cprime

Installing Apache Web Server with PHP and MariaDB on AWS EC2

NAME : T MANOJ

EMPID : LYAKE2KHS

Objective

The objective of this guide is to install and configure an Apache web server with PHP and MariaDB on an AWS EC2 instance. This setup allows users to host dynamic websites or applications with database connectivity.

By the end of this guide, you will be able to:

1. Launch an EC2 instance and connect via SSH.
2. Install and configure Apache, PHP, and MariaDB.
3. Set up a database and configure PHP scripts to interact with it.
4. Secure and optimize your web server for production use.

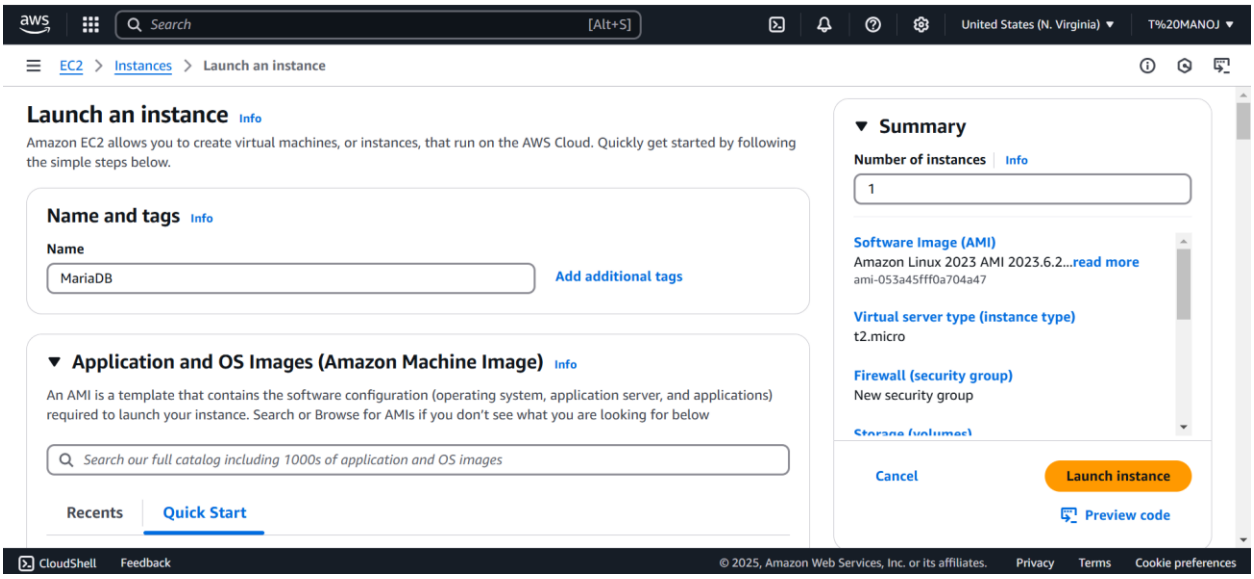
Advantages of Using Apache with PHP and MariaDB

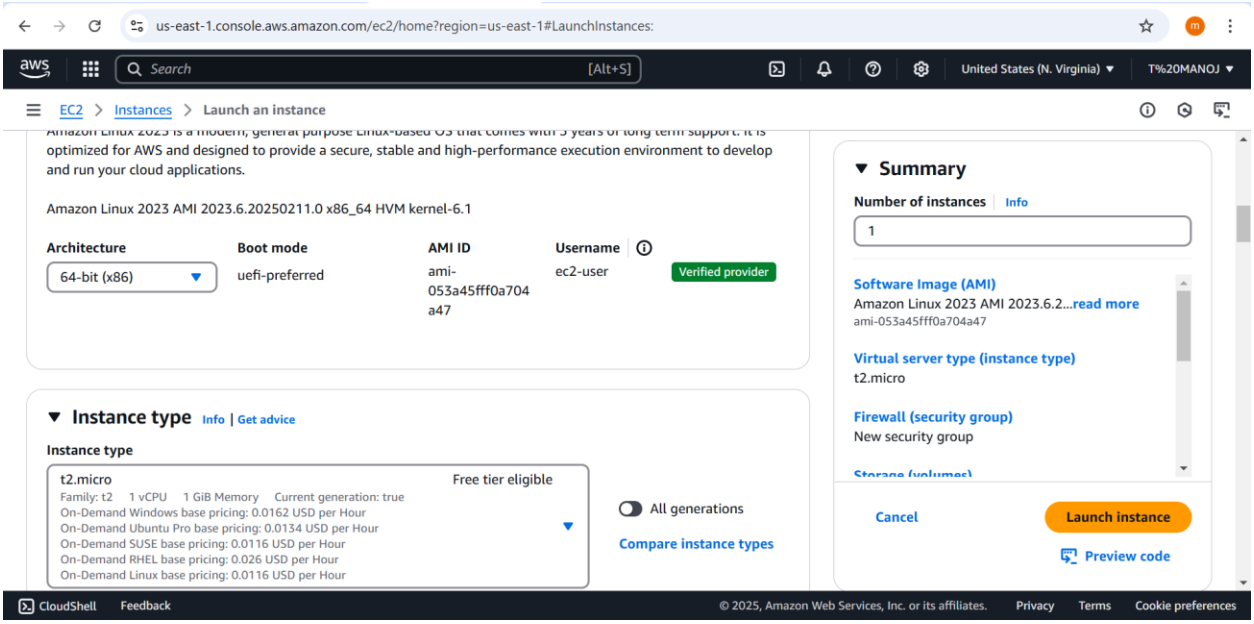
1. Scalability: Easily scales with AWS resources for high traffic websites.
2. Open Source & Cost-Effective: Apache, PHP, and MariaDB are open-source technologies with no licensing costs.
3. Security: MariaDB provides enhanced security features.
4. Flexibility: PHP supports various web frameworks, making development easier.
5. Integration: Compatible with AWS services like RDS, S3, CloudFront, and Route 53.

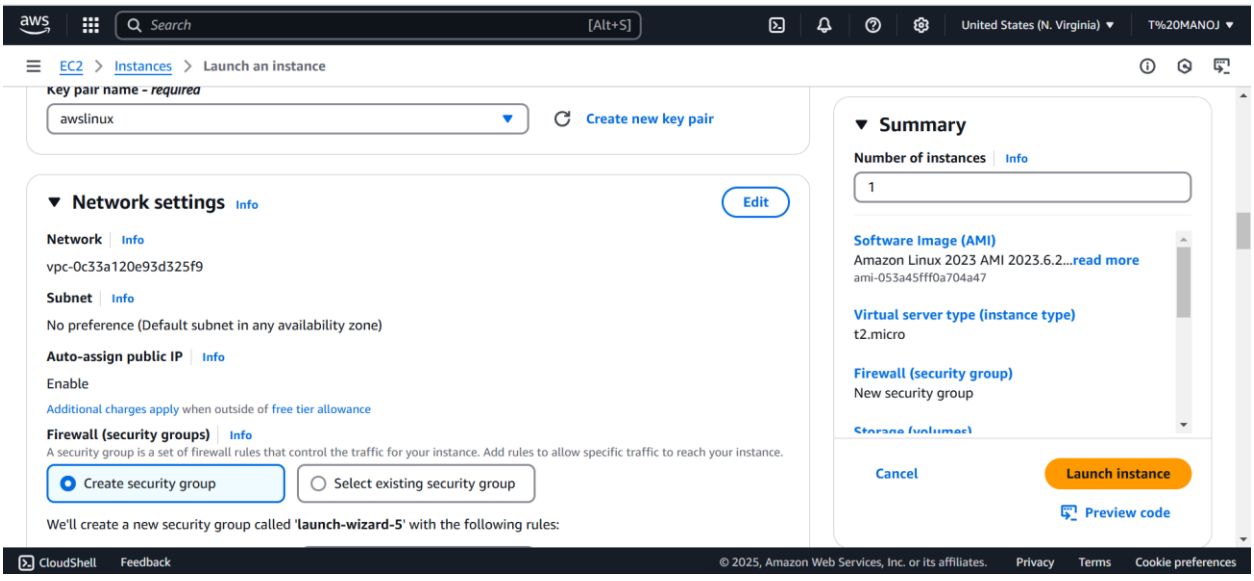
Steps to Install and Configure Apache Web Server with PHP and MariaDB

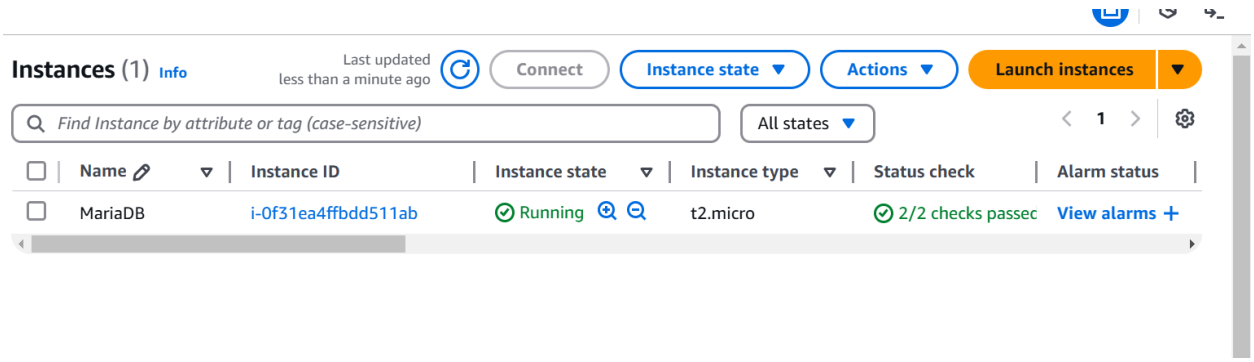
Step 1: Launch an EC2 Instance

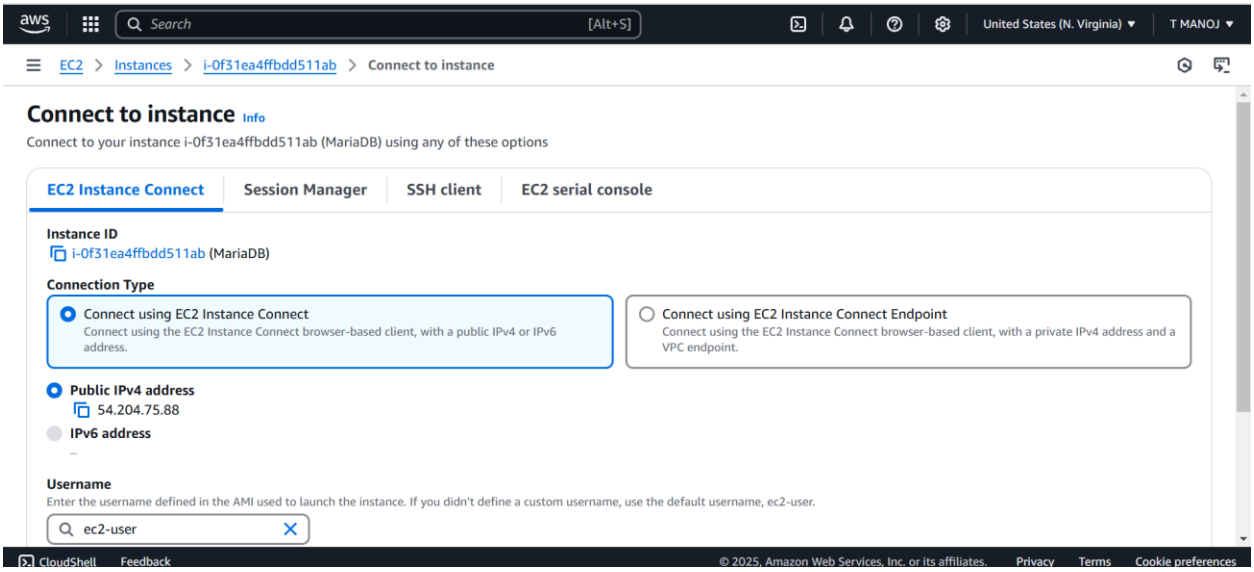
1. Go to the AWS Management Console and navigate to EC2.
2. Click Launch Instance and choose an Amazon Linux 2023 AMI.
3. Select an instance type .
4. Create or choose an existing key pair for SSH access.
5. Configure security group rules to allow:
    a. SSH (port 22) – Your IP only
    b. HTTP (port 80) – Open to all
    c. HTTPS (port 443) – Open to all
6. Launch the instance .




us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#LaunchInstances:
Search [Alt+S]
United States (N. Virginia)
T%20MANOJ
EC2 > Instances > Launch an instance
optimized for AWS and designed to provide a secure, stable and high-performance execution environment to develop and run your cloud applications.
Amazon Linux 2023 AMI 2023.6.20250211.0 x86_64 HVM kernel-6.1
Architecture
64-bit (x86)
Boot mode
uefi-preferred
AMI ID
ami-053a45fff0a704a47
Username
ec2-user
Verified provider
Instance type Info | Get advice
Instance type
t2.micro
Free tier eligible
Family: t2 1 vCPU 1 GiB Memory Current generation: true
On-Demand Windows base pricing: 0.0162 USD per Hour
On-Demand Ubuntu Pro base pricing: 0.0134 USD per Hour
On-Demand SUSE base pricing: 0.0116 USD per Hour
On-Demand RHEL base pricing: 0.026 USD per Hour
On-Demand Linux base pricing: 0.0116 USD per Hour
All generations
Compare instance types
Summary
Number of instances Info
1
Software Image (AMI)
Amazon Linux 2023 AMI 2023.6.2...read more
ami-053a45fff0a704a47
Virtual server type (instance type)
t2.micro
Firewall (security group)
New security group
Cancel
Launch instance
Preview code
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

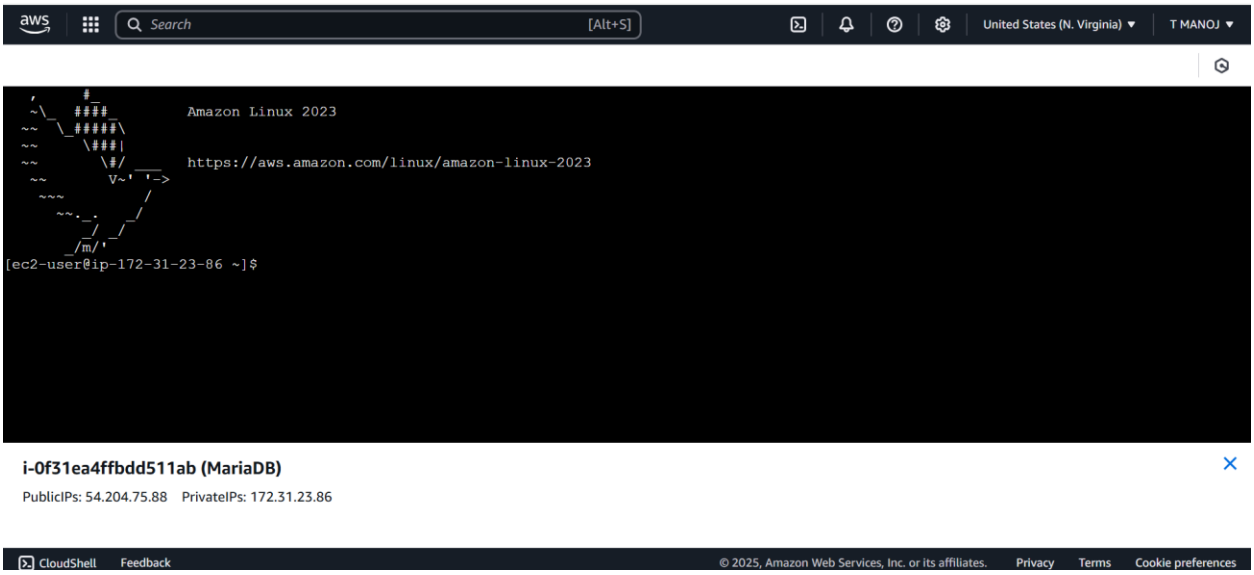

Search [Alt+S]
United States (N. Virginia)
T%20MANOJ
EC2 > Instances > Launch an instance
Key pair name - required
awslinux
Create new key pair
Network settings Info
Edit
Network Info
vpc-0c33a120e93d325f9
Subnet Info
No preference (Default subnet in any availability zone)
Auto-assign public IP Info
Enable
Additional charges apply when outside of free tier allowance
Firewall (security groups) Info
A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.
Create security group
Select existing security group
We'll create a new security group called 'launch-wizard-5' with the following rules:
Summary
Number of instances Info
1
Software Image (AMI)
Amazon Linux 2023 AMI 2023.6.2...read more
ami-053a45fff0a704a47
Virtual server type (instance type)
t2.micro
Firewall (security group)
New security group
Cancel
Launch instance
Preview code
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


Instances (1) Info
Last updated less than a minute ago
Connect
Instance state
Actions
Launch instances
Find Instance by attribute or tag (case-sensitive)
All states
1
Name
Instance ID
Instance state
Instance type
Status check
Alarm status
MariaDB
i-0f31ea4ffbdd511ab
Running
t2.micro
2/2 checks passec
View alarms +



Step 2: Connect to EC2 Instance via SSH

1. Open a terminal and use the following command
2. Confirm the connection by checking the system version

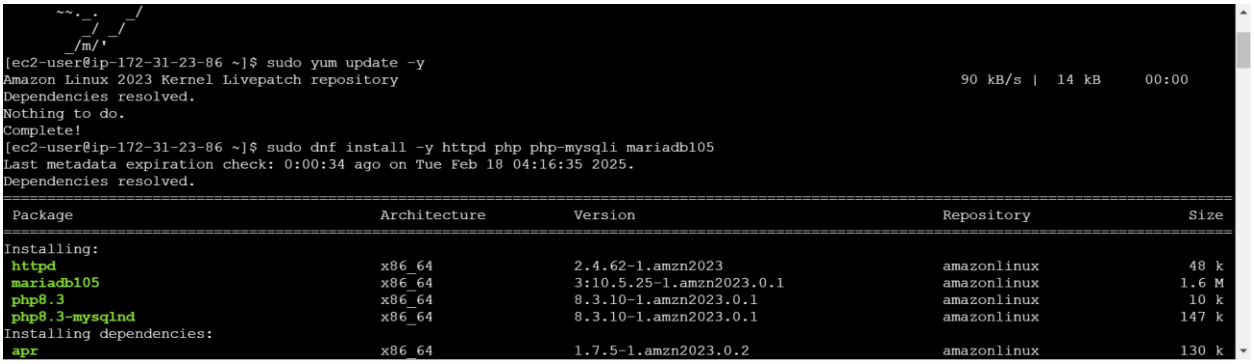


Step 3: Get the latest bug fixes and security updates by updating the software on your EC2 instance. To do this, use the following command.

Cmd : sudo dnf update –y

Step 4: After the updates complete, install the Apache web server, PHP, and MariaDB or PostgreSQL software using the following commands. This command installs multiple software packages and related dependencies at the same time .

Cmd : sudo dnf install -y httpd php php-mysqli mariadb105



Step 5 : Start the web server with the command shown following.

Cmd : sudo systemctl start httpd

Step 6 : Configure the web server to start with each system boot using the systemctl command.

Cmd : sudo systemctl enable httpd

Step 7 : **To set file permissions for the Apache web server**

1. Add the ec2-user user to the apache group.

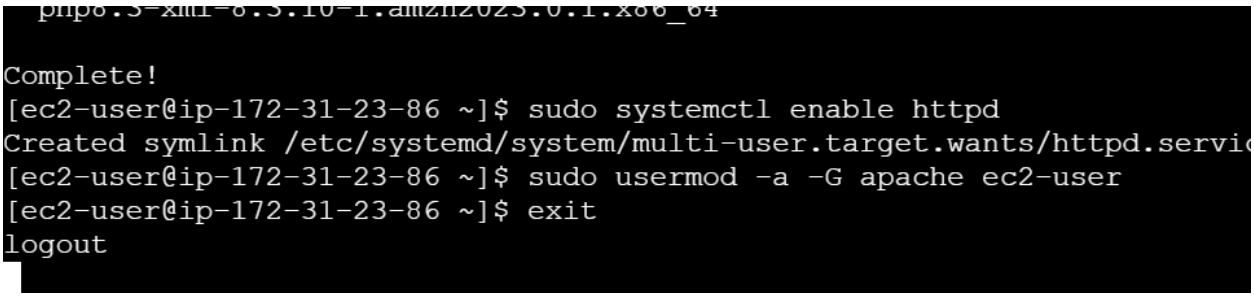
   Cmd : sudo usermod -a -G apache ec2-user

2. Log out to refresh your permissions and include the new apache group.

Cmd : exit

3. Log back in again and verify that the apache group exists with the groups command.

Cmd : groups



Step 8 : Change the group ownership of the /var/www directory and its contents to the apache group.

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

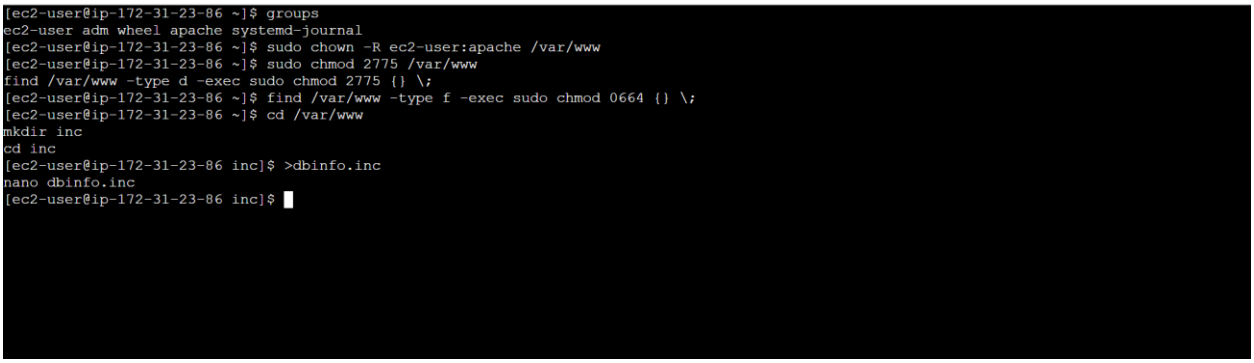
Step 9 : Change the directory permissions of /var/www and its subdirectories to add group write permissions and set the group ID on subdirectories created in the future.

Cmd : sudo chmod 2775 /var/www

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

Step 10 : Recursively change the permissions for files in the /var/www directory and its subdirectories to add group write permissions.

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



Connect your Apache web server to your DB instance

Step 11 : While still connected to your EC2 instance, change the directory to /var/www and create a new subdirectory named inc.

Cmd : cd /var/www

mkdir inc

cd inc

Step 12 : Create a new file in the inc directory named dbinfo.inc, and then edit the file by calling nano (or the editor of your choice).

Cmd : >dbinfo.inc

nano dbinfo.inc

Step 13 : Add the following contents to the dbinfo.inc file. Here, *db_instance_endpoint* is your DB instance endpoint, without the port, for your DB instance.

Cmd :

```
<?php

define('DB_SERVER', 'db_instance_endpoint');
define('DB_USERNAME', 'tutorial_user');
define('DB_PASSWORD', 'master password');
define('DB_DATABASE', 'sample');
?>
```

```
[ec2-user@ip-172-31-23-86 inc]$ >dbinfo.inc
nano dbinfo.inc
[ec2-user@ip-172-31-23-86 inc]$ cat dbinfo.inc
<?php

define('DB_SERVER', 'database-1.c61guiwoghcg.us-east-1.rds.amazonaws.com');
define('DB_USERNAME', 'Ec2_user');
define('DB_PASSWORD', 'cprime2025');
define('DB_DATABASE', 'sample');
?>
[ec2-user@ip-172-31-23-86 inc]$
```

Step 14 : Save and close the dbinfo.inc file. If you are using nano, save and close the file by using Ctrl+S and Ctrl+X.

Step 15 : Change the directory to /var/www/html.

Step 16 : >SamplePage.php

nano SamplePage.php

Step 17 : Add the following contents to the SamplePage.php file:

Sample page

/* Connect to MySQL and select the database. */ $connection = mysqli_connect(DB_SERVER, DB_USERNAME, DB_PASSWORD);

if (mysqli_connect_errno()) echo "Failed to connect to MySQL: " . mysqli_connect_error();

$database = mysqli_select_db($connection, DB_DATABASE);

/* Ensure that the EMPLOYEES table exists. */ VerifyEmployeesTable($connection, DB_DATABASE);

/* If input fields are populated, add a row to the EMPLOYEES table. */ $employee_name = htmlentities($_POST['NAME']); $employee_address = htmlentities($_POST['ADDRESS']);

if (strlen($employee_name) || strlen($employee_address)) { AddEmployee($connection, $employee_name, $employee_address); } ?>

NAME ADDRESS

"; echo "", "", ""; echo ""; } ?>

ID	NAME	ADDRESS
",$query_data[0], "	",$query_data[1], "	",$query_data[2], "

Error adding employee data.

"); } /* Check whether the table exists and, if not, create it. */ function VerifyEmployeesTable($connection, $dbName) { if(!TableExists("EMPLOYEES", $connection, $dbName)) { $query = "CREATE TABLE EMPLOYEES (ID int(11) UNSIGNED

AUTO_INCREMENT PRIMARY KEY, NAME VARCHAR(45), ADDRESS VARCHAR(90))";

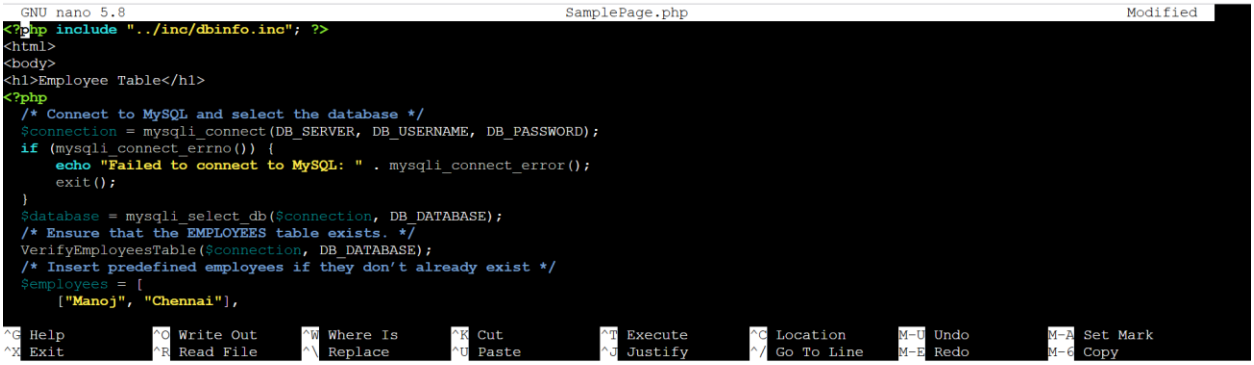
if(!mysqli_query($connection, $query)) echo("

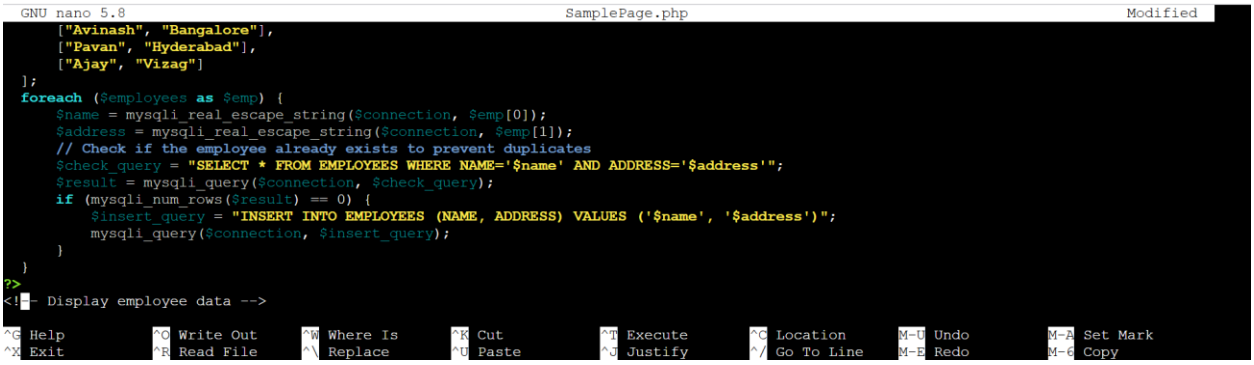
Error creating table.

"); } } /* Check for the existence of a table. */ function TableExists($tableName, $connection, $dbName) { $t = mysqli_real_escape_string($connection, $tableName); $d = mysqli_real_escape_string($connection, $dbName); $checktable = mysqli_query($connection, "SELECT TABLE_NAME FROM information_schema.TABLES WHERE TABLE_NAME = '$t' AND TABLE_SCHEMA = '$d'");

if(mysqli_num_rows($checktable) > 0) return true; return false; } ?>

Step 18 : Save and close the SamplePage.php file.





Step 19 : Verify that your web server successfully connects to your DB instance by opening a web browser and browsing to http://*EC2 instance endpoint*/SamplePage.php

Sample page

NAME ADDRESS

Add Data

ID	NAME	ADDRESS

Sample page

NAME ADDRESS

Add Data

ID	NAME	ADDRESS
1	Bhargav	chennai
2	bhavadeep	hyderabad
3	deva	mumbai
4	manoj	delhi