

**Creating a simple PHP application hosted on EC2 instance
to collect and manage user details, with data stored in an
RDS MySQL database and periodic database backups
uploaded to an S3 bucket**

1. Launch a Red Hat Enterprise Linux 10 t2.micro instance in the default VPC with subnet ap-south-1a, enable auto-assign public IP, create a key pair, and configure a security group allowing SSH (22), HTTP (80), and MySQL/Aurora (3306), then launch the instance.

The screenshot displays the AWS Management Console interface for the EC2 service. The top navigation bar includes the AWS logo, a search bar, and various service icons. The left sidebar shows the 'EC2' menu with options like Dashboard, EC2 Global View, Events, Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, and Capacity Reservations. The main content area is titled 'Instances (1/1)' and shows a table with one instance, 'web_server', which is in a 'Running' state. Below the table, the 'Details' tab for instance 'i-00e179e5734f2c61e' is selected, showing various attributes such as Instance ID, IPv4 address, Instance state, Hostname type, Answer private resource DNS name, Auto-assigned IP address, Public IPv4 address, Private IPv4 addresses, Public DNS, Private IP DNS name (IPv4 only), Instance type, VPC ID, and Elastic IP addresses.

Instances (1/1)

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic
web_server	i-00e179e5734f2c61e	Running	t2.micro	2/2 checks passed	View alarms +	ap-south-1a	ec2-65-2-38-250.ap-so...	65.2.38.250	-

i-00e179e5734f2c61e (web_server)

Instance summary

- Instance ID**: i-00e179e5734f2c61e
- IPv4 address**: -
- Hostname type**: IP name: ip-172-31-42-150.ap-south-1.compute.internal
- Answer private resource DNS name**: -
- Auto-assigned IP address**: 65.2.38.250 [Public IP]
- Public IPv4 address**: 65.2.38.250 | open address
- Instance state**: Running
- Private IP DNS name (IPv4 only)**: ip-172-31-42-150.ap-south-1.compute.internal
- Instance type**: t2.micro
- VPC ID**: vpc-099e47a9d31b97ddc
- Private IPv4 addresses**: 172.31.42.150
- Public DNS**: ec2-65-2-38-250.ap-south-1.compute.amazonaws.com | open address
- Elastic IP addresses**: -
- AWS Compute Optimizer finding**: Opt-in to AWS Compute Optimizer for recommendations. | Learn more

B. Configure a security group allowing SSH (22), HTTP (80), HTTPS (443) and MySQL/Aurora (3306)

EC2 > Security Groups > sg-0b6bb78ad736b9d85 - my-security-new > Edit inbound rules

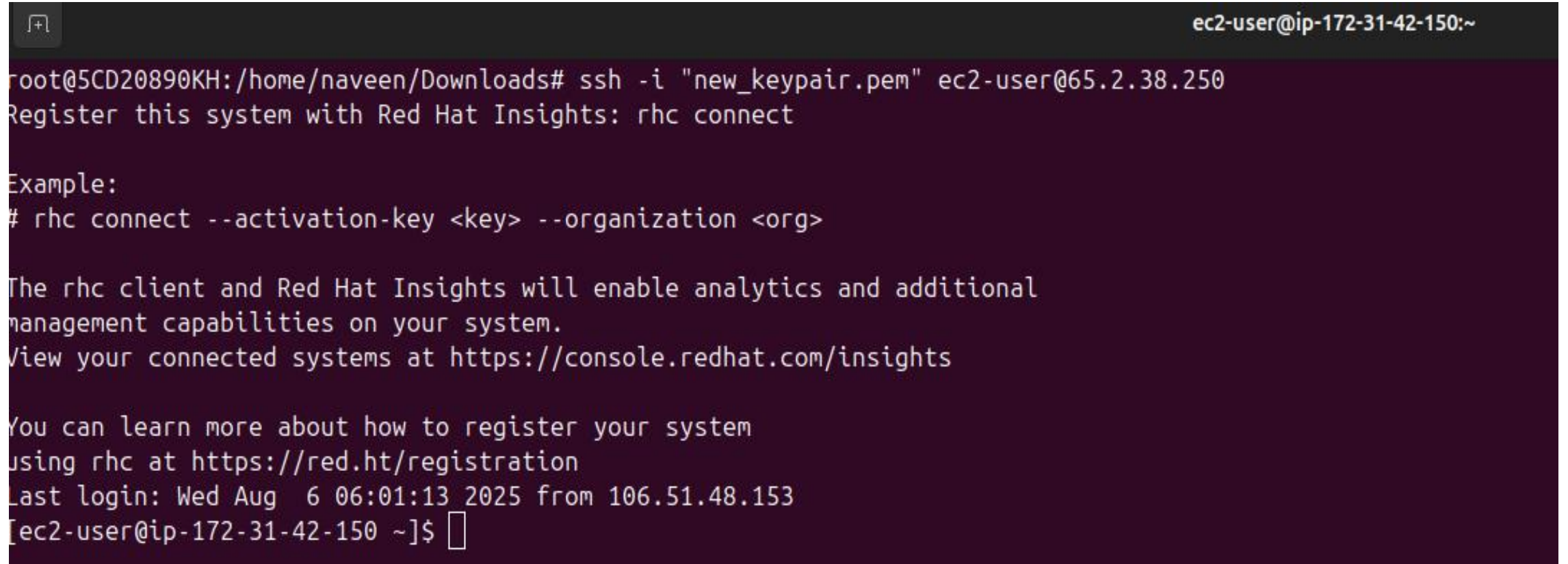
Edit inbound rules Info

Inbound rules control the incoming traffic that's allowed to reach the instance.

Security group rule ID	Type <small>Info</small>	Protocol <small>Info</small>	Port range <small>Info</small>	Source <small>Info</small>	Description - optional <small>Info</small>	
sgr-0854d287441afca20	HTTP	TCP	80	Custom	<input type="text" value="106.51.48.153/32"/>	<input type="button" value="Delete"/>
sgr-0a1dbbc17f3d6e690	MySQL/Aurora	TCP	3306	Custom	<input type="text" value="172.31.42.150/32"/>	<input type="button" value="Delete"/>
sgr-03fa5a8e1e2d12d52	SSH	TCP	22	Custom	<input type="text" value="106.51.48.153/32"/>	<input type="button" value="Delete"/>
sgr-09823e35940480ebf	HTTPS	TCP	443	Custom	<input type="text" value="106.51.48.153/32"/>	<input type="button" value="Delete"/>

2. After the launch, connect your instance using the downloaded keypair

ssh -i "<keypair>" ec2-user@ip-address



```
ec2-user@ip-172-31-42-150:~  
root@5CD20890KH:/home/naveen/Downloads# ssh -i "new_keypair.pem" ec2-user@65.2.38.250  
Register this system with Red Hat Insights: rhc connect  
  
Example:  
# rhc connect --activation-key <key> --organization <org>  
  
The rhc client and Red Hat Insights will enable analytics and additional  
management capabilities on your system.  
View your connected systems at https://console.redhat.com/insights  
  
You can learn more about how to register your system  
using rhc at https://red.ht/registration  
Last login: Wed Aug 6 06:01:13 2025 from 106.51.48.153  
[ec2-user@ip-172-31-42-150 ~]$
```

3. Patch update the system and install apache service.

sudo yum update -y
sudo yum install httpd -y

```
[root@ip-172-31-42-150 ec2-user]# yum install httpd -y
Updating Subscription Management repositories.
Unable to read consumer identity

This system is not registered with an entitlement server. You can use "rhc" or "subscription-manager" to register.

Last metadata expiration check: 0:00:18 ago on Wed Aug  6 06:12:32 2025.
Dependencies resolved.
=====
Package                                Architecture      Version            Repository          Size
=====
Installing:
httpd                                  x86_64            2.4.63-1.el10      rhel-10-appstream-rhui-rpms 53 k
Installing dependencies:
apr                                    x86_64            1.7.5-2.el10       rhel-10-appstream-rhui-rpms 132 k
apr-util                              x86_64            1.6.3-21.el10      rhel-10-appstream-rhui-rpms 101 k
apr-util-ldb                          x86_64            1.6.3-21.el10      rhel-10-appstream-rhui-rpms 16 k
httpd-core                            x86_64            2.4.63-1.el10      rhel-10-appstream-rhui-rpms 1.5 M
httpd-filesystem                      noarch            2.4.63-1.el10      rhel-10-appstream-rhui-rpms 17 k
httpd-tools                           x86_64            2.4.63-1.el10      rhel-10-appstream-rhui-rpms 87 k
mailcap                               noarch            2.1.54-8.el10      rhel-10-baseos-rhui-rpms 37 k
redhat-logos-httpd                   noarch            100.1-1.el10_0     rhel-10-appstream-rhui-rpms 17 k
Installing weak dependencies:
apr-util-openssl                      x86_64            1.6.3-21.el10      rhel-10-appstream-rhui-rpms 18 k
mod_http2                             x86_64            2.0.29-2.el10      rhel-10-appstream-rhui-rpms 169 k
mod_lua                               x86_64            2.4.63-1.el10      rhel-10-appstream-rhui-rpms 61 k
Transaction Summary
=====
Install 12 Packages

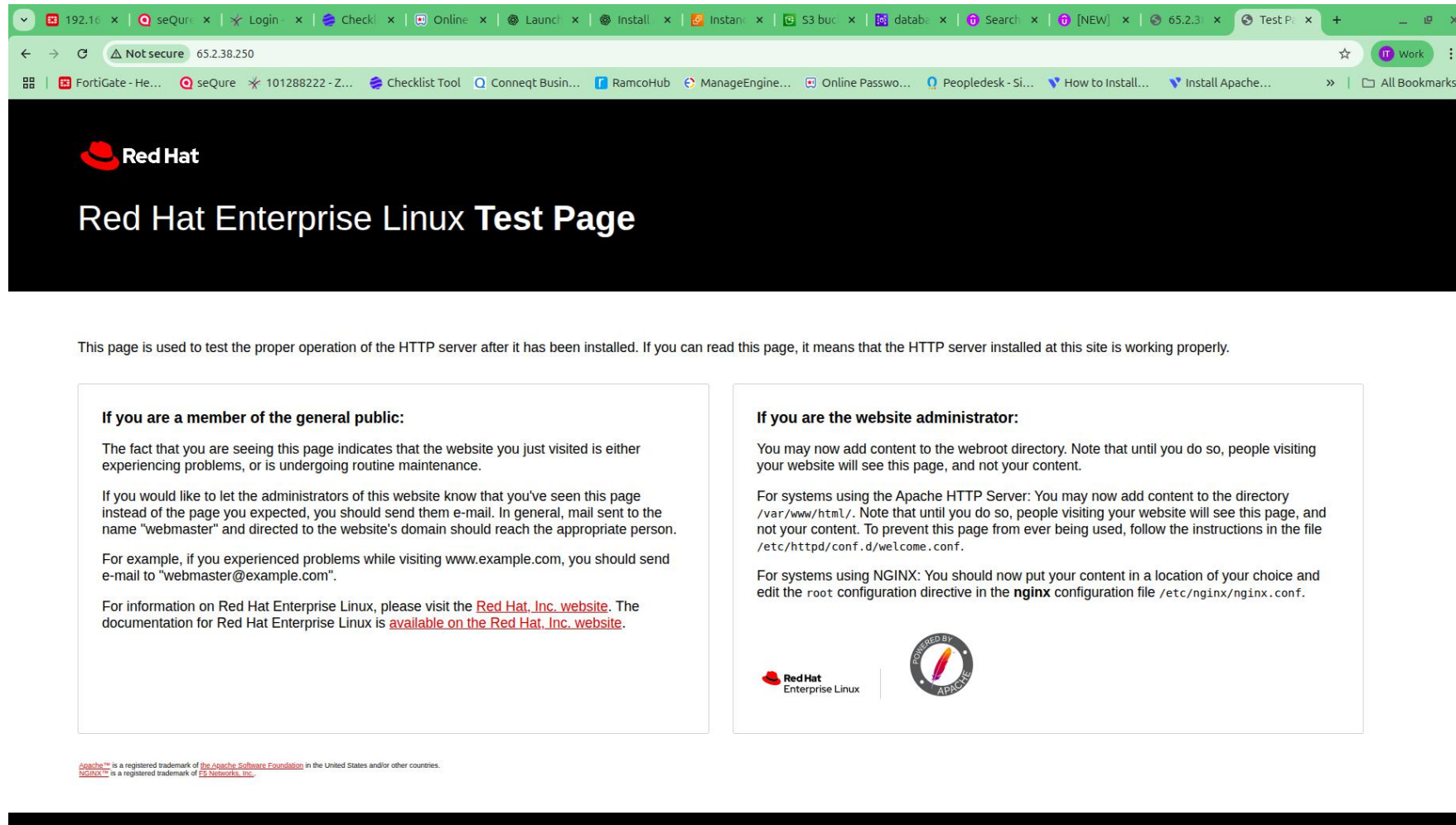
Total download size: 2.2 M
Installed size: 6.1 M
Downloading Packages:
(1/12): apr-util-ldb-1.6.3-21.el10.x86_64.rpm 338 kB/s | 16 kB 00:00
(2/12): httpd-tools-2.4.63-1.el10.x86_64.rpm 1.6 MB/s | 87 kB 00:00
(3/12): mod_http2-2.0.29-2.el10.x86_64.rpm 2.9 MB/s | 169 kB 00:00
(4/12): apr-util-1.6.3-21.el10.x86_64.rpm 8.1 MB/s | 101 kB 00:00
(5/12): mod_lua-2.4.63-1.el10.x86_64.rpm 8.2 MB/s | 61 kB 00:00
```


4. Start the apache services by entering the below commands

```
sudo systemctl start httpd  
sudo systemctl status httpd  
sudo systemctl enable httpd
```

```
[root@ip-172-31-42-150 ec2-user]# systemctl start httpd  
[root@ip-172-31-42-150 ec2-user]# systemctl status httpd  
● httpd.service - The Apache HTTP Server  
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; preset: disabled)  
   Active: active (running) since Wed 2025-08-06 06:13:50 UTC; 1s ago  
 Invocation: 0de282f801ba4b7a81104dac19f13936  
    Docs: man:httpd.service(8)  
 Main PID: 12719 (httpd)  
   Status: "Started, listening on: port 80"  
    Tasks: 177 (limit: 5687)  
  Memory: 13.7M (peak: 13.9M)  
     CPU: 93ms  
   CGroup: /system.slice/httpd.service  
           └─12719 /usr/sbin/httpd -DFOREGROUND  
             └─12720 /usr/sbin/httpd -DFOREGROUND  
               └─12721 /usr/sbin/httpd -DFOREGROUND  
                 └─12722 /usr/sbin/httpd -DFOREGROUND  
                   └─12759 /usr/sbin/httpd -DFOREGROUND  
  
Aug 06 06:13:50 ip-172-31-42-150.ap-south-1.compute.internal systemd[1]: Starting httpd.service - The Apache HTTP Server...  
Aug 06 06:13:50 ip-172-31-42-150.ap-south-1.compute.internal (httpd)[12719]: httpd.service: Referenced but unset environment variable evaluates to an empty string: OPTIONS  
Aug 06 06:13:50 ip-172-31-42-150.ap-south-1.compute.internal systemd[1]: Started httpd.service - The Apache HTTP Server.  
Aug 06 06:13:50 ip-172-31-42-150.ap-south-1.compute.internal httpd[12719]: Server configured, listening on: port 80  
[root@ip-172-31-42-150 ec2-user]# ^C  
[root@ip-172-31-42-150 ec2-user]#  
[root@ip-172-31-42-150 ec2-user]#  
[root@ip-172-31-42-150 ec2-user]#  
[root@ip-172-31-42-150 ec2-user]#  
[root@ip-172-31-42-150 ec2-user]# systemctl enable httpd  
Created symlink '/etc/systemd/system/multi-user.target.wants/httpd.service' → '/usr/lib/systemd/system/httpd.service'.  
[root@ip-172-31-42-150 ec2-user]#
```

5. Check the apache index page using the URL **http://<ec2-ipaddress>**



6. In the RDS Console, create a MySQL database using the standard method. Choose **db.t4g.micro** (Free Tier), General Purpose (SSD) storage, and Single-AZ. Set a master username and password, disable public access, attach a VPC security group allowing port **3306** from the EC2 security group, disable automated backups, and launch the instance.

The screenshot shows the Amazon RDS console interface. On the left is a navigation sidebar with the following links: Aurora and RDS, Databases, Dashboard, Query editor, Performance Insights, Snapshots, Exports In Amazon S3, Automated backups, Reserved Instances, and Proxies. The main content area is titled 'Databases (1)' and includes a search bar with the placeholder text 'Filter by databases'. Below the search bar is a table listing database instances. The table has columns for DB identifier, Status, Role, Engine, Region, Size, Recommendations, and CPU. A single instance is listed with the identifier 'database-new', status 'Available', role 'Instance', engine 'MySQL Co...', region 'ap-south-1a', size 'db.t4g.micro', and 2 informational recommendations. Above the table are buttons for 'Group resources', 'Modify', 'Actions', and 'Create database'. The 'Create database' button is highlighted in orange.

DB identifier	Status	Role	Engine	Region	Size	Recommendations	CPU
database-new	Available	Instance	MySQL Co...	ap-south-1a	db.t4g.micro	2 Informational	3.0

7. Now copy the **Endpoint** from the RDS instance to establish the database connection from EC2 instance.

The screenshot displays the AWS Management Console for an RDS instance named 'database-new'. The left sidebar shows the navigation menu with 'Aurora and RDS' selected. The main content area shows the instance details under the 'Connectivity & security' tab. The 'Endpoint' is highlighted with a red box.

Aurora and RDS

- Dashboard
- Databases**
- Query editor
- Performance Insights
- Snapshots
- Exports in Amazon S3
- Automated backups
- Reserved Instances
- Proxies

database-new

Summary

DB identifier database-new	Status Available	Role Instance	Engine MySQL Community	Recommendations 2 Informational
CPU 3.66%	Class db.t4g.micro	Current activity 2 Connections	Region & AZ ap-south-1a	

Connectivity & security

Endpoint & port

Endpoint
database-new.ctiwwk22jli.ap-south-1.rds.amazonaws.com

Port
3306

Networking

Availability Zone
ap-south-1a

VPC
vpc-099e47a9d31b97ddc

Subnet group
default-vpc-099e47a9d31b97ddc

Subnets
subnet-05583ad33c6702230
subnet-01ea2efaf7e5691e7
subnet-0975fed6cbf3fd7f0

Network type
IPv4

Security

VPC security groups
my-security-new (sg-0b6bb78ad736b9d85)
Active

Publicly accessible
No

Certificate authority Info
rds-ca-rsa2048-g1

Certificate authority date
May 20, 2021, 00:10 (UTC+05:30)

DB instance certificate expiration date
August 06, 2026, 11:44 (UTC+05:30)

8. Now download and install the MySQL community in the RedHat server.

a. Download the MySQL Yum Repository RPM

`sudo dnf install -y https://dev.mysql.com/get/mysql80-community-release-el9-1.noarch.rpm`

b. Enable the repository

`sudo dnf repolist enabled | grep mysql`

```
[root@ip-172-31-42-150 ec2-user]# sudo dnf install -y https://dev.mysql.com/get/mysql80-community-release-el9-1.noarch.rpm
Updating Subscription Management repositories.
Unable to read consumer identity

This system is not registered with an entitlement server. You can use "rhc" or "subscription-manager" to register.

Last metadata expiration check: 0:00:34 ago on Wed Aug  6 06:21:20 2025.
mysql80-community-release-el9-1.noarch.rpm                                31 kB/s | 10 kB      00:00
Dependencies resolved.
=====
Package                                Architecture      Version           Repository        Size
=====
Installing:
mysql80-community-release              noarch            el9-1             @commandline      10 k
=====
Transaction Summary
=====
Install 1 Package

Total size: 10 k
Installed size: 5.7 k
Downloading Packages:
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing      :                                1/1
  Installing     : mysql80-community-release-el9-1.noarch 1/1
Installed products updated.

Installed:
mysql80-community-release-el9-1.noarch

Complete!
[root@ip-172-31-42-150 ec2-user]# sudo dnf repolist enabled | grep mysql
mysql-connectors-community    MySQL Connectors Community
mysql-tools-community         MySQL Tools Community
mysql80-community             MySQL 8.0 Community Server
```

9. After the download import the GPG key and clean the DNF cache (recommended)

a. Import the new GPG key

sudo rpm --import https://repo.mysql.com/RPM-GPG-KEY-mysql-2023

b. Clean the DNF cache & install it

sudo dnf clean packages

sudo dnf install -y mysql-community-client

```
[root@ip-172-31-42-150 ec2-user]# sudo rpm --import https://repo.mysql.com/RPM-GPG-KEY-mysql-2023
[root@ip-172-31-42-150 ec2-user]# sudo rpm -q gpg-pubkey --qf '%{NAME}-%{VERSION}-%{RELEASE}\n' | grep 3a79bd29
gpg-pubkey-3a79bd29-61b8bab7
[root@ip-172-31-42-150 ec2-user]# sudo rpm -e gpg-pubkey-3a79bd29-*
[root@ip-172-31-42-150 ec2-user]# sudo rpm --import https://repo.mysql.com/RPM-GPG-KEY-mysql-2023
[root@ip-172-31-42-150 ec2-user]# sudo dnf clean packages
Updating Subscription Management repositories.
Unable to read consumer identity
```

This system is not registered with an entitlement server. You can use "rhc" or "subscription-manager" to register.

4 files removed

```
[root@ip-172-31-42-150 ec2-user]# sudo dnf install -y mysql-community-client
Updating Subscription Management repositories.
Unable to read consumer identity
```

This system is not registered with an entitlement server. You can use "rhc" or "subscription-manager" to register.

Last metadata expiration check: 0:01:32 ago on Wed Aug 6 06:22:10 2025.
Dependencies resolved.

Package	Architecture	Version	Repository
Installing:			
mysql-community-client	x86_64	8.0.43-1.el9	mysql80-community
Installing dependencies:			
mysql-community-client-plugins	x86_64	8.0.43-1.el9	mysql80-community
mysql-community-common	x86_64	8.0.43-1.el9	mysql80-community
mysql-community-libs	x86_64	8.0.43-1.el9	mysql80-community

Transaction Summary

Install 4 Packages

Total download size: 6.7 M

Installed size: 96 M

Downloading Packages:

(1/4): mysql-community-client-plugins-8.0.43-1.el9.x86_64.rpm	60 MB/s 1.4 MB	00:
(2/4): mysql-community-libs-8.0.43-1.el9.x86_64.rpm	89 MB/s 1.5 MB	00:
(3/4): mysql-community-common-8.0.43-1.el9.x86_64.rpm	12 MB/s 556 kB	00:
(4/4): mysql-community-client-8.0.43-1.el9.x86_64.rpm	36 MB/s 3.3 MB	00:

10. Now install the PHP modules to connect with mysql database.

```
sudo yum install -y php php-mysqlnd  
php -m | grep mysqli
```

```
[root@ip-172-31-42-150 ec2-user]# sudo yum install -y php php-mysqlnd
Updating Subscription Management repositories.
Unable to read consumer identity

This system is not registered with an entitlement server. You can use "rhc" or "subscription-manager" to register.

Last metadata expiration check: 0:02:06 ago on Wed Aug  6 06:22:10 2025.
Dependencies resolved.
=====
Package                                Architecture          Version                Repository              Size
=====
Installing:
php                                     x86_64                8.3.19-1.el10_0        rhel-10-appstream-rhui-rpms 8.1 k
php-mysqlnd                           x86_64                8.3.19-1.el10_0        rhel-10-appstream-rhui-rpms 142 k
Installing dependencies:
capstone                               x86_64                5.0.1-6.el10           rhel-10-appstream-rhui-rpms 1.0 M
libxslt                                x86_64                1.1.39-8.el10_0        rhel-10-appstream-rhui-rpms 190 k
nginxfilesystem                        noarch                2:1.26.3-1.el10        rhel-10-appstream-rhui-rpms 14 k
php-common                             x86_64                8.3.19-1.el10_0        rhel-10-appstream-rhui-rpms 717 k
php-pdo                                x86_64                8.3.19-1.el10_0        rhel-10-appstream-rhui-rpms 89 k
Installing weak dependencies:
php-cli                                x86_64                8.3.19-1.el10_0        rhel-10-appstream-rhui-rpms 3.6 M
php-fpm                                x86_64                8.3.19-1.el10_0        rhel-10-appstream-rhui-rpms 1.9 M
php-mbstring                           x86_64                8.3.19-1.el10_0        rhel-10-appstream-rhui-rpms 522 k
php-opcache                             x86_64                8.3.19-1.el10_0        rhel-10-appstream-rhui-rpms 368 k
php-xml                                 x86_64                8.3.19-1.el10_0        rhel-10-appstream-rhui-rpms 151 k
=====
Transaction Summary
=====
Install 12 Packages

Total download size: 8.6 M
Installed size: 51 M
Downloading Packages:
(1/12): nginx-filessystem-1.26.3-1.el10.noarch.rpm                292 kB/s | 14 kB    00:00
(2/12): php-8.3.19-1.el10_0.x86_64.rpm                          162 kB/s | 8.1 kB   00:00
(3/12): capstone-5.0.1-6.el10.x86_64.rpm                         15 MB/s | 1.0 MB    00:00
(4/12): php-common-8.3.19-1.el10_0.x86_64.rpm                   24 MB/s | 717 kB    00:00
(5/12): php-cli-8.3.19-1.el10_0.x86_64.rpm                       51 MB/s | 3.6 MB    00:00
(6/12): php-fpm-8.3.19-1.el10_0.x86_64.rpm                       27 MB/s | 1.9 MB    00:00
(7/12): php-mbstring-8.3.19-1.el10_0.x86_64.rpm                  8.6 MB/s | 522 kB   00:00
(8/12): php-mysqlnd-8.3.19-1.el10_0.x86_64.rpm                   5.7 MB/s | 142 kB   00:00
(9/12): php-opcache-8.3.19-1.el10_0.x86_64.rpm                  24 MB/s | 368 kB    00:00
(10/12): php-xml-8.3.19-1.el10_0.x86_64.rpm                     14 MB/s | 151 kB    00:00
```


11. Now edit the PHP script as mentioned below to generate a user input web page.

sudo nano /var/www/html/form.php

```
<?php
ini_set('display_errors', 1);
ini_set('display_startup_errors', 1);
error_reporting(E_ALL);
?>

<!DOCTYPE html>
<html>
<body>
<h2>User Signup</h2>
<form method="POST" action="form.php">
    Name: <input type="text" name="name"><br><br>
    Email: <input type="email" name="email"><br><br>
    <input type="submit" value="Submit">
</form>
</body>
</html>

<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    echo "<br>Form submitted.<br>";

    // Replace these with your actual credentials
    $conn = new mysqli("database-new.ctiwiwk22jji.ap-south-1.rds.amazonaws.com", "admin", "Digitide1234", "myapp");

    if ($conn->connect_error) {
        die("Connection failed: " . $conn->connect_error);
    } else {
        echo "Connected to database.<br>";
    }

    $name = $_POST['name'];
    $email = $_POST['email'];

    echo "Received name: $name, email: $email<br>";

    $stmt = $conn->prepare("INSERT INTO users (name, email) VALUES (?, ?)");
    if (!$stmt) {
        die("Prepare failed: " . $conn->error);
    }

    $stmt->bind_param("ss", $name, $email);

    if (!$stmt->execute()) {
        die("Execute failed: " . $stmt->error);
    } else {
        echo "User data saved successfully!";
    }

    $stmt->close();
    $conn->close();
}
?>
```



```
<?php
ini_set('display_errors', 1);
ini_set('display_startup_errors', 1);
error_reporting(E_ALL);
?>
```

```
<!DOCTYPE html>
<html>
<body>
<h2>User Signup</h2>
<form method="POST" action="form.php">
  Name: <input type="text" name="name"><br><br>
  Email: <input type="email" name="email"><br><br>
  <input type="submit" value="Submit">
</form>
</body>
</html>
```

```
<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    echo "<br>Form submitted.<br>";

    // Replace these with your actual credentials
    $conn = new mysqli("your-rds-endpoint", "your-db-user", "your-db-password", "myapp");

    if ($conn->connect_error) {
        die("Connection failed: " . $conn->connect_error);
    } else {
        echo "Connected to database.<br>";
    }

    $name = $_POST['name'];
    $email = $_POST['email'];

    echo "Received name: $name, email: $email<br>";

    $stmt = $conn->prepare("INSERT INTO users (name, email) VALUES (?, ?)");
    if (!$stmt) {
        die("Prepare failed: " . $conn->error);
    }

    $stmt->bind_param("ss", $name, $email);

    if (!$stmt->execute()) {
        die("Execute failed: " . $stmt->error);
    } else {
        echo "User data saved successfully!";
    }

    $stmt->close();
    $conn->close();
}
?>
```

12. Connect to the RDS Mysql database using the endpoint copied from step 7.

mysql -h <RDS-ENDPOINT> -u <DB-USER> -p

```
root@ip-172-31-42-150:/home/ec2-user
[root@ip-172-31-42-150 ec2-user]# mysql -h database-new.ctiwiwk22jji.ap-south-1.rds.amazonaws.com -u admin -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 29
Server version: 8.0.42 Source distribution

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

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

mysql> SHOW databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
+-----+
4 rows in set (0.00 sec)

mysql> CREATE DATABASE myapp;
Query OK, 1 row affected (0.02 sec)

mysql> USE myapp;
Database changed
mysql> CREATE TABLE users (
  ->   id INT AUTO_INCREMENT PRIMARY KEY,
  ->   name VARCHAR(100),
  ->   email VARCHAR(100),
  ->   created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
  -> );
Query OK, 0 rows affected (0.06 sec)

mysql> SHOW tables;
+-----+
| Tables_in_myapp |
+-----+
| users |
+-----+
1 row in set (0.00 sec)
```

List a Database
SHOW database;

Create a Database and Table for User Data
CREATE DATABASE myapp;

Switch to myapp Database
USE myapp;

List the tables
SHOW TABLES;

Create a new table for the user inputs
CREATE TABLE users (
 id INT AUTO INCREMENT PRIMARY KEY,
 name VARCHAR(100),
 email VARCHAR(100),
 created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

To insert a user manually for testing.
INSERT INTO users (name, email) VALUES ('Alice', 'alice@example.com');

To see the user permissions
SHOW GRANTS FOR 'admin'@'%';

To give the user full permissions
GRANT ALL PRIVILEGES ON myapp.* TO 'admin'@'%'
FLUSH PRIVILEGES;

To see the user inputs
SELECT * from users;

```
mysql> SHOW GRANTS FOR 'admin'@'%';
+-----+
| Grants for admin@% |
+-----+
| GRANT USAGE ON *.* TO `admin`@`%` |
| GRANT ALL PRIVILEGES ON `myapp`.* TO `admin`@`%` |
| GRANT `rds_superuser_role`@`%` TO `admin`@`%` |
+-----+
3 rows in set (0.00 sec)
```

13. Now you can access the php web page by entering the URL **http://<ec2-ipaddress>:form.php**

As you can see the image below, now the user input will be stored in the RDS Mysql database

User Signup

Name:

Email:

Form submitted.

Connected to database.

Received name: naveen, email: naveen.raaj@test.com


User data saved successfully!

```
mysql> SELECT * from users;
```

id	name	email	created_at
1	Alice	alice@example.com	2025-08-06 06:32:53
2	naveen	naveen.raaj@test.com	2025-08-06 07:08:05

2 rows in set (0.00 sec)






14. Create **S3 bucket** in the mumbai region with block all public access enabled.


 [Amazon S3](#) > Buckets


General purpose buckets All AWS Regions


Directory buckets

General purpose buckets (1) [Info](#)

  Copy ARN  Empty  Delete 

 Find buckets by name

< 1 > 

Name	AWS Region	Creation date
 tempbucket-cbe	Asia Pacific (Mumbai) ap-south-1	August 6, 2025, 11:28:07 (UTC+05:30)

15. Now create a **ec2-s3-role** in the IAM to connect EC2 instance with S3 bucket for database file backup.

ec2-s3-role [Info](#)

Allows EC2 Instances to call AWS services on your behalf.

Delete

Summary

Creation date
July 15, 2025, 12:56 (UTC+05:30)

Last activity
✔ 20 days ago

ARN
[arn:aws:iam::182399724378:role/ec2-s3-role](#)

Maximum session duration
1 hour

Instance profile ARN
[arn:aws:iam::182399724378:instance-profile/ec2-s3-role](#)

Edit

Permissions

Trust relationships

Tags

Last Accessed

Revoke sessions

Permissions policies (1) [Info](#)

You can attach up to 10 managed policies.

Search

Filter by Type
All types

< 1 > ⚙

☐

Policy name [\[?\]](#)

▲

Type

▼

Attached entities

▼

☐

[AmazonS3FullAccess](#)

AWS managed

1

► Permissions boundary (not set)

▼ Generate policy based on CloudTrail events

You can generate a new policy based on the access activity for this role, then customize, create, and attach it to this role. AWS uses your CloudTrail events to identify the services and actions used and generate a policy. [Learn more \[?\]](#)

Generate policy


16. Now attach the created ec2-s3-role in ec2 instance by clicking on **Actions > Security > Modify IAM role**.

EC2 > Instances > i-0d5c2393649f95b88 > Modify IAM role

Modify IAM role [Info](#)

Attach an IAM role to your instance.

Instance ID


 i-0d5c2393649f95b88 (web_server)

IAM role

Select an IAM role to attach to your instance or create a new role if you haven't created any. The role you select replaces any roles that are currently attached to your instance.

ec2-s3-role

▼

 Create new IAM role [↗](#)

Cancel

Update IAM role

17. Install aws-cli in the ec2 server to connect the S3 bucket in terminal.

sudo yum install -y awscli

```
[root@ip-172-31-42-150 bin]# sudo yum install -y awscli
Updating Subscription Management repositories.
Unable to read consumer identity

This system is not registered with an entitlement server. You can use "rhc" or "subscription-manager" to register.

Last metadata expiration check: 0:42:23 ago on Wed Aug  6 07:32:21 2025.
Dependencies resolved.
=====
Package                                Architecture      Version           Repository        Size
=====
Installing:
awscli2                                noarch            2.27.0-1.el10_0   rhel-10-appstream-rhui-rpms 15 M
Installing dependencies:
python3-awscrt                         x86_64            0.27.2-1.el10_0   rhel-10-appstream-rhui-rpms 1.0 M
python3-cffi                           x86_64            1.16.0-7.el10     rhel-10-baseos-rhui-rpms 312 k
python3-colorama                       noarch            0.4.6-13.el10     rhel-10-appstream-rhui-rpms 76 k
python3-cryptography                   x86_64            43.0.0-4.el10     rhel-10-baseos-rhui-rpms 1.4 M
python3-docutils                       noarch            0.20.1-5.el10     rhel-10-appstream-rhui-rpms 1.1 M
python3-jmespath                       noarch            1.0.1-8.el10      rhel-10-appstream-rhui-rpms 62 k
python3-ply                            noarch            3.11-25.el10      rhel-10-baseos-rhui-rpms 139 k
python3-prompt-toolkit                 noarch            3.0.41-5.el10     rhel-10-appstream-rhui-rpms 879 k
python3-pycparser                      noarch            2.20-16.el10      rhel-10-baseos-rhui-rpms 162 k
python3-ruamel-yaml                    noarch            0.18.5-5.el10     rhel-10-appstream-rhui-rpms 309 k
python3-ruamel-yaml-clib               x86_64            0.2.7-8.el10      rhel-10-appstream-rhui-rpms 154 k
python3-wcwidth                        noarch            0.2.6-6.el10      rhel-10-baseos-rhui-rpms 50 k
Installing weak dependencies:
groff                                  x86_64            1.23.0-10.el10    rhel-10-appstream-rhui-rpms 1.4 M
=====
```

18. Check the aws-cli version by the following command **aws --version**

Installed:

awscli2-2.27.0-1.el10_0.noarch	groff-1.23.0-10.el10.x86_64	python3-awscrt-0.27.2-1.el10_0.x86_64	python3-cffi-1.16.0-7.el10.x86_64
python3-colorama-0.4.6-13.el10.noarch	python3-cryptography-43.0.0-4.el10.x86_64	python3-docutils-0.20.1-5.el10.noarch	python3-jmespath-1.0.1-8.el10.noarch
python3-ply-3.11-25.el10.noarch	python3-prompt-toolkit-3.0.41-5.el10.noarch	python3-pycparser-2.20-16.el10.noarch	python3-ruamel-yaml-0.18.5-5.el10.noarch
python3-ruamel-yaml-clib-0.2.7-8.el10.x86_64	python3-wcwidth-0.2.6-6.el10.noarch		

Complete!

```
[root@ip-172-31-42-150 bin]# aws --version
```

```
aws-cli/2.27.0 Python/3.12.9 Linux/6.12.0-55.18.1.el10_0.x86_64 source/x86_64.rhel.10
```

19. Now create mysql dump script **db_backup_s3.sh** to take the .sql backup from the RDS and move it to the S3 bucket.

```
#!/bin/bash

# Variables
DB_HOST="database-new.ctiwiwk22jji.ap-south-1.rds.amazonaws.com"
DB_USER="admin"
DB_PASS="Digitide1234"
DB_NAME="myapp"
S3_BUCKET="tempbucket-cbe"
TIMESTAMP=$(date +%F-%H-%M)
DUMP_FILE="/tmp/user_data_${TIMESTAMP}.sql"

# Dump user data with safer and consistent options
mysqldump \
  --single-transaction \
  --set-gtid-purged=OFF \
  -h "$DB_HOST" -u"$DB_USER" -p"$DB_PASS" "$DB_NAME" users > "$DUMP_FILE"

# Upload to S3
aws s3 cp "$DUMP_FILE" s3://"${S3_BUCKET}/

# Clean up
rm -f "$DUMP_FILE"
```

```
[root@ip-172-31-42-150 bin]# ./db_backup_s3.sh
mysqldump: [Warning] Using a password on the command line interface can be insecure.
upload: ../../../../tmp/user_data_2025-08-06-08-25.sql to s3://tempbucket-cbe/user_data_2025-08-06-08-25.sql
```



```
#!/bin/bash
```

```
# Variables
```

```
DB_HOST="your-rds-endpoint"
```

```
DB_USER="your-db-user"
```

```
DB_PASS="your-db-password"
```

```
DB_NAME="myapp"
```

```
S3_BUCKET="my-app-db-backups"
```

```
TIMESTAMP=$(date +%F-%H-%M)
```

```
DUMP_FILE="/tmp/user_data_${TIMESTAMP}.sql"
```

```
# Dump user data
```

```
mysqldump -h $DB_HOST -u$DB_USER -p$DB_PASS $DB_NAME users > $DUMP_FILE
```

```
# Upload to S3
```

```
aws s3 cp $DUMP_FILE s3://$S3_BUCKET/
```

```
# Clean up
```

```
rm -f $DUMP_FILE
```

Make this backup script as a executable file

chmod +x /usr/local/bin/db_backup_s3.sh

Schedule backup every day at 2AM using cronjob

crontab -e

echo "0 2 * * * root /usr/local/bin/db_backup_s3.sh"

View the S3 bucket in the terminal

aws s3 ls s3://tempbucket-cbe/

20. Verify the backup file in the S3 bucket in aws web console.

Amazon S3 > Buckets > tempbucket-cbe

tempbucket-cbe

Objects | Properties | Permissions | Metrics | Management | Access Points

Objects (1)

Copy S3 URI | Copy URL | Download | Open | Delete | Actions | Create folder | Upload

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

Find objects by prefix

< 1 > ⚙

<input type="checkbox"/>	Name ▲	Type	Last modified ▼	Size ▼	Storage class ▼
<input type="checkbox"/>	user_data_2025-08-06-08-25.sql	sql	August 6, 2025, 13:55:40 (UTC+05:30)	2.3 KB	Standard

Thank You