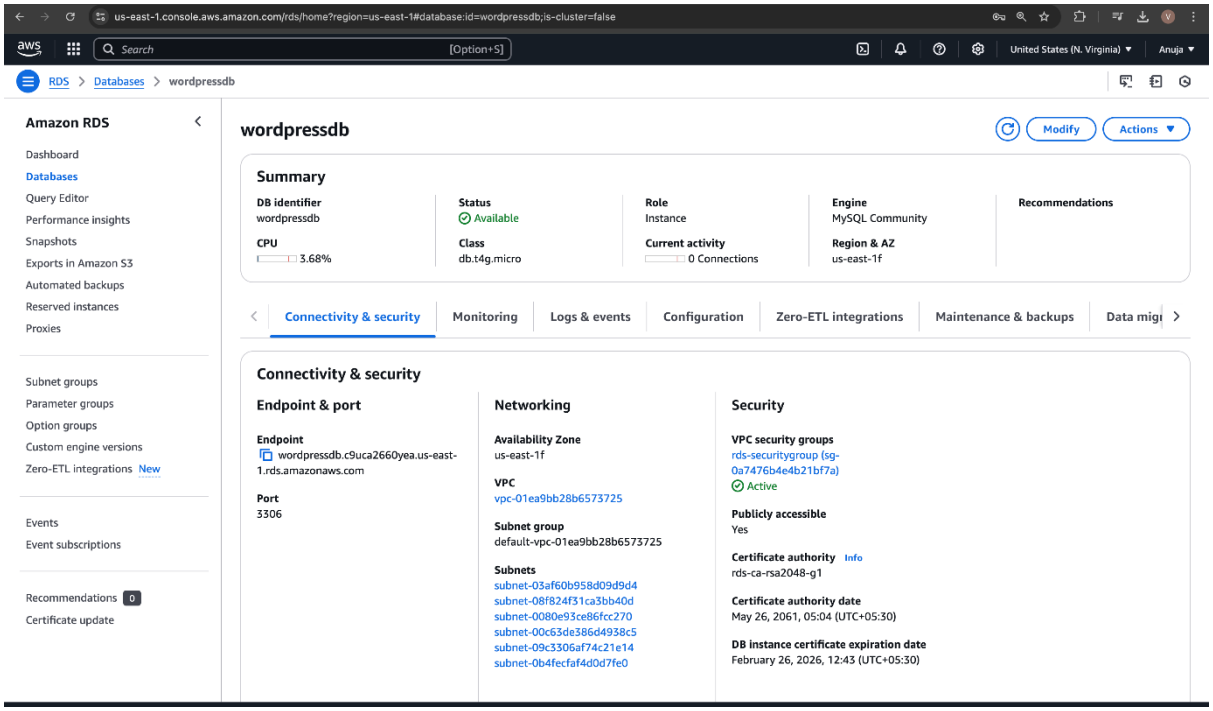
once your database is created you can see the below status. It will show database status as avaliable.



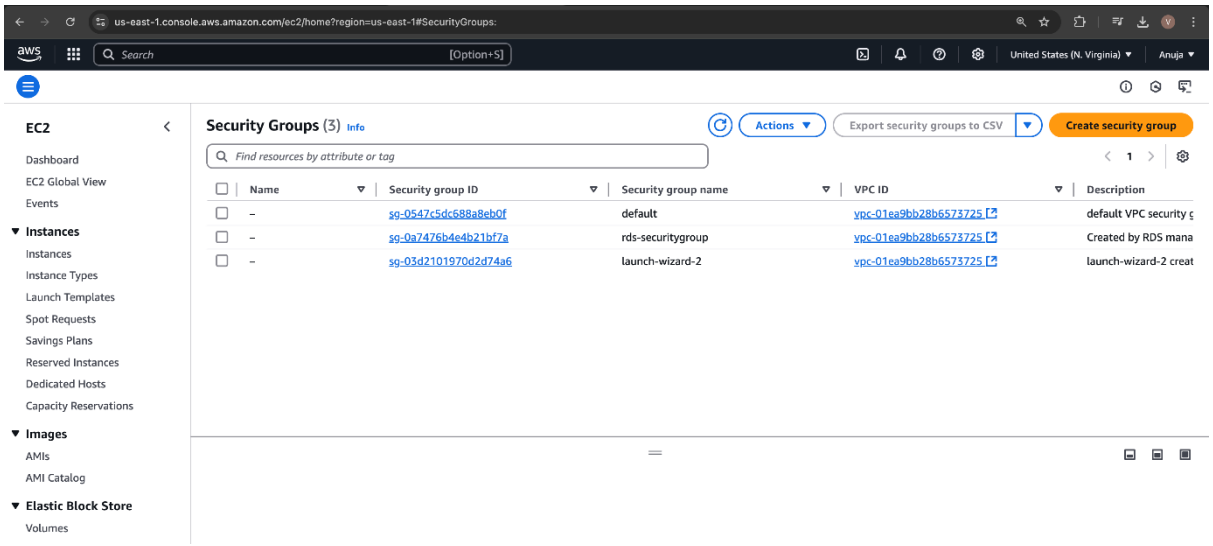
On top of this page, you can see in green colour view connection details. If you press that one you will find your database credentials make sure to note them.

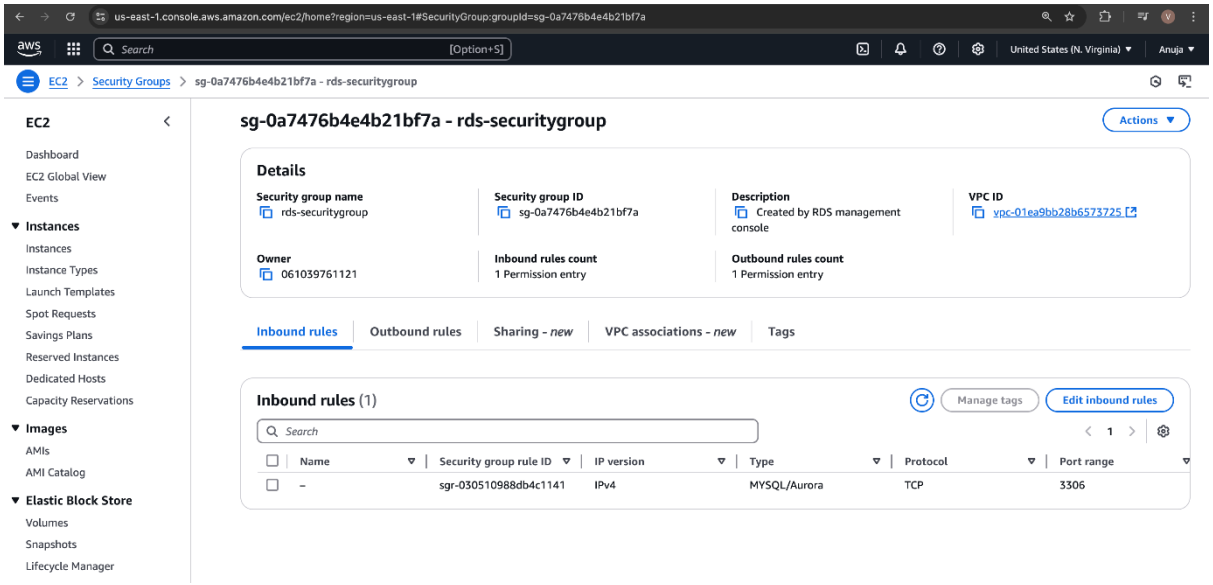


After that go to databases then you press on our database then you will find the end point to our database this will be most important for our website.

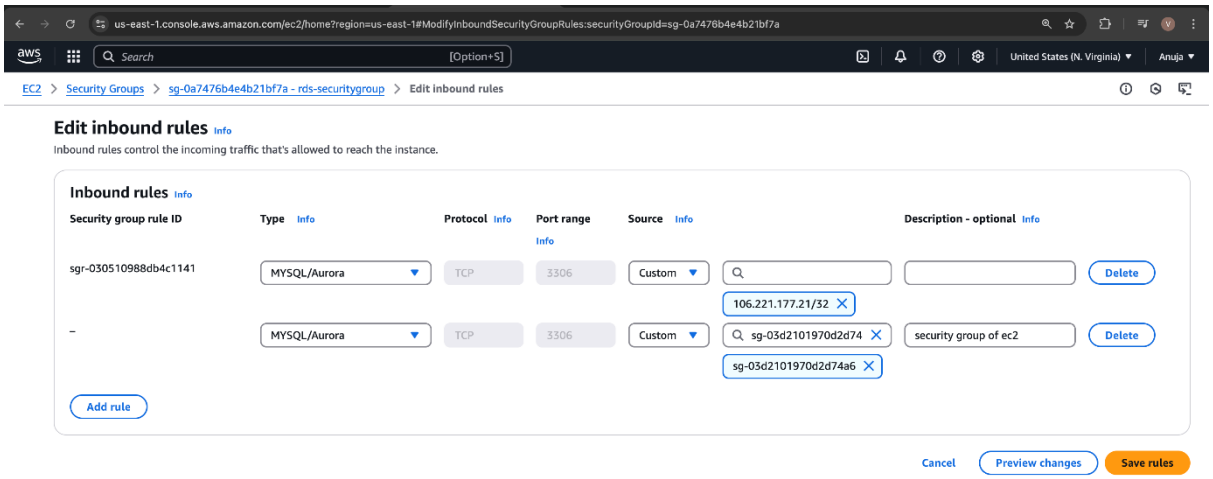


Go to security group of your db instance and add the inbound rules.
Copy your ec2 security group.

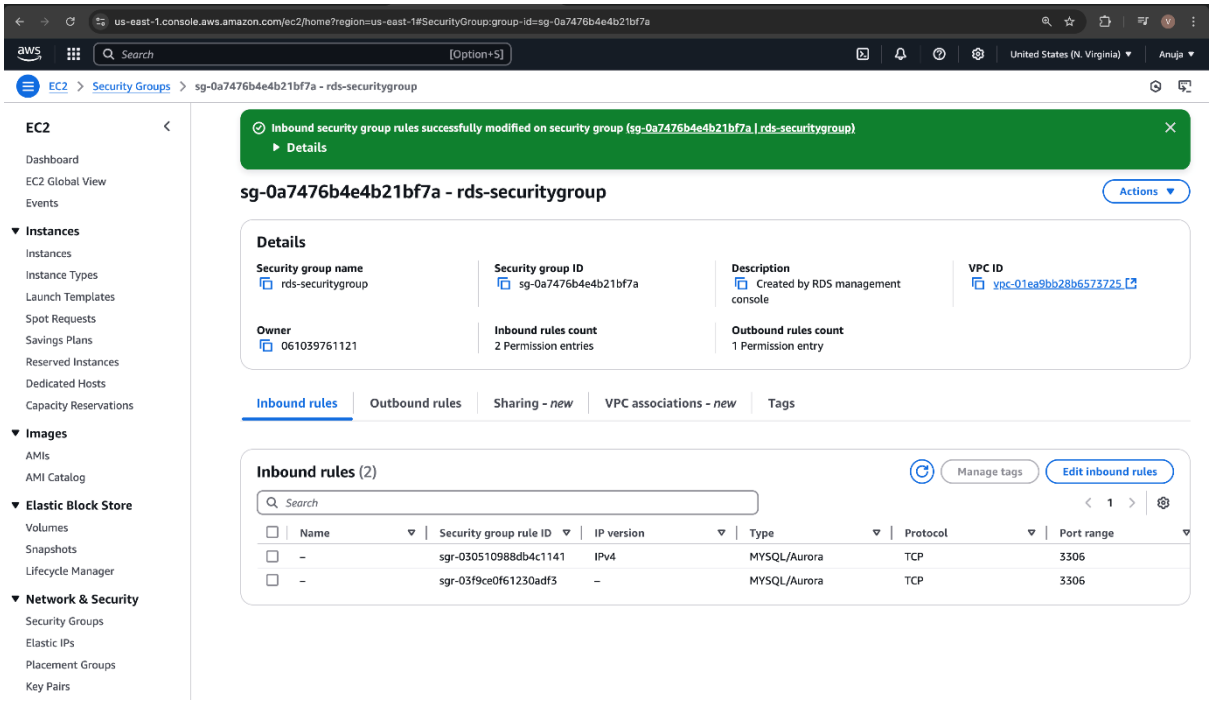




Then in inbound rule add your rule MySql/Aurora and in source add your security group. And click on save rules.

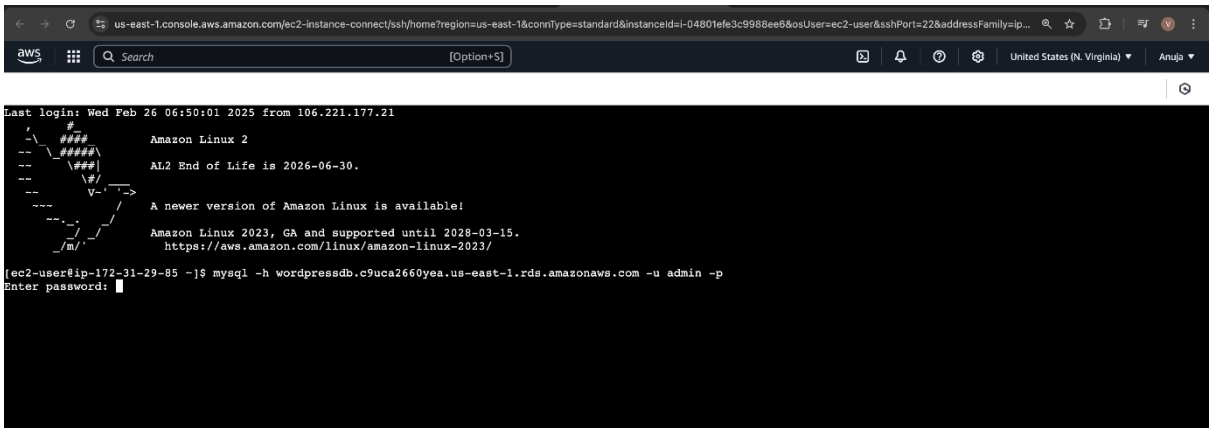


And you can see your Ec2 security group is added in the database security group.

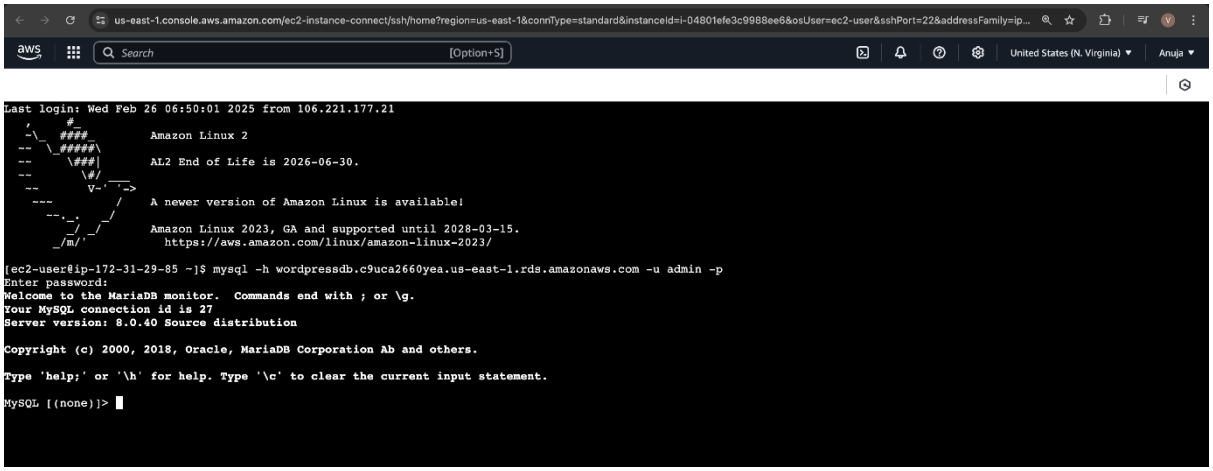


To check your database you can connect to ec2 instance connect and type this command.

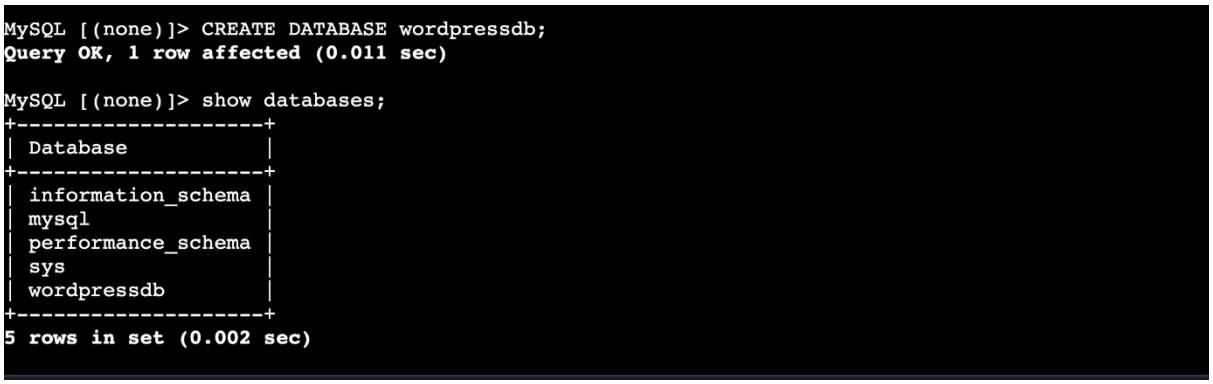
Mysql -h Your database end point -u admin -p and enter your database password.



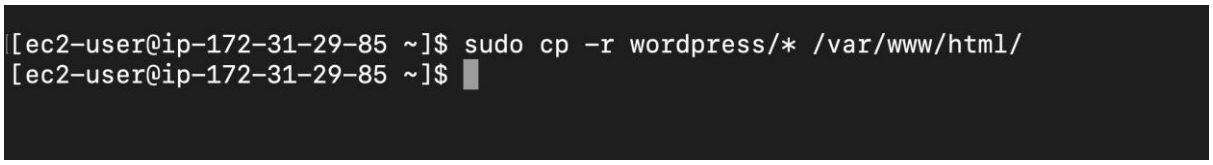
Finally you can see database is added.



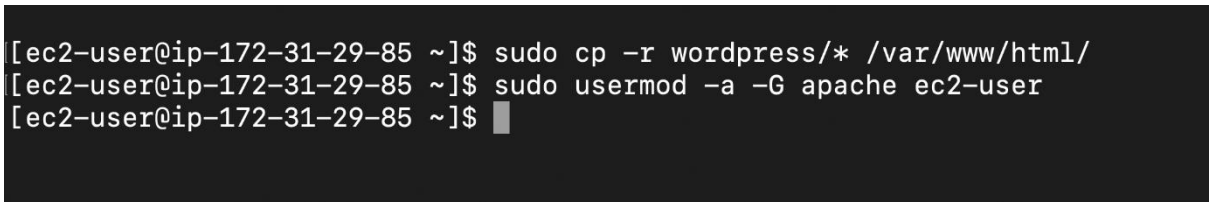
And then press show databases; command you can see your word press db instance.



Now copy your wordpress folder to var/www/html/ using the following command.
sudo cp -r wordpress/* /var/www/html/



Now Adding your current user to Apache group.



Then you need to exit from the current Ec2 instance and login back again.

```
[ec2-user@ip-172-31-29-85 ~]$ sudo cp -r wordpress/* /var/www/html/
[ec2-user@ip-172-31-29-85 ~]$ sudo usermod -a -G apache ec2-user
[ec2-user@ip-172-31-29-85 ~]$ exit
logout
Connection to ec2-100-25-218-33.compute-1.amazonaws.com closed.
raj@Rajs-Laptop desktop %
```

Now we need to change permission on var/www folder using these 3 commands.

sudo chown -R ec2-user:apache /var/www

sudo chmod 2775 /var/www && find /var/www -type d -exec sudo chmod 2775 {} \;

find /var/www -type f -exec sudo chmod 0664 {} \;

and go to that cd /var/www/html and do ls we will find all the required files.

```
[ec2-user@ip-172-31-29-85 ~]$ sudo chown -R ec2-user:apache /var/www
[ec2-user@ip-172-31-29-85 ~]$ sudo chmod 2775 /var/www && find /var/www -type d -exec sudo chmod 2775 {} \;
[ec2-user@ip-172-31-29-85 ~]$ find /var/www -type f -exec sudo chmod 0664 {} \;
[ec2-user@ip-172-31-29-85 ~]$ cd /var/www/html
[ec2-user@ip-172-31-29-85 html]$ ls
index.php    wp-activate.php     wp-comments-post.php  wp-cron.php        wp-load.php   wp-settings.php   xmlrpc.php
license.txt  wp-admin            wp-config-sample.php  wp-includes        wp-login.php  wp-signup.php
readme.html  wp-blog-header.php  wp-content            wp-links-opml.php  wp-mail.php   wp-trackback.php
[ec2-user@ip-172-31-29-85 html]$
```

Now we need to copy the config file to another file. Using this command.

cp wp-config-sample.php wp-config.php

```
[ec2-user@ip-172-31-29-85 ~]$ sudo chown -R ec2-user:apache /var/www
[ec2-user@ip-172-31-29-85 ~]$ sudo chmod 2775 /var/www && find /var/www -type d -exec sudo chmod 2775 {} \;
[ec2-user@ip-172-31-29-85 ~]$ find /var/www -type f -exec sudo chmod 0664 {} \;
[ec2-user@ip-172-31-29-85 ~]$ cd /var/www/html
[ec2-user@ip-172-31-29-85 html]$ ls
index.php    wp-activate.php     wp-comments-post.php  wp-cron.php        wp-load.php   wp-settings.php   xmlrpc.php
license.txt  wp-admin            wp-config-sample.php  wp-includes        wp-login.php  wp-signup.php
readme.html  wp-blog-header.php  wp-content            wp-links-opml.php  wp-mail.php   wp-trackback.php
[ec2-user@ip-172-31-29-85 html]$ cp wp-config-sample.php wp-config.php
[ec2-user@ip-172-31-29-85 html]$ ls
index.php    wp-activate.php     wp-comments-post.php  wp-content   wp-links-opml.php  wp-mail.php      wp-trackback.php
license.txt  wp-admin            wp-config.php         wp-cron.php  wp-load.php        wp-settings.php  xmlrpc.php
readme.html  wp-blog-header.php  wp-config-sample.php  wp-includes  wp-login.php       wp-signup.php
[ec2-user@ip-172-31-29-85 html]$
```

And now we need to open config file using Nano editor.

nano wp-config.php

```
[ec2-user@ip-172-31-29-85 html]$ nano wp-config.php
```

Here we need to add all our database details like user name, password, our db end point.

```
  GNU nano 2.9.8                                wp-config.php

<?php
/**
 * The base configuration for WordPress
 *
 * The wp-config.php creation script uses this file during the installation.
 * You don't have to use the website, you can copy this file to "wp-config.php"
 * and fill in the values.
 *
 * This file contains the following configurations:
 *
 * * Database settings
 * * Secret keys
 * * Database table prefix
 * * ABSPATH
 *
 * @link https://developer.wordpress.org/advanced-administration/wordpress/wp-config/
 *
 * @package WordPress
 */

// ** Database settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define( 'DB_NAME', 'database_name_here' );

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

/** Database password */
define( 'DB_PASSWORD', 'password_here' );

/** Database hostname */
define( 'DB_HOST', 'localhost' );

                [ Read 102 lines (Converted from DOS format) ]
^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text  ^J Justify  ^C Cur Pos  M-U Undo  M-A Mark Text
^X Exit      ^R Read File  ^\ Replace   ^U Uncut Text ^T To Spell ^_ Go To Line M-E Redo  M-6 Copy Text
```

After editing we can see the file like this press cntl+s to save and cntl+x to exit.

```
// ** Database settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define( 'DB_NAME', 'wordpressdb' );

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

/** Database password */
define( 'DB_PASSWORD', 'admin123' );

/** Database hostname */
define( 'DB_HOST', 'wordpressdb.c9uca2660yea.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.
```

After that open config file using vim editor.

sudo vim /etc/httpd/conf/httpd.conf

```
[ec2-user@ip-172-31-29-85 html]$ sudo vim /etc/httpd/conf/httpd.conf
```

after that file looks like this.

```
#
# DocumentRoot: The directory out of which you will serve your
# documents. By default, all requests are taken from this directory, but
# symbolic links and aliases may be used to point to other locations.
#
DocumentRoot "/var/www/html"

#
# Relax access to content within /var/www.
#
<Directory "/var/www">
    AllowOverride None
    # Allow open access:
    Require all granted
</Directory>

# Further relax access to the default document root:
<Directory "/var/www/html">
    #
    # Possible values for the Options directive are "None", "All",
    # or any combination of:
    #   Indexes Includes FollowSymLinks SymLinksifOwnerMatch ExecCGI MultiViews
    #
    # Note that "MultiViews" must be named *explicitly* --- "Options All"
    # doesn't give it to you.
    #
    # The Options directive is both complicated and important.  Please see
    # http://httpd.apache.org/docs/2.4/mod/core.html#options
    # for more information.
    #
```

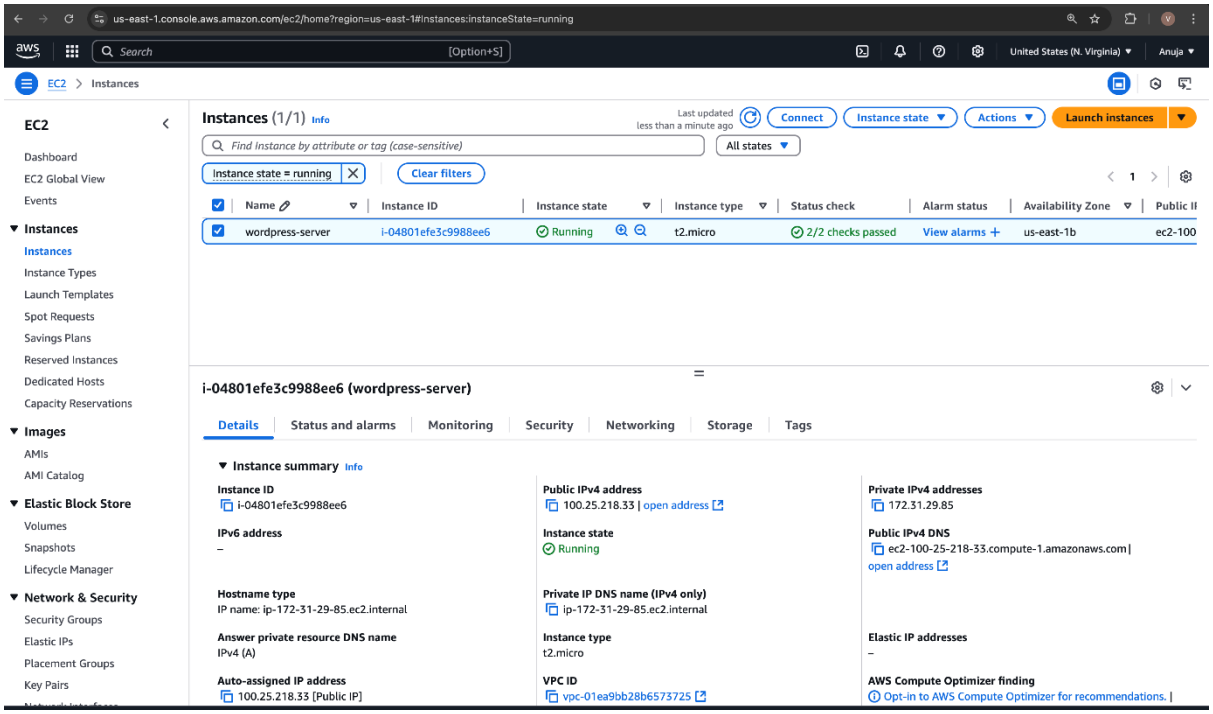
Now we need to change AllowOverride none to All

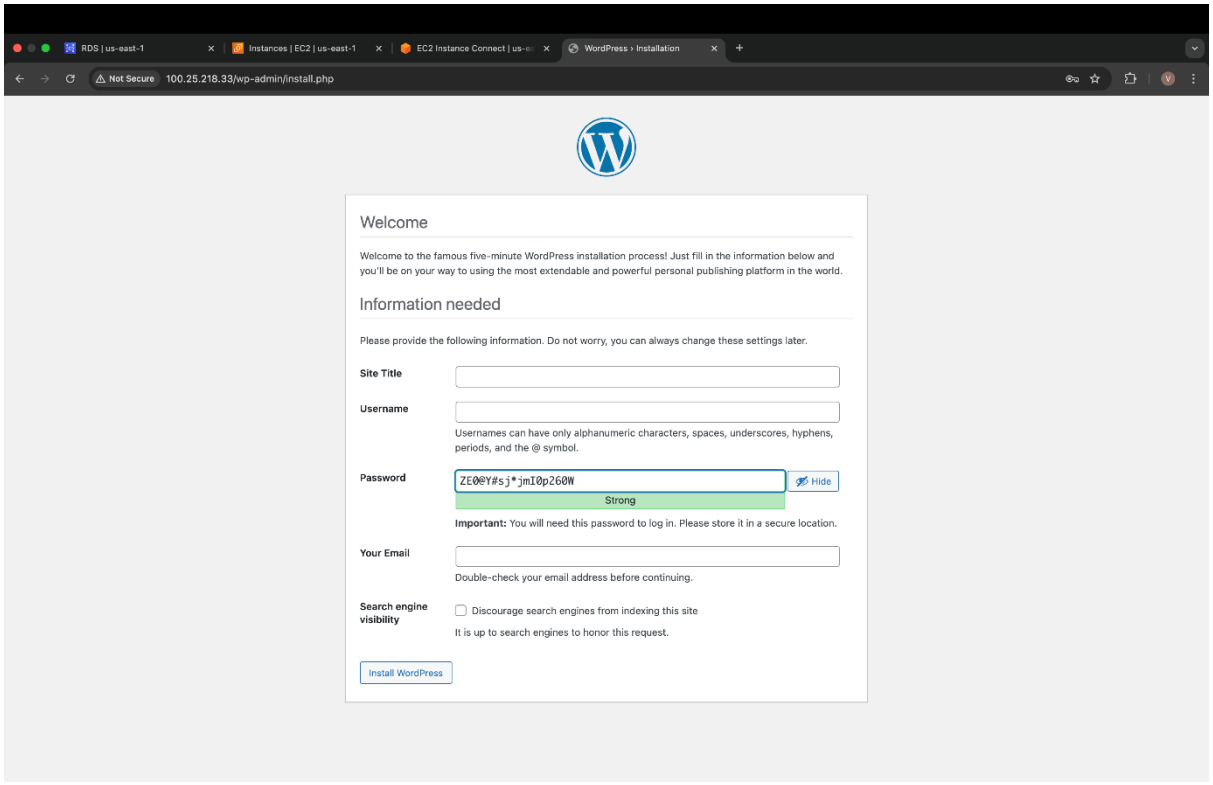
```
#
# DocumentRoot: The directory out of which you will serve your
# documents. By default, all requests are taken from this directory, but
# symbolic links and aliases may be used to point to other locations.
#
DocumentRoot "/var/www/html"

#
# Relax access to content within /var/www.
#
<Directory "/var/www">
    AllowOverride All
    # Allow open access:
    Require all granted
</Directory>

# Further relax access to the default document root:
<Directory "/var/www/html">
    #
    # Possible values for the Options directive are "None", "All",
    # or any combination of:
    #   Indexes Includes FollowSymLinks SymLinksifOwnerMatch ExecCGI MultiViews
    #
    # Note that "MultiViews" must be named *explicitly* --- "Options All"
    # doesn't give it to you.
    #
    # The Options directive is both complicated and important.  Please see
    # http://httpd.apache.org/docs/2.4/mod/core.html#options
    # for more information.
    #
    Options Indexes FollowSymLinks
```

Now we need to restart the httpd server.

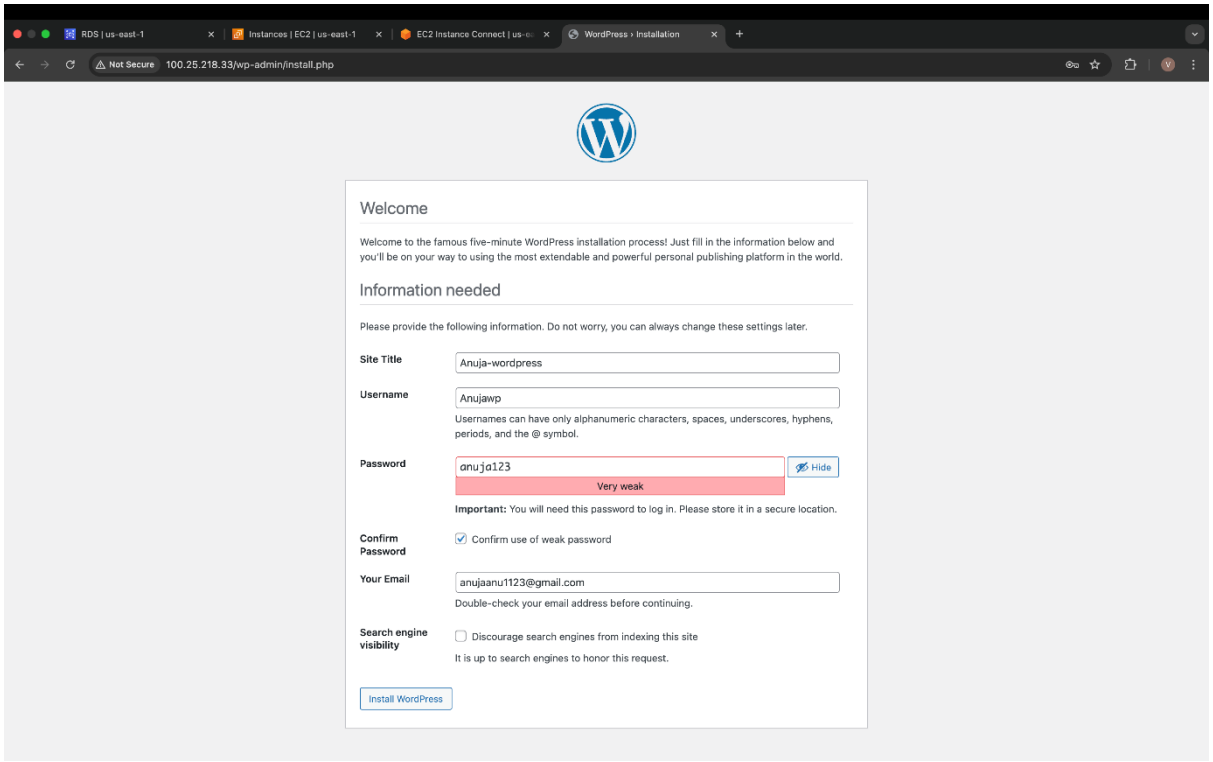
```
[ec2-user@ip-172-31-29-85 html]$ sudo systemctl restart httpd
[ec2-user@ip-172-31-29-85 html]$
```

Finally you are ready to open your instance. In a browser open your public ip. Then you will find the wordpress website opened.

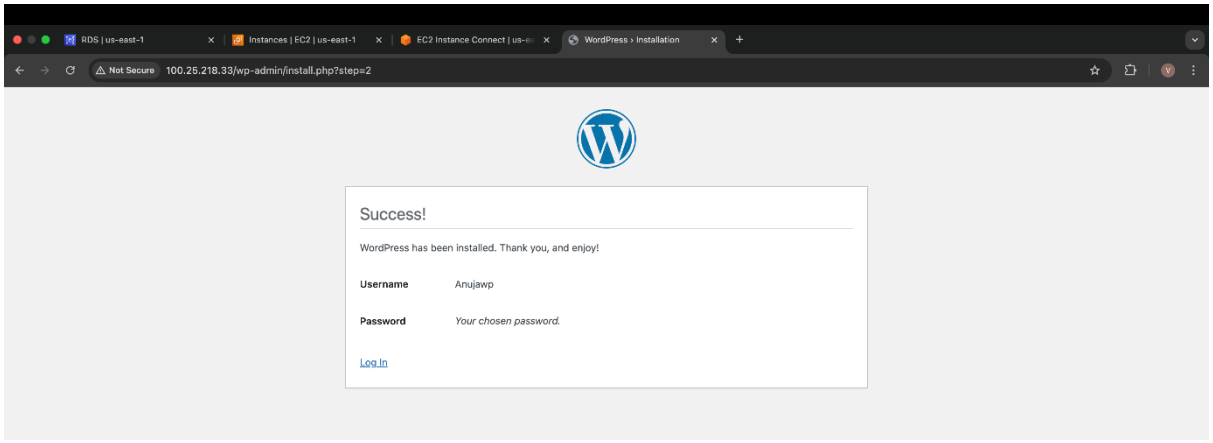




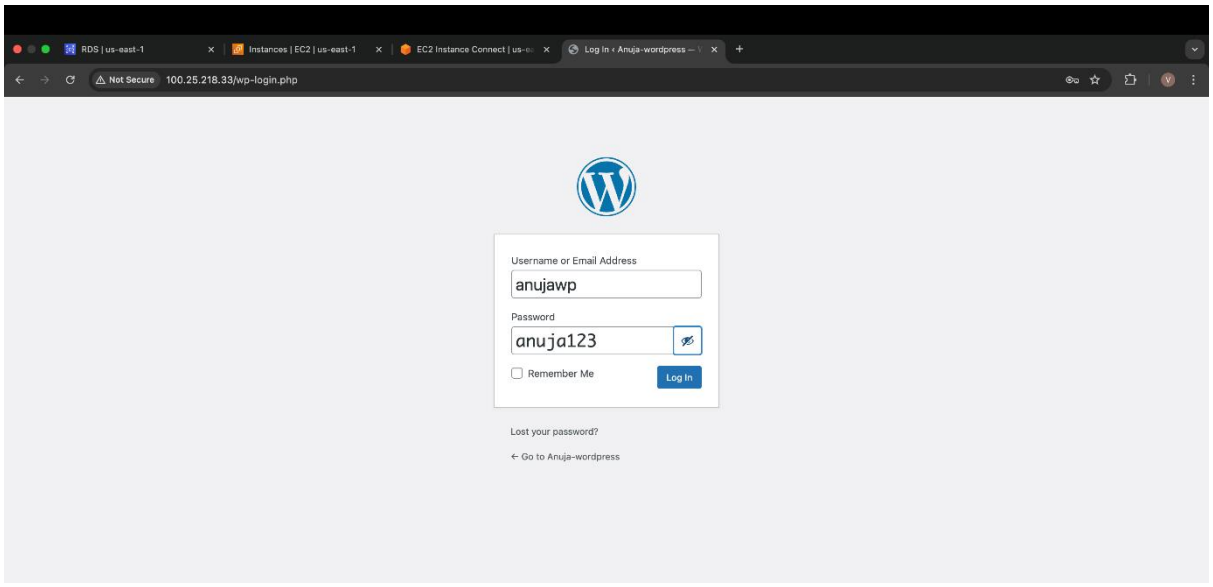
Fill this form and then press install wordpress.



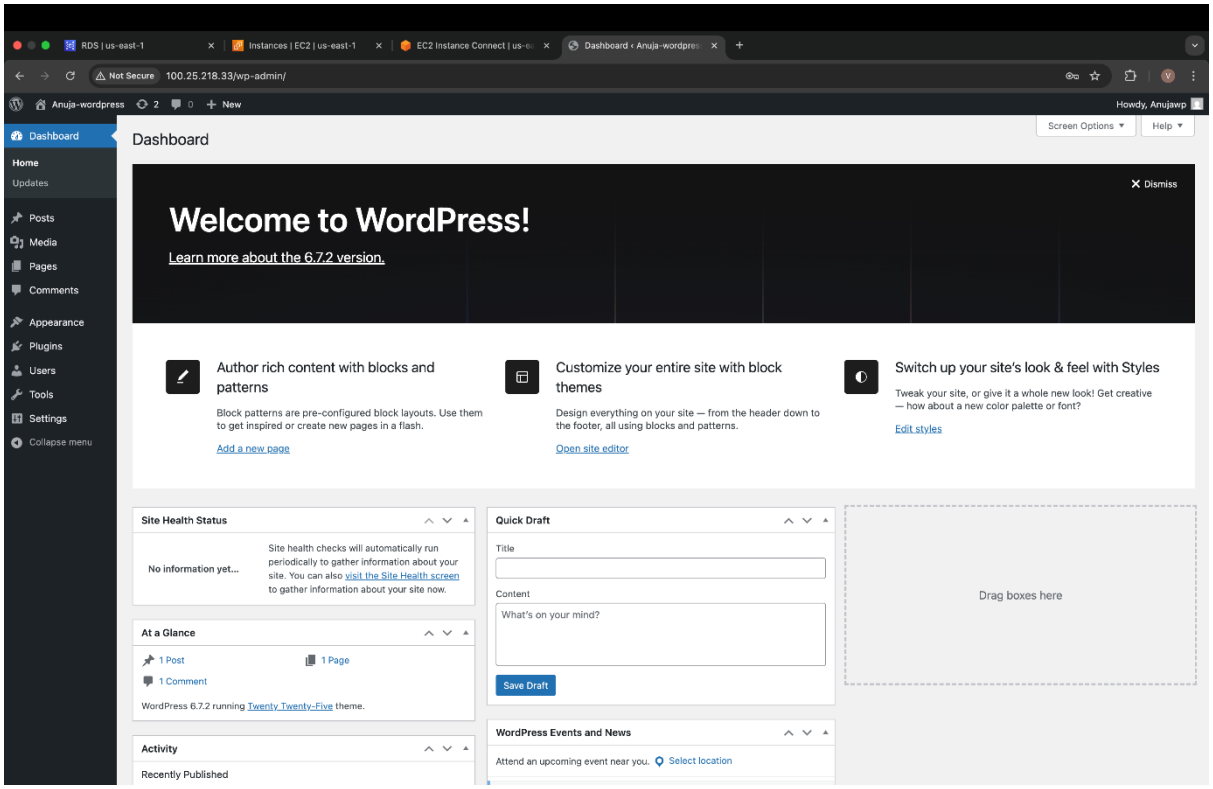
After that you will find success page then login to word press instance.

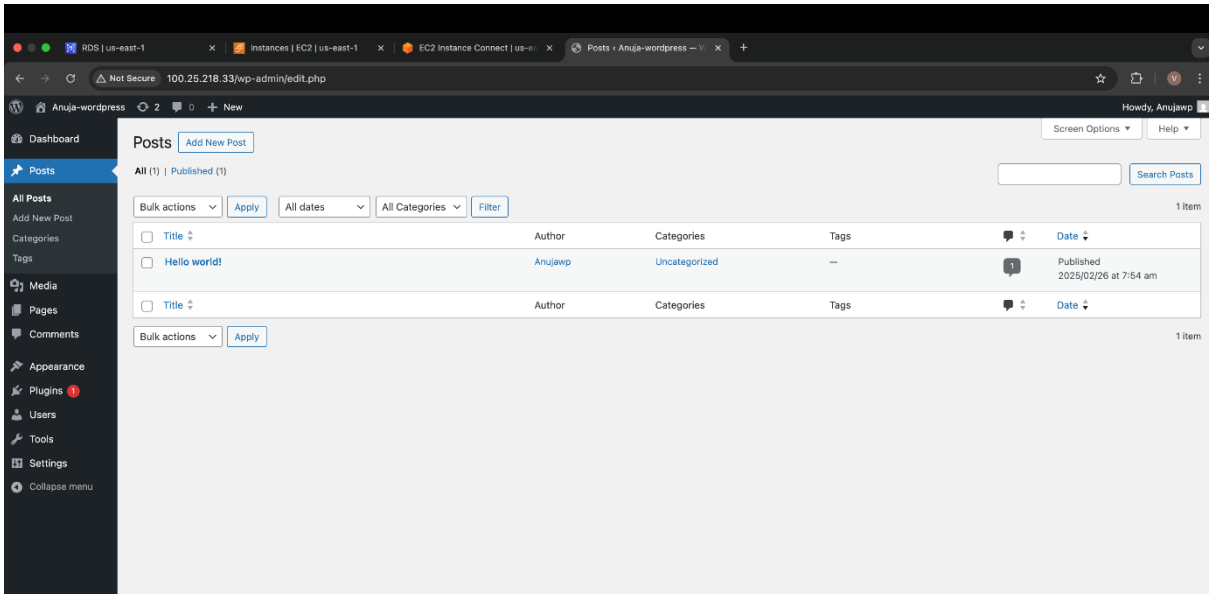


Use your username and password to login to wordpress site.



Finally your word press website is ready.


RDS | us-east-1
Instances | EC2 | us-east-1
EC2 Instance Connect | us-e
Dashboard ‹ Anuja-wordpres
Not Secure
100.25.218.33/wp-admin/
Anuja-wordpress
2
0
New
Howdy, Anujawp
Screen Options
Help
Dashboard
Home
Updates
Posts
Media
Pages
Comments
Appearance
Plugins
Users
Tools
Settings
Collapse menu
Dashboard
Dismiss
Welcome to WordPress!
Learn more about the 6.7.2 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 about your site now.
At a Glance
1 Post
1 Page
1 Comment
WordPress 6.7.2 running Twenty Twenty-Five theme.
Activity
Recently Published
Quick Draft
Title
Content
What's on your mind?
Save Draft
WordPress Events and News
Attend an upcoming event near you.
Select location
Drag boxes here


RDS | us-east-1
Instances | EC2 | us-east-1
EC2 Instance Connect | us-e
Posts ‹ Anuja-wordpress
Not Secure
100.25.218.33/wp-admin/edit.php
Anuja-wordpress
2
0
New
Howdy, Anujawp
Screen Options
Help
Dashboard
Posts
All Posts
Add New Post
Categories
Tags
Media
Pages
Comments
Appearance
Plugins
Users
Tools
Settings
Collapse menu
Posts
Add New Post
All (1) | Published (1)
Search Posts
Bulk actions
Apply
All dates
All Categories
Filter
1 item
Title
Author
Categories
Tags
Date
Hello world!
Anujawp
Uncategorized
—
Published
2025/02/26 at 7:54 am
Title
Author
Categories
Tags
Date
Bulk actions
Apply
1 item