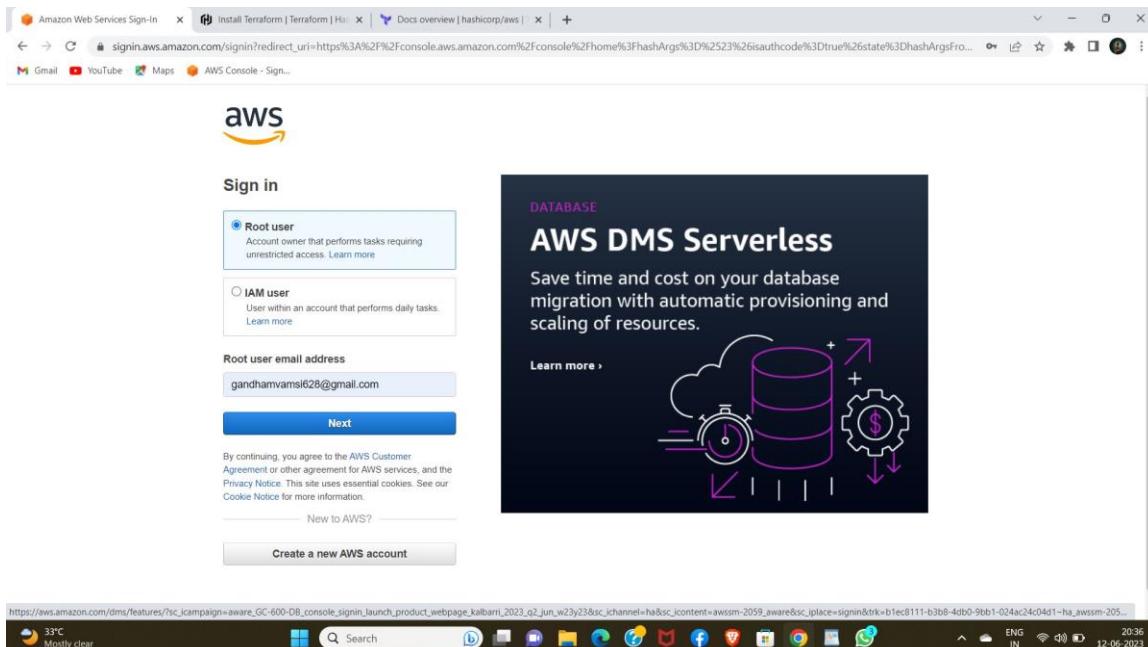


# PROJECT – 2

## Deploy wordpress application with amazon web services

- What is wordpress.
- ✓ .At its core, WordPress is the simplest, most popular way to create your own website or blog. In fact, WordPress powers over 43.3% of all the websites on the Internet. Yes – more than one in four websites that you visit are likely powered by WordPress.

### 1. goto console log into root user.



## 2.create IAM user log into IAM user.

The screenshot shows the AWS IAM Management Console. On the left, there's a sidebar with navigation links like Identity and Access Management (IAM), Dashboard, Access management, and Access reports. The main area is the IAM dashboard, which includes sections for Security recommendations, IAM resources, and AWS Account details. A tooltip indicates that the Account ID has been copied.

The screenshot shows the AWS Sign-in page. It features a 'Sign in as IAM user' form with fields for Account ID (193633228050), IAM user name (gandhamvamsi628@gmail.com), and Password. Below the form are links for 'Sign in using root user email' and 'Forgot password?'. To the right, there's a promotional banner for 'Boost price performance' with a gear icon containing a dollar sign.

## 3.create a vpc in cidr "10.0.0.0/20"

**VPC Details:**

Setting	Value
VPC ID	vpc-092b5fb14b0743f5a
State	Available
Tenancy	Default
DHCP option set	dopt-0adcb0506889d2f
Default VPC	No
IPv4 CIDR	10.0.0.0/20
Network Address Usage metrics	Disabled
Main route table	rtb-0c01619ff7da9c31f
IPv6 pool	-
Route 53 Resolver DNS Firewall rule groups	-
DNS hostnames	Disabled
DNS resolution	Enabled
Owner ID	193633228050
Main network ACL	acl-03740408a8713e961
IPv6 CIDR (Network border group)	-

## 4. create private subnet and public subnet with different availability zones.

**Subnet Details:**

Setting	Value
Subnet ID	subnet-02dad0e345a7b46c6
Subnet ARN	arn:aws:ec2:us-east-1:193633228050:subnet/subnet-02dad0e345a7b46c6
State	Available
Available IPv4 addresses	505
IPv4 CIDR	-
Availability Zone	us-east-1c
Route table	rtb-041d9e650ad440910   my-rt-public
Network ACL	acl-03740408a8713e961
Default subnet	No
VPC	vpc-092b5fb14b0743f5a   my-vpc-route
Auto-assign IPv4 address	Yes
IPv4 CIDR reservations	-
Customer-owned IPv4 pool	-
Outpost ID	-
Resource name DNS A record	Disabled
IP name	-
Auto-assign customer-owned IPv4 address	No
IPv6 CIDR reservations	-
IPv6-only	No
Hostname type	IP name
DNS64	Disabled
Resource name DNS AAAA record	Disabled

The screenshot shows the AWS VPC Management Console. The left sidebar is titled "Virtual private cloud" and includes sections for "Your VPCs" (with a "New" button), "Subnets", "Route tables", "Internet gateways", "Egress-only internet gateways", "Carrier gateways", "DHCP option sets", "Elastic IPs", "Managed prefix lists", "Endpoints", "Endpoint services", "NAT gateways", and "Peering connections". The main content area is titled "VPC > Subnets > subnet-04b6b5a8d54a925d6" and displays the following details:

Details	Value
Subnet ID	subnet-04b6b5a8d54a925d6
Subnet ARN	arn:aws:ec2:us-east-1:193633228050:subnet/subnet-04b6b5a8d54a925d6
State	Available
IPv4 CIDR	10.0.2.0/24
Availability Zone	us-east-1d
Route table	rtb-02a9625bbc0407f00   RDS-Pvt-rt
Network ACL	acl-03740408a8713e961
Default subnet	No
VPC	vpc-092b5fb14b0743f5a   my-vpc-route
Customer-owned IPv4 pool	No
Auto-assign public IPv4 address	No
IPv6 CIDR	-
Auto-assign IPv6 address	No
IPv6 CIDR reservations	-
IPv6 CIDR reservations	-
IPv6 CIDR reservations	-
Outpost ID	-
Resource name DNS A record	Enabled
Hostname type	IP name
Disabled	

At the bottom, there are links for "CloudShell", "Feedback", "Language", and a search bar. The status bar at the bottom right shows "ENG IN" and the date "23-06-2023".

## 5. create internet gateway for vpc and associate with public subnet.

The screenshot shows the AWS VPC Management Console. The left sidebar is titled "Virtual private cloud" and includes sections for "Your VPCs" (with a "New" button), "Subnets", "Route tables", and "Internet gateways". The main content area is titled "VPC > Internet gateways > igw-000243bb4b1d7c885" and displays the following details:

Details	Value
Internet gateway ID	igw-000243bb4b1d7c885
State	Attached
VPC ID	vpc-092b5fb14b0743f5a   my-vpc-route
Owner	193633228050

Below the table, there is a "Tags" section with a table:

Key	Value
Name	my-igw

At the bottom, there are links for "CloudShell", "Feedback", "Language", and a search bar. The status bar at the bottom right shows "ENG IN" and the date "23-06-2023".

create nat gateway using public

# instance associate with pvt instance.

NAT gateway ID	Connectivity type	State	State message
nat-0bf847dd8711350f7	Public	Available	-
NAT gateway ARN	Primary public IPv4 address	Primary private IPv4 address	Primary network interface ID
arn:aws:ec2:us-east-1:19363322805:natgateway/nat-0bf847dd8711350f7	107.21.203.66	10.0.0.84	eni-0c0a420dfdb0c2a27
VPC	Subnet	Created	Deleted
vpc-092b5fb14b0743f5a / my-vpc-route	subnet-02dad0e345a7b46c6 / my-public	Friday, June 23, 2023 at 10:54:47 GMT+5:30	-

6.create public routetable and private route table for vpc.

1.in public rt attatch internet gateway and public subnet.

The screenshot shows the AWS VPC Management Console. On the left, a sidebar lists various VPC-related services and options like 'Virtual private cloud', 'Route tables', and 'Subnets'. The main panel displays the details of a specific route table:

Route table ID	Main	Explicit subnet associations	Edge associations
rtb-041d9e650ad440910	No	subnet-02dad0e345a7b46c6 / my-public	-
VPC	Owner ID		
vpc-092b5fb14b0743f5a   my-vpc-route	193633228050		

Below this, a table shows the routes defined in the route table:

Destination	Target	Status	Propagated
0.0.0.0/0	igw-000243bb4b1d7c885	Active	No
10.0.0.0/20	local	Active	No

The screenshot shows the list of route tables in the VPC Management Console. It includes columns for Name, Route table ID, Explicit subnet associations, Edge associations, Main status, and VPC. One route table is selected, showing its details:

Name	Route table ID	Explicit subnet associations	Edge associations	Main	VPC
my-rt-private	rtb-0a2648083b1a87987	subnet-08fc8b5fa2ac87269 / my-private	-	No	vpc-092b5fb14b0743f5a
	rtb-0fedfebf85f07c77d	-	-	Yes	vpc-040ef369

Below the table, the details of the selected route table (rtb-041d9e650ad440910) are shown, mirroring the structure of the first screenshot.

## 2. in private rt attatch private subnet and nat gateway.

The screenshot shows the AWS VPC Management console with the 'Route tables' page open. The left sidebar shows various VPC-related services like VPC dashboard, EC2 Global View, and Route tables. The main area shows a table of route tables with one entry:

Name	Route table ID	Explicit subnet associations	Edge associations	Main	VPC
my-rt-private	rtb-0a2648083b1a87987	subnet-08fc8b5fa2ac87269 / my-private	-	No	vpc-092b5fb
	rtb-0fffeff85f0777d	-	-	Yes	vnr-040f1569

Below the table, the details for 'rtb-0a2648083b1a87987 / my-rt-private' are shown. The 'Routes' tab is selected, displaying two routes:

- Destination: 0.0.0.0/0 Target: nat-0bf847dd8711350f7 Status: Active Propagated: No
- Destination: 10.0.0.0/20 Target: local Status: Active Propagated: No

## 7.create public instance attatch vpc and sg give port 80

**Instance summary for i-069ca67b63e93c3d3 (public instance)**

Instance ID	i-069ca67b63e93c3d3 (public instance)	Public IPv4 address	10.0.0.43
IPv6 address	-	Instance state	Running
Hostname type	IP name: ip-10-0-0-43.ec2.internal	Private IP DNS name (IPv4 only)	ip-10-0-0-43.ec2.internal
Answer private resource DNS name	-	Instance type	t2.micro
Auto-assigned IP address	34.229.103.136 [Public IP]	VPC ID	vpc-092b5fb14b0745f5a (my-vpc-route)
IAM Role	-	Subnet ID	subnet-02dad0e345a7b46c6 (my-public)
IMDSv2	Optional	Elastic IP addresses	-
		AWS Compute Optimizer finding	Opt-in to AWS Compute Optimizer for recommendations.
			Learn more
		Auto Scaling Group name	-

**Security group name**: my-sg

**Description**: my-sg

**VPC ID**: vpc-092b5fb14b0745f5a

**Inbound rules (3)**

Name	Security group rule...	IP version	Type	Protocol	Port range
sgr-0dd221b5ba5d89...	sgr-0dd221b5ba5d89...	IPv6	HTTP	TCP	80
sgr-07f97dc00157ed478	sgr-07f97dc00157ed478	IPv4	SSH	TCP	22
sgr-0930241e3ab1d8...	sgr-0930241e3ab1d8...	IPv4	HTTP	TCP	80

## 8. create a ec2 instance in vpc and sg assain port 80 & 3306

The screenshot shows the AWS EC2 Management Console. The left sidebar is collapsed, and the main area displays the 'Instances' section. A specific instance, 'i-06bdd6a71424aeafa4 (my-private)', is selected. The 'Details' tab is active, showing the following information:

Instance ID	Public IPv4 address	Private IPv4 addresses	
i-06bdd6a71424aeafa4 (my-private)	-	10.0.9.80	
IPv6 address	Instance state	Public IPv4 DNS	
-	Running	-	
Hostname type	Private IP DNS name (IPv4 only)	Elastic IP addresses	
IP name: ip-10-0-9-80.ec2.internal	ip-10-0-9-80.ec2.internal	-	
Answer private resource DNS name	Instance type	AWS Compute Optimizer finding	
-	t2.micro	<a href="#">Opt-in to AWS Compute Optimizer for recommendations.</a>	
Auto-assigned IP address	VPC ID	Auto Scaling Group name	
-	vpc-092b5fb14b0745f5a (my-vpc-route)	-	
IAM Role	Subnet ID	<a href="#">Learn more</a>	
-	subnet-08fc8b5fa2ac87269 (my-private)		
IMDSv2	Optional		

Below the table, there are tabs for Details, Security, Networking, Storage, Status checks, Monitoring, and Tags. The status bar at the bottom indicates the date and time as 23-06-2023.

The screenshot shows the AWS Network & Security - Security Groups page. The left sidebar is collapsed, and the main area displays the 'Inbound rules' section for a security group named 'sg-07a39ab88a399c63'. The 'Inbound rules' tab is active, showing four entries:

Name	Security group rule...	IP version	Type	Protocol	Port range
-	sgr-0db84e160cad0a...	IPv6	HTTP	TCP	80
-	sgr-07d304435e595b...	IPv4	SSH	TCP	22
-	sgr-09c2b8280f06524d	IPv4	HTTP	TCP	80
-	sgr-0e41126b901ee80...	-	MySQL/Aurora	TCP	3306

Below the table, there are tabs for Inbound rules, Outbound rules, and Tags. A message box says 'You can now check network connectivity with Reachability Analyzer' with a 'Run Reachability Analyzer' button. The status bar at the bottom indicates the date and time as 23-06-2023.

## 9.create RDS database using mysql and create default subnet group ...

The screenshot shows the AWS RDS Management Console. On the left, a sidebar lists various services like Database, Query Editor, and Performance insights. The main area displays the 'Summary' tab for a database named 'database-1'. Key details shown include:

DB identifier	CPU	Status	Class
database-1	3.81%	Available	db.t2.micro
Role	Current activity	Engine	Region & AZ
Instance	0 Connections	MySQL Community	us-east-1b

Below the summary, there are tabs for Connectivity & security, Monitoring, Logs & events, Configuration, Maintenance & backups, and Tags. The Connectivity & security tab is selected, showing the endpoint information and security settings.

The screenshot shows the AWS RDS Management Console. On the left, a sidebar lists various services like Database, Query Editor, and Performance insights. The main area displays the 'Subnet group details' for a subnet group named 'rds-ec2-db-subnet-group-2'. Key details shown include:

VPC ID
vpc-092b5fb14b0743f5a

Below the subnet group details, there is a table for 'Subnets (5)'. The table lists five subnets across two availability zones:

Availability zone	Subnet ID	CIDR block
us-east-1c	subnet-0c21c9683147f8476	10.0.3.0/25
us-east-1b	subnet-04a952a280a67410e	10.0.2.0/25

## 10.connect public instance in terminal.

```

Windows PowerShell
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\gandh> cd .\Downloads>
PS C:\Users\gandh\Downloads> ssh -i "nani.pem" ec2-user@34.229.103.136
The authenticity of host '34.229.103.136 (34.229.103.136)' can't be established.
ED25519 key fingerprint is SHA256:Au8EJlex/nzs2E+2WXAko7hHvMnBhKjh7j195e0Mc.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '34.229.103.136' (ED25519) to the list of known hosts.

 _-| _--|_
 _|- ( _- / Amazon Linux 2 AMI
 ___\_\_\_|\_\_\_|

https://aws.amazon.com/amazon-linux-2/

```

11. open new window and copy pem file using command (scp -i pem.pem pem.pem ec2-user@public ip :~)

```

Windows PowerShell
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\gandh> scp -i nani.pem nani.pem ec2-user@34.229.103.136:~
Warning: Identity file nani.pem not accessible: No such file or directory.
ec2-user@34.229.103.136: Permission denied (publickey,gssapi-keyex,gssapi-with-mic).
lost connection
PS C:\Users\gandh> scp -i nani.pem nani.pem ec2-user@34.229.103.136:~
Warning: Identity file nani.pem not accessible: No such file or directory.
ec2-user@34.229.103.136: Permission denied (publickey,gssapi-keyex,gssapi-with-mic).
lost connection
PS C:\Users\gandh> cd .\Downloads\>
PS C:\Users\gandh\Downloads> scp -i nani.pem nani.pem ec2-user@34.229.103.136:~
nani.pem
100% 1678      6.5KB/s   00:00
PS C:\Users\gandh\Downloads> |

```

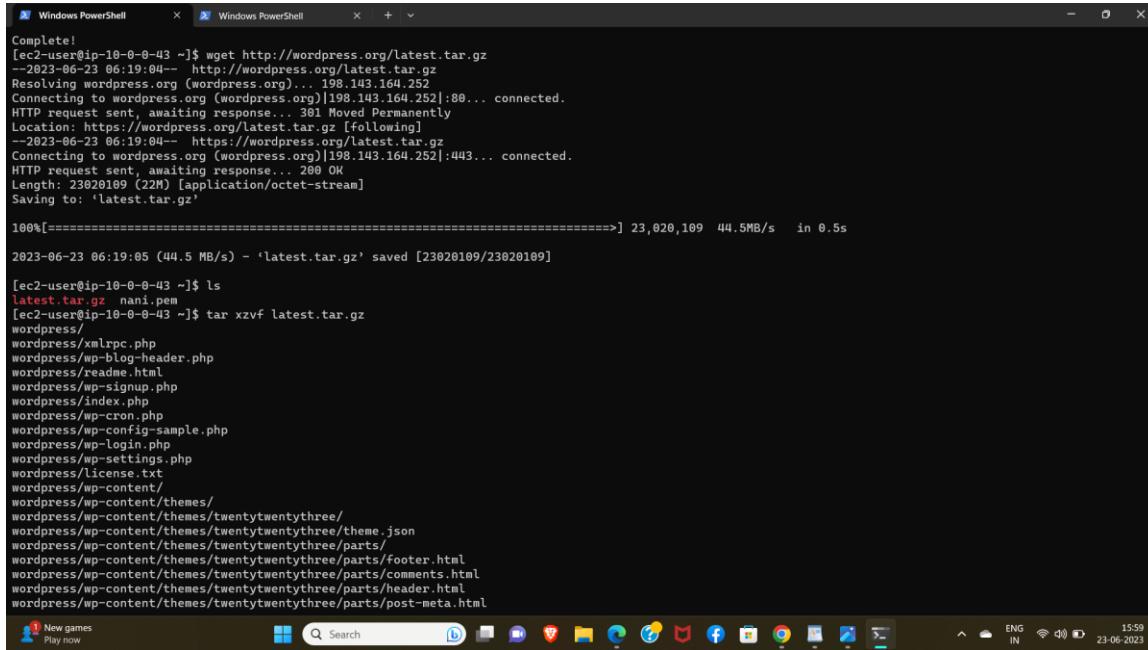
12. check for public instance and change root module.

```

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-10-0-0-43 ~]$ ls
[ec2-user@ip-10-0-0-43 ~]$ ls
nani.pem
[ec2-user@ip-10-0-0-43 ~]$ ll
total 4
-rw-rw-r-- 1 ec2-user ec2-user 1678 Jun 23 05:47 nani.pem
[ec2-user@ip-10-0-0-43 ~]$ chmod 400 nani.pem
[ec2-user@ip-10-0-0-43 ~]$ ll
total 4
-r----- 1 ec2-user ec2-user 1678 Jun 23 05:47 nani.pem
[ec2-user@ip-10-0-0-43 ~]$ scp -i "nani.pem" ec2-user@19.8.9.89

```

## 13. install wordpress for our local mishene.



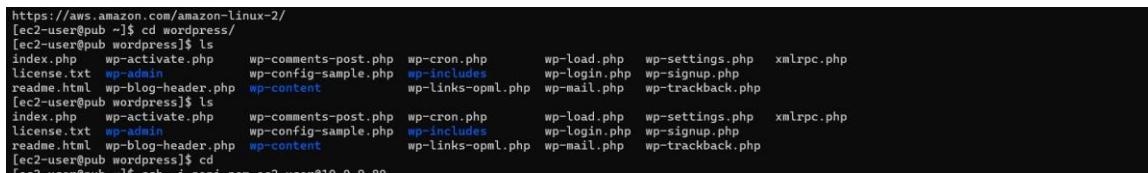
```
[ec2-user@ip-10-0-0-43 ~]$ wget http://wordpress.org/latest.tar.gz
--2023-06-23 06:19:04--  http://wordpress.org/latest.tar.gz
Resolving wordpress.org (wordpress.org)... 198.143.164.252
Connecting to wordpress.org (wordpress.org)|198.143.164.252|:80... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: https://wordpress.org/latest.tar.gz [following]
--2023-06-23 06:19:04--  https://wordpress.org/latest.tar.gz
Connecting to wordpress.org (wordpress.org)|198.143.164.252|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 23020109 (22M) [application/octet-stream]
Saving to: 'latest.tar.gz'

100%[=====] 23,020,109  44.5MB/s   in 0.5s

2023-06-23 06:19:05 (44.5 MB/s) - 'latest.tar.gz' saved [23020109/23020109]

[ec2-user@ip-10-0-0-43 ~]$ ls
latest.tar.gz  nani.pem
[ec2-user@ip-10-0-0-43 ~]$ tar xzvf latest.tar.gz
wordpress/
wordpress/xrplrc.php
wordpress/wp-blog-header.php
wordpress/readme.html
wordpress/wp-signup.php
wordpress/index.php
wordpress/wp-cron.php
wordpress/wp-config-sample.php
wordpress/wp-login.php
wordpress/wp-settings.php
wordpress/license.txt
wordpress/wp-content/
wordpress/wp-content/themes/
wordpress/wp-content/themes/twentytwentythree/
wordpress/wp-content/themes/twentytwentythree/theme.json
wordpress/wp-content/themes/twentytwentythree/part/
wordpress/wp-content/themes/twentytwentythree/part/footer.html
wordpress/wp-content/themes/twentytwentythree/part/comments.html
wordpress/wp-content/themes/twentytwentythree/part/header.html
wordpress/wp-content/themes/twentytwentythree/part/post-meta.html
```

## 14.goto wordpress directory and give ls to see files.



```
https://aws.amazon.com/amazon-linux-2/
[ec2-user@pub ~]$ cd wordpress/
[ec2-user@pub wordpress]$ 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@pub wordpress]$ 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@pub wordpress]$ cd
```

## 15.install mariodb , php , mysql.(to search for commands install php mariodb in amazon linux)

```
Windows PowerShell x Windows PowerShell x + - 
[ec2-user@pub ~]$ MariaDB on Amazon Linux 2
n
We will learn how to set up PHP and Maria-Bash: MariaDB: command not found
[ec2-user@pub ~]$ #
[ec2-user@pub ~]$ aws
DB on Amazon Linux 2 in this tutorial. We will also discover how to set up PHP so that it functions with the Apache web server. We will also discover how to set up MariaDB so that it functions with PHP.

Prerequisites
To follow along with this tutorial, you will need:

An Amazon Linux 2 EC2 instance with a public IP address.
A non-root user with sudo privileges.
A domain name pointing to the public IP address of your EC2 instance.
Apache web server installed and running. How to Install Apache Web Server on Amazon Linux 2.
Installing PHP/MariaDB, setting up MariaDB, and running a basic PHP demo
Step 1 - Installing PHP
PHP is a free and open-source scripting language that is used to create dynamic web pages. It is the most popular web scripting language in the world.

At first, we will enable amazon-linux-extras so that we can specify the PHP version that we want to install.
Note: AWS CLI version 2, the latest major version of the AWS CLI, is now stable and recommended for general use. For more information, see the AWS CLI version 2 installation instructions at: https://docs.aws.amazon.com/cli/latest/userguide/install-cliv2.html

usage: aws [options] <command> <subcommand> [<subcommand> ...] [parameters]
To see help text, you can run:

aws help
aws <command> help
aws <command> <subcommand> help
aws: error: too few arguments
[ec2-user@pub ~]$ #
[ec2-user@pub ~]$ ec2
-bash: ec2: command not found
[ec2-user@pub ~]$ #
[ec2-user@pub ~]$ amazonlinux2
-bash: amazonlinux2: command not found
[ec2-user@pub ~]$ #
[ec2-user@pub ~]$ php
-bash: php: command not found

1 New games
Play now
Search ENG IN 23-06-2023 16:05
```

```
Windows PowerShell x Windows PowerShell x + - 
# yum clean metadata
# yum install php-cli php-pdo php-fpm php-json php-mysqld
[ec2-user@pub ~]$ sudo yum install php php-{pear,cgi,common,curl,mbstring,gd,mysqlnd,gettext,bcmath,json,xml,fpm,intl,zip,imap} -y
> ~c
[ec2-user@pub ~]$ sudo yum install php php-{pear,cgi,common,curl,mbstring,gd,mysqlnd,gettext,bcmath,json,xml,fpm,intl,zip,imap} -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core
amzn2extra-docker
amzn2extra-kernel-5.10
amzn2extra-php7.4
(1/2): amzn2extra-php7.4/2/x86_64/updateinfo
(2/2): amzn2extra-php7.4/2/x86_64/primary_db
No package php-imap available.
Resolving Dependencies
--> Running transaction check
--> Package php.x86_64 0:7.4.33-1.amzn2 will be installed
--> Processing Dependency: httpd-mm = 20120211x8664 for package: php-7.4.33-1.amzn2.x86_64
--> Package php-bcmath.x86_64 0:7.4.33-1.amzn2 will be installed
--> Package php-cli.x86_64 0:7.4.33-1.amzn2 will be installed
--> Package php-common.x86_64 0:7.4.33-1.amzn2 will be installed
--> Processing Dependency: libzip.so.5()(64bit) for package: php-common-7.4.33-1.amzn2.x86_64
--> Package php-fpm.x86_64 0:7.4.33-1.amzn2 will be installed
--> Package php-gd.x86_64 0:7.4.33-1.amzn2 will be installed
--> Package php-intl.x86_64 0:7.4.33-1.amzn2 will be installed
--> Package php-json.x86_64 0:7.4.33-1.amzn2 will be installed
--> Package php-mbstring.x86_64 0:7.4.33-1.amzn2 will be installed
--> Processing Dependency: libbonig.so.2()(64bit) for package: php-mbstring-7.4.33-1.amzn2.x86_64
--> Package php-mysqlnd.x86_64 0:7.4.33-1.amzn2 will be installed
--> Processing Dependency: php-pdo(x86-64) = 7.4.33-1.amzn2 for package: php-mysqlnd-7.4.33-1.amzn2.x86_64
--> Package php-pEAR.noarch 1:1.10.12-9.amzn2 will be installed
--> Processing Dependency: php-pear for package: 1:php-pear-1.10.12-9.amzn2.noarch
--> Package php-xml.x86_64 0:7.4.33-1.amzn2 will be installed
--> Processing Dependency: libxml.so.1(LIBXML2_1.0.24)(64bit) for package: php-xml-7.4.33-1.amzn2.x86_64
--> Processing Dependency: libxml.so.1(LIBXML2_1.0.22)(64bit) for package: php-xml-7.4.33-1.amzn2.x86_64
--> Processing Dependency: libxml.so.1(LIBXML2_1.0.18)(64bit) for package: php-xml-7.4.33-1.amzn2.x86_64
--> Processing Dependency: libxml.so.1(LIBXML2_1.0.13)(64bit) for package: php-xml-7.4.33-1.amzn2.x86_64
--> Processing Dependency: libxml.so.1(LIBXML2_1.0.11)(64bit) for package: php-xml-7.4.33-1.amzn2.x86_64
--> Processing Dependency: libxml.so.1()(64bit) for package: php-xml-7.4.33-1.amzn2.x86_64
--> Processing Dependency: libxml.so.0()(64bit) for package: php-xml-7.4.33-1.amzn2.x86_64
--> Running transaction check

1 New games
Play now
Search ENG IN 23-06-2023 16:08
```

```

[ec2-user@pub ~]$ php -v
PHP 7.4.33 (cli) (built: Nov 19 2022 00:22:13) ( NTS )
Copyright (c) The PHP Group
Zend Engine v3.4.0, Copyright (c) Zend Technologies
[ec2-user@pub ~]$ sudo yum install mariadb-server -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Resolving Dependencies
--> Running transaction check
--> Package mariadb-server.x86_64 1:5.5.68-1.amzn2.0.1 will be installed
--> Processing Dependency: perl-DBI for package: 1:mariadb-server-5.5.68-1.amzn2.0.1.x86_64
--> Processing Dependency: perl-DBD-MySQL for package: 1:mariadb-server-5.5.68-1.amzn2.0.1.x86_64
--> Processing Dependency: perl(Data:Dumper) for package: 1:mariadb-server-5.5.68-1.amzn2.0.1.x86_64
--> Processing Dependency: perl(DBI) for package: 1:mariadb-server-5.5.68-1.amzn2.0.1.x86_64
--> Running transaction check
--> Package perl-DBD-MySQL.x86_64 0:4.023-6.amzn2 will be installed
--> Package perl-DBI.x86_64 0:1.627-4.amzn2.0.2 will be installed
--> Processing Dependency: perl(RPC::PlServer) >= 0.2001 for package: perl-DBI-1.627-4.amzn2.0.2.x86_64
--> Processing Dependency: perl(RPC::PlClient) >= 0.2000 for package: perl-DBI-1.627-4.amzn2.0.2.x86_64
--> Package perl-Data-Dumper.x86_64 0:2.145-3.amzn2.0.2 will be installed
--> Running transaction check
--> Package perl-PLRPC.noarch 0:0.2020-14.amzn2 will be installed
--> Processing Dependency: perl(Net::Daemon) >= 0.13 for package: perl-PLRPC-0.2020-14.amzn2.noarch
--> Processing Dependency: perl(Net::Daemon::Test) for package: perl-PLRPC-0.2020-14.amzn2.noarch
--> Processing Dependency: perl(Net::Daemon::Log) for package: perl-PLRPC-0.2020-14.amzn2.noarch
--> Processing Dependency: perl(Compress::Zlib) for package: perl-PLRPC-0.2020-14.amzn2.noarch
--> Running transaction check
--> Package perl-IO-Compress.noarch 0:2.061-2.amzn2 will be installed
--> Processing Dependency: perl(Compress::Raw::Bzip2) >= 2.061 for package: perl-IO-Compress-2.061-2.amzn2.noarch
--> Processing Dependency: perl(Compress::Raw::Zlib) >= 2.061 for package: perl-IO-Compress-2.061-2.amzn2.noarch
--> Package perl-Net-Daemon.noarch 0:0.48-5.amzn2 will be installed
--> Running transaction check
--> Package perl-Compress-Raw-Bzip2.x86_64 0:2.061-3.amzn2.0.2 will be installed
--> Package perl-Compress-Raw-Zlib.x86_64 1:2.061-4.amzn2.0.2 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

=====
Arch          Version           Repository      Size
=====
perl-Data-Dumper.x86_64 0:2.145-3.amzn2.0.2
perl-Net-Daemon.noarch 0:0.48-5.amzn2

perl-IO-Compress.noarch 0:2.061-2.amzn2
perl-PLRPC.noarch 0:0.2020-14.amzn2

```

## 16. after instalation start and enable mariodb and generate a new password.

```

perl-Data-Dumper.x86_64 0:2.145-3.amzn2.0.2
perl-Net-Daemon.noarch 0:0.48-5.amzn2

perl-IO-Compress.noarch 0:2.061-2.amzn2
perl-PLRPC.noarch 0:0.2020-14.amzn2

[ec2-user@pub ~]$ sudo systemctl start mariadb
[ec2-user@pub ~]$ sudo systemctl enable mariadb
Created symlink from /etc/systemd/system/multi-user.target.wants/mariadb.service to /usr/lib/systemd/system/mariadb.service.
[ec2-user@pub ~]$ sudo mysql_secure_installation

NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MariaDB
      SERVERS IN PRODUCTION USE!  PLEASE READ EACH STEP CAREFULLY!

In order to log into MariaDB to secure it, we'll need the current
password for the root user. If you've just installed MariaDB, and
you haven't set the root password yet, the password will be blank,
so you should just press enter here.

Enter current password for root (enter for none):
OK, successfully used password, moving on...

Setting the root password ensures that nobody can log into the MariaDB
root user without the proper authorisation.

Set root password? [Y/n] y
New password:
Re-enter new password:
Sorry, passwords do not match.

New password:
Re-enter new password:
Password updated successfully!
Reloading privilege tables..
... Success!

By default, a MariaDB installation has an anonymous user, allowing anyone
to log into MariaDB without having to have a user account created for
them. This is intended only for testing, and to make the installation
go a bit smoother. You should remove them before moving into a
production environment.

[ec2-user@pub ~]$ 

```

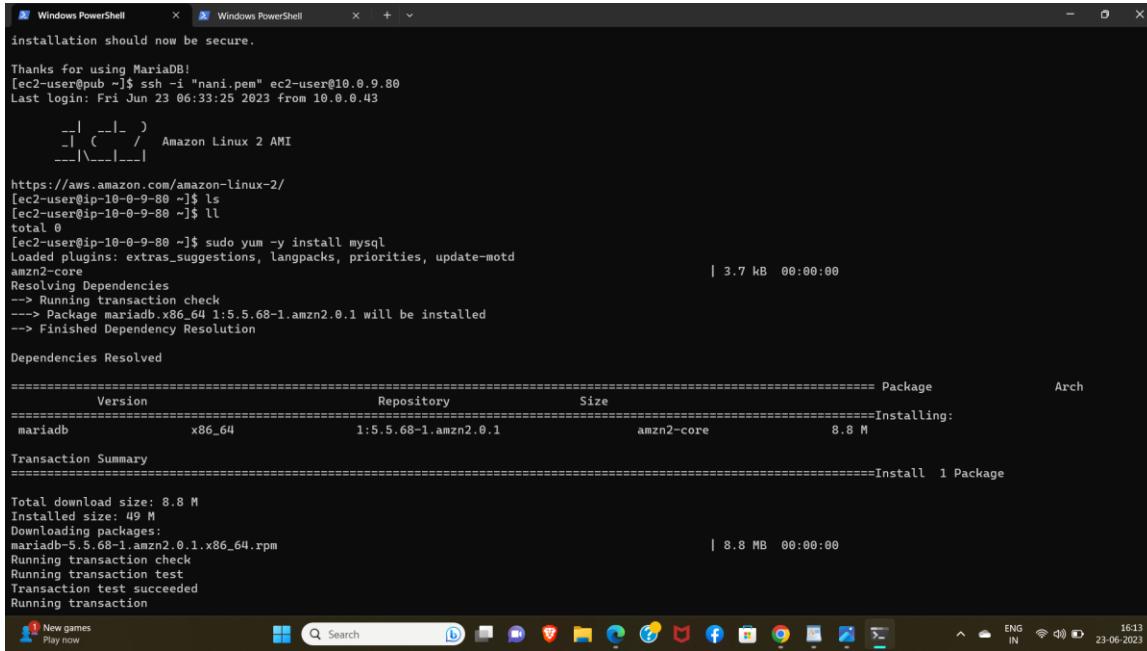
```

Setting the root password ensures that nobody can log into the MariaDB
root user without the proper authorisation.

Set root password? [Y/n] y
New password:
Re-enter new password:
Sorry, passwords do not match.

New password:
Re-enter new password:
Password updated successfully!
Reloading privilege tables..
... Success!
```

## 17. after installing mariadb connect public through private instance using comand (ssh -i pem.pem ec2-user@private ip)



```

Windows PowerShell      Windows PowerShell      +  ×
installation should now be secure.

Thanks for using MariaDB!
[ec2-user@ip-10-0-9-80 ~]$ ssh -i "nani.pem" ec2-user@10.0.9.80
Last login: Fri Jun 23 06:33:25 2023 from 10.0.0.43
 _|_ / Amazon Linux 2 AMI
 ___\___|___|
https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-10-0-9-80 ~]$ ls
[ec2-user@ip-10-0-9-80 ~]$ ll
total 0
[ec2-user@ip-10-0-9-80 ~]$ sudo yum -y install mysql
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core
Resolving Dependencies
--> Running transaction check
--> Package mariadb.x86_64 1:5.5.68-1.amzn2.0.1 will be installed
--> Finished Dependency Resolution
Dependencies Resolved
=====
 Version           Repository          Size
 ======          ======          =====
 mariadb        x86_64            1:5.5.68-1.amzn2.0.1      amzn2-core          8.8 M
=====
                                                Installing: 1 Package
Transaction Summary
=====
Total download size: 8.8 M
Installed size: 49 M
Downloading packages:
mariadb-5.5.68-1.amzn2.0.1.x86_64.rpm
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
| 8.8 MB  00:00:00
ENG IN 23-06-2023 16:13
```

## 18.connect into mysql and create rds user for mysql and exit .

```
> Ctrl-C -- exit!
Aborted
[ec2-user@pvt ~]$ sudo mysql -h database-1.chhnir3vkno.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 36
Server version: 8.0.32 Source distribution

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

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

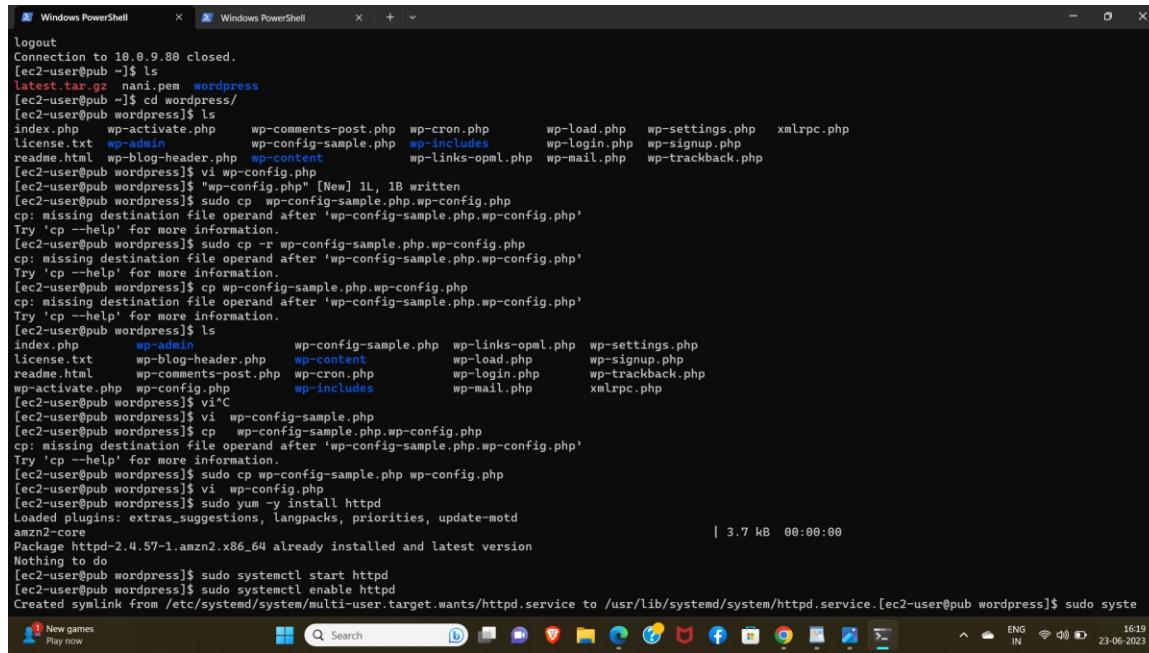
MySQL [(none)]> CREATE USER 'admin'@'localhost' IDENTIFIED BY 'admin1234';
Query OK, 0 rows affected (0.01 sec)

MySQL [(none)]> GRANT ALLPRIVILEGES ON admin.* TO 'admin'@'localhost';
ERROR 3619 (HY000): Illegal privilege level specified for ALLPRIVILEGES
MySQL [(none)]> GRANT ALL PRIVILEGES ON admin.* TO 'admin'@'localhost';
Query OK, 0 rows affected (0.01 sec)

MySQL [(none)]> show grants for "admin"@"localhost";
+-----+
| Grants for admin@localhost |
+-----+
| GRANT USAGE ON *.* TO 'admin'@'localhost' |
| GRANT ALL PRIVILEGES ON `admin`.* TO 'admin'@'localhost' |
+-----+
2 rows in set (0.00 sec)

MySQL [(none)]> select user,host from mysql.user;
+-----+-----+
| user | host |
+-----+-----+
| admin | %   |
| admin | localhost |
| mysql.infoschema | localhost |
| mysql.session | localhost |
| mysql.sys | localhost |
| rdsadmin | localhost |
+-----+-----+
```

19.after exit for priv ate instance u r in public instance then u go into wordpress and "create vi file for wp-config.php "and give "ls"to see files and "copy files from wp-config-simple.php to wp-config.php".

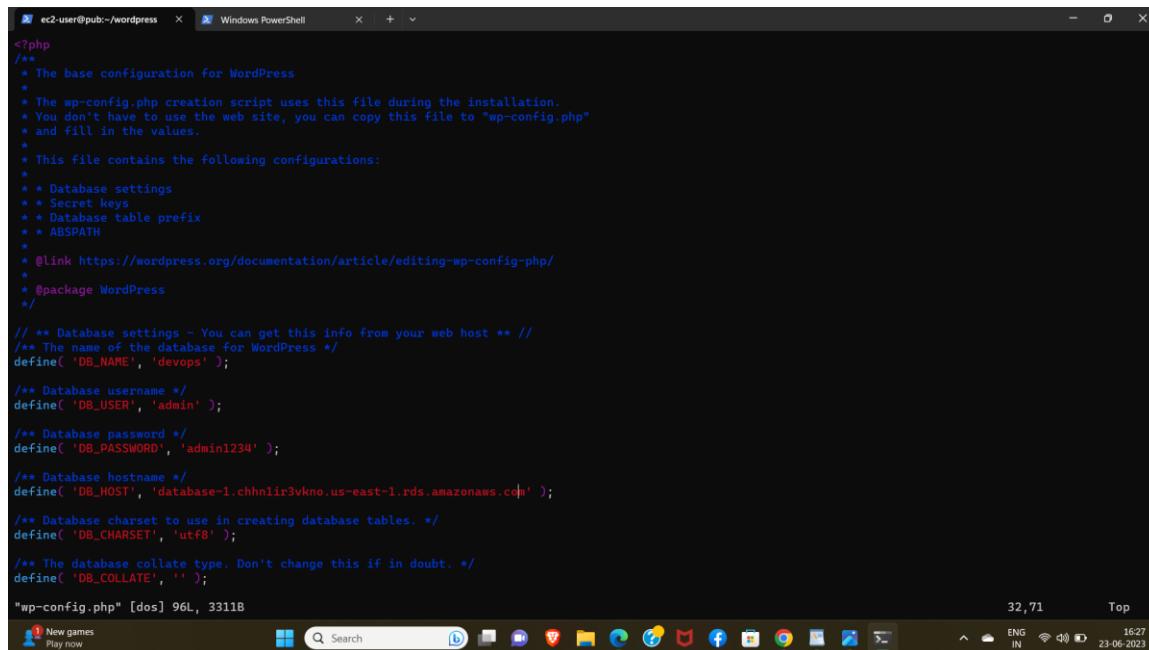


```

logout
Connection to 10.0.9.80 closed.
[ec2-user@pub ~]$ ls
latest.tar.gz nani.pem wordpress
[ec2-user@pub ~]$ cd wordpress/
[ec2-user@pub wordpress]$ 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@pub wordpress]$ vi wp-config.php
[ec2-user@pub wordpress]$ "wp-config.php" [New] 1L, 1B written
[ec2-user@pub wordpress]$ sudo cp wp-config-sample.php wp-config.php
cp: missing destination file operand after 'wp-config-sample.php.wp-config.php'
Try 'cp --help' for more information.
[ec2-user@pub wordpress]$ sudo cp -r wp-config-sample.php.wp-config.php
cp: missing destination file operand after 'wp-config-sample.php.wp-config.php'
Try 'cp --help' for more information.
[ec2-user@pub wordpress]$ cp wp-config-sample.php.wp-config.php
cp: missing destination file operand after 'wp-config-sample.php.wp-config.php'
Try 'cp --help' for more information.
[ec2-user@pub wordpress]$ ls
index.php wp-admin wp-config-sample.php wp-links-opml.php wp-settings.php
license.txt wp-blog-header.php wp-content wp-load.php wp-signup.php
readme.html wp-comments-post.php wp-cron.php wp-login.php wp-trackback.php
wp-activate.php wp-config.php wp-includes wp-mail.php xmlrpc.php
[ec2-user@pub wordpress]$ vi 'C'
[ec2-user@pub wordpress]$ vi wp-config-sample.php
[ec2-user@pub wordpress]$ cp wp-config-sample.php wp-config.php
cp: missing destination file operand after 'wp-config-sample.php.wp-config.php'
Try 'cp --help' for more information.
[ec2-user@pub wordpress]$ sudo cp wp-config-sample.php wp-config.php
[ec2-user@pub wordpress]$ vi wp-config.php
[ec2-user@pub wordpress]$ sudo yum -y install httpd
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core
Package httpd-2.4.57-1.amzn2.x86_64 already installed and latest version
Nothing to do
[ec2-user@pub wordpress]$ sudo systemctl start httpd
[ec2-user@pub wordpress]$ sudo systemctl enable httpd
Created symlink from /etc/systemd/system/multi-user.target.wants/httpd.service to /usr/lib/systemd/system/httpd.service.[ec2-user@pub wordpress]$ sudo syste
| 3.7 kB  00:00:00

```

20.open config file like " vi wp-config.php" and modify keys and username and password and database end point in host. and save .



```

<?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 web site, 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://wordpress.org/documentation/article/editing-wp-config-php/
 *
 * @package WordPress
 */
// ** Database settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define( 'DB_NAME', 'devops' );

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

/** Database password */
define( 'DB_PASSWORD', 'admin1234' );

/** Database hostname */
define( 'DB_HOST', 'database-1.chhnir3vkno.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', '' );

```

```


define( 'DB_HOST', 'database-1.chhmlir3vkno.us-east-1.rds.amazonaws.com' );


define( 'DB_CHARSET', 'utf8' );


define( 'DB_COLLATE', '' );

/*@@+
 * Authentication unique Keys and salts.
 *
 * Change these to different unique phrases! You can generate these using
 * the {@link https://api.wordpress.org/secret-key/1.1/salt/ WordPress.org secret-key service}.
 *
 * You can change these at any point in time to invalidate all existing cookies.
 * This will force all users to have to log in again.
 *
 * Since 2.6.0
 */
define('AUTH_KEY', 'D%eGEKwM[Su39aXJ+C+0Z)*+Wl(=0Z_ )jWG+.M22vqg,xrcM9c|dU@]d1.!Xjca');
define('SECURE_AUTH_KEY', 'bW00-l)+4<lt(XoC_K4w)Ul_17w^B@ral9nC19_jh2_(xNNGyx)a958WzY:n6|a');
define('LOGGED_IN_KEY', '#E{gj^eYo)QSBygxk-E]3cR(@:W:Wlg-BB|-s:Mwjs+ZC>+$Im(Eao6u+j0|vOs');
define('NONCE_KEY', '*59F7%&EWoU-Ns+N(f)uVbV.yiXunscLzeD&c%_ty=j;nRH69HB*+-5*TjPt');
define('AUTH_SALT', '-Jtxb:09p[E:b,^~H)9Eb+4_!eF<rBXUF,B_mp&q)-ys6T'pum@/Ijz+67RmO');
define('SECURE_AUTH_SALT', '>+PkySEH>Z-XH-ud_0,Onl4/bxc{_Zlb_+[L02-<0+3G2tFvc_1wzisvSwHPj');
define('LOGGED_IN_SALT', 'r:-iwbSlvR.iactSWFCJC9U0'?ggsa5Toh8BY_264B[c=sXCV[$xX<-8wKw');
define('NONCE_SALT', 'vv$8K-v/-u-j Gq<-jSz/SyY-[{:UiuaIs#@Ib@TK;X}]{Ft#k;e[?]7YZMn');

/*@@*/
/*
 * 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_';

-- INSERT --

```

21.install and start and enable httpd  
and restart also.

```

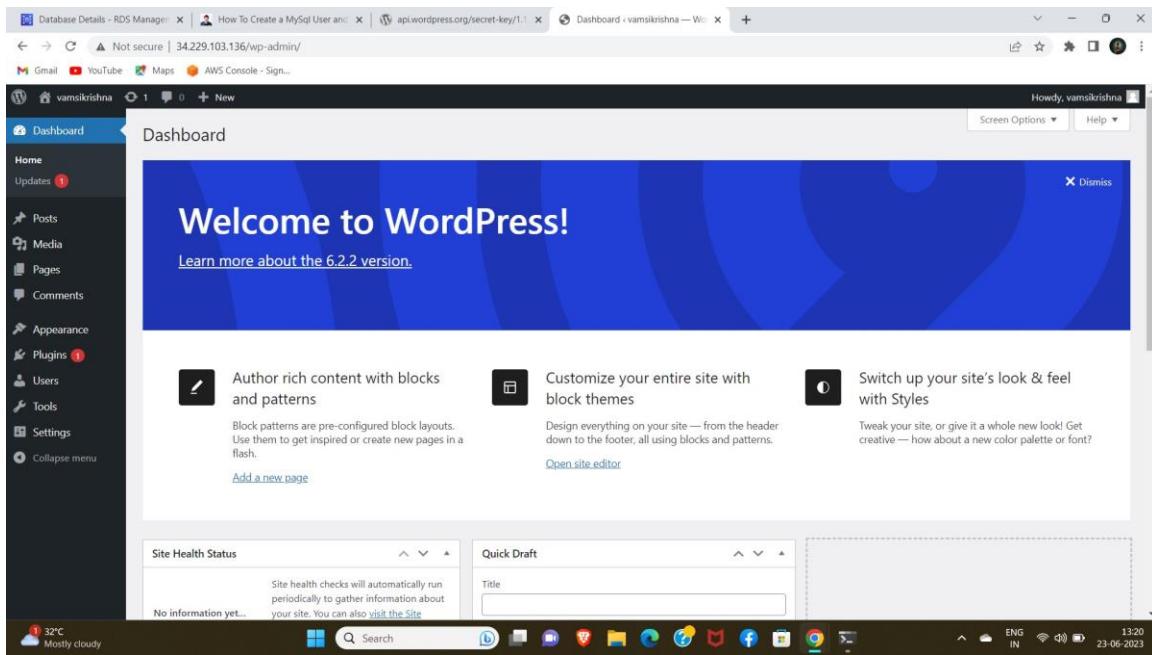
[ec2-user@pub wordpress]$ vi C
[ec2-user@pub wordpress]$ vi wp-config-sample.php
[ec2-user@pub wordpress]$ cp wp-config-sample.php wp-config.php
cp: missing destination file operand after 'wp-config-sample.php wp-config.php'
Try 'cp --help' for more information.
[ec2-user@pub wordpress]$ sudo cp wp-config-sample.php wp-config.php
[ec2-user@pub wordpress]$ vi wp-config.php
[ec2-user@pub wordpress]$ sudo yum -y install httpd
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core
Package httpd-2.4.57-1.amzn2.x86_64 already installed and latest version
Nothing to do
[ec2-user@pub wordpress]$ sudo systemctl start httpd
[ec2-user@pub wordpress]$ sudo systemctl enable httpd
Created symlink from /etc/systemd/system/multi-user.target.wants/httpd.service to /usr/lib/systemd/system/httpd.service.[ec2-user@pub wordpress]$ sudo systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; vendor preset: disabled)
   Drop-In: /usr/lib/systemd/system/httpd.service.d
             └─php-fpm.conf
     Active: active (running) since Fri 2023-06-23 07:40:30 UTC; 18s ago
       Docs: man:httpd.service(8)
 Main PID: 4100 (httpd)
    Status: "Total requests: 0; Idle/Busy workers 100/0;Requests/sec: 0; Bytes served/sec: 0 B/sec"
   CGroup: /system.slice/httpd.service
           ├─4100 /usr/sbin/httpd -DFOREGROUND
           ├─4107 /usr/sbin/httpd -DFOREGROUND
           ├─4108 /usr/sbin/httpd -DFOREGROUND
           ├─4109 /usr/sbin/httpd -DFOREGROUND
           ├─4110 /usr/sbin/httpd -DFOREGROUND
           └─4111 /usr/sbin/httpd -DFOREGROUND

Jun 23 07:40:30 pub systemd[1]: Starting The Apache HTTP Server...
Jun 23 07:40:30 pub systemd[1]: Started The Apache HTTP Server.
Jun 23 07:40:30 pub httpd[4100]: AH00558: httpd: Could not reliably determine the server's fully qualified domain name...messageHint: Some lines were ellipsized, use -l to show in full.
[ec2-user@pub wordpress]$ sudo systemctl restart httpd
[ec2-user@pub wordpress]$ vi wp-config.php
[ec2-user@pub wordpress]$ sudo systemctl restart httpd
[ec2-user@pub wordpress]$ ls /var/www/html/

```

22. then browse public instance public ip

adress and assain port ":80"  
then show the wordpress page.



login the page give user name and  
password & mail.





