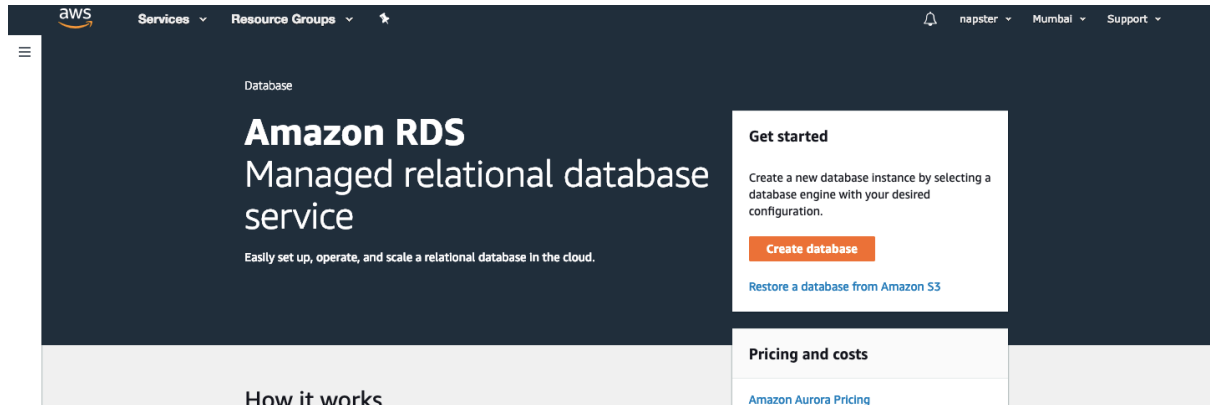


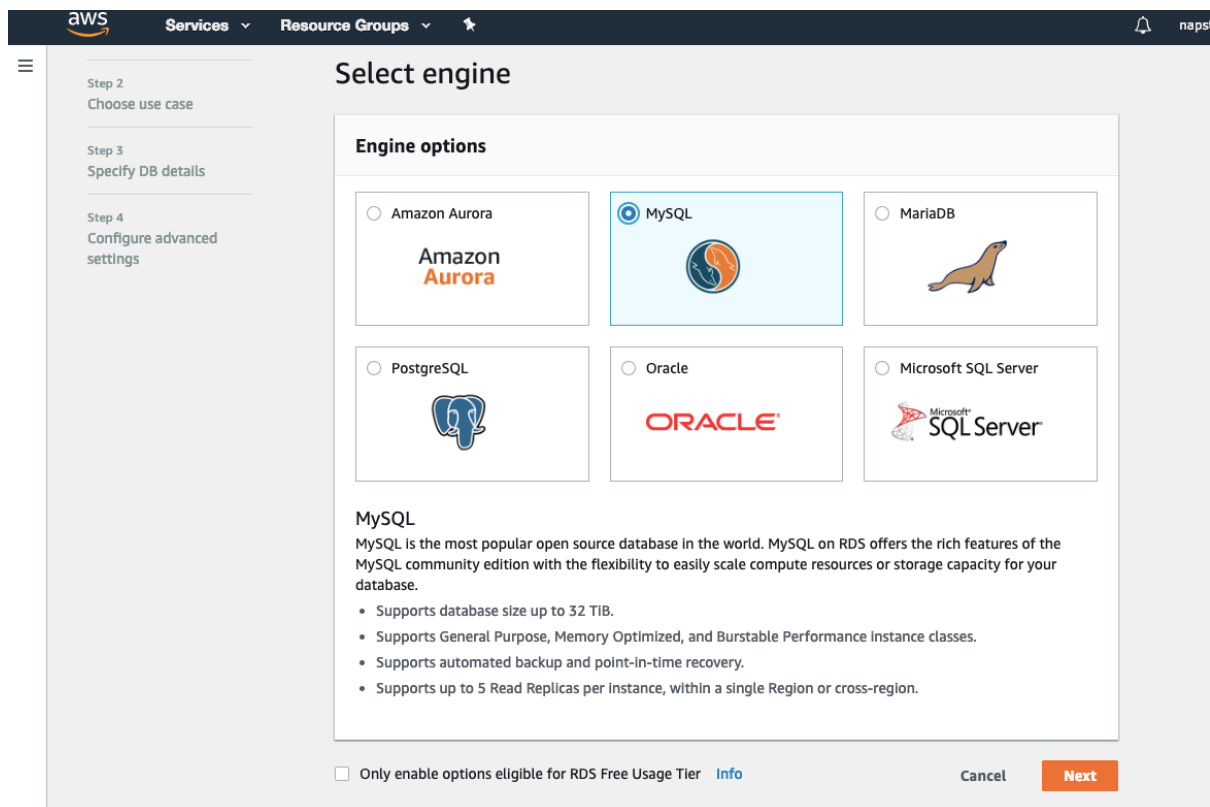
Task: RDS Lab

Launching an RDS Instance and connecting it with EC2 Instance

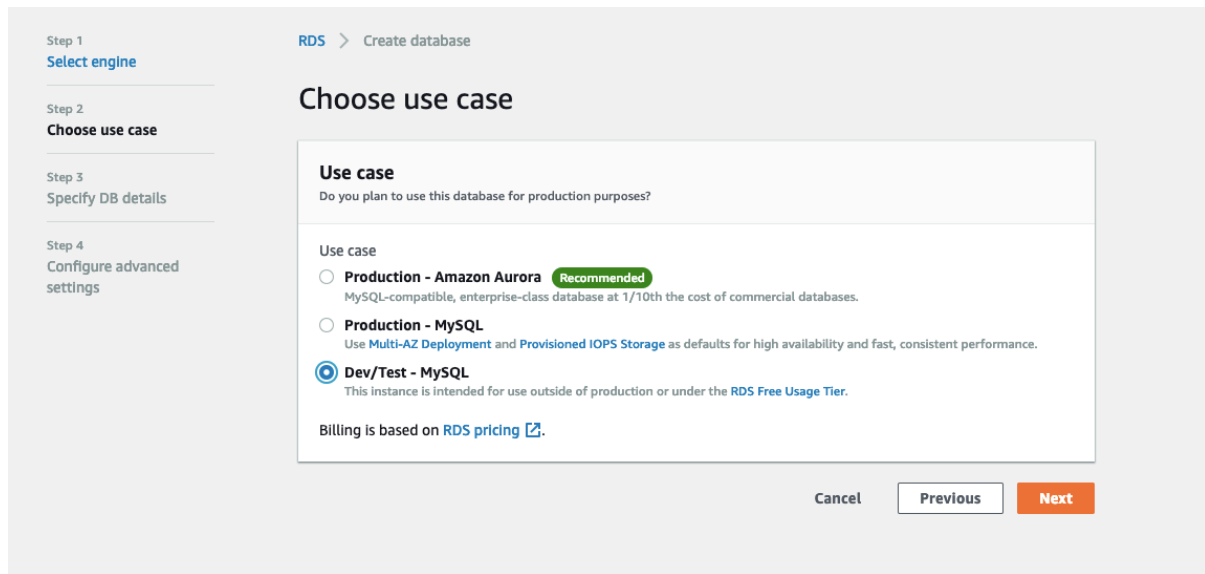
Step 1 : Log in to RDS Console



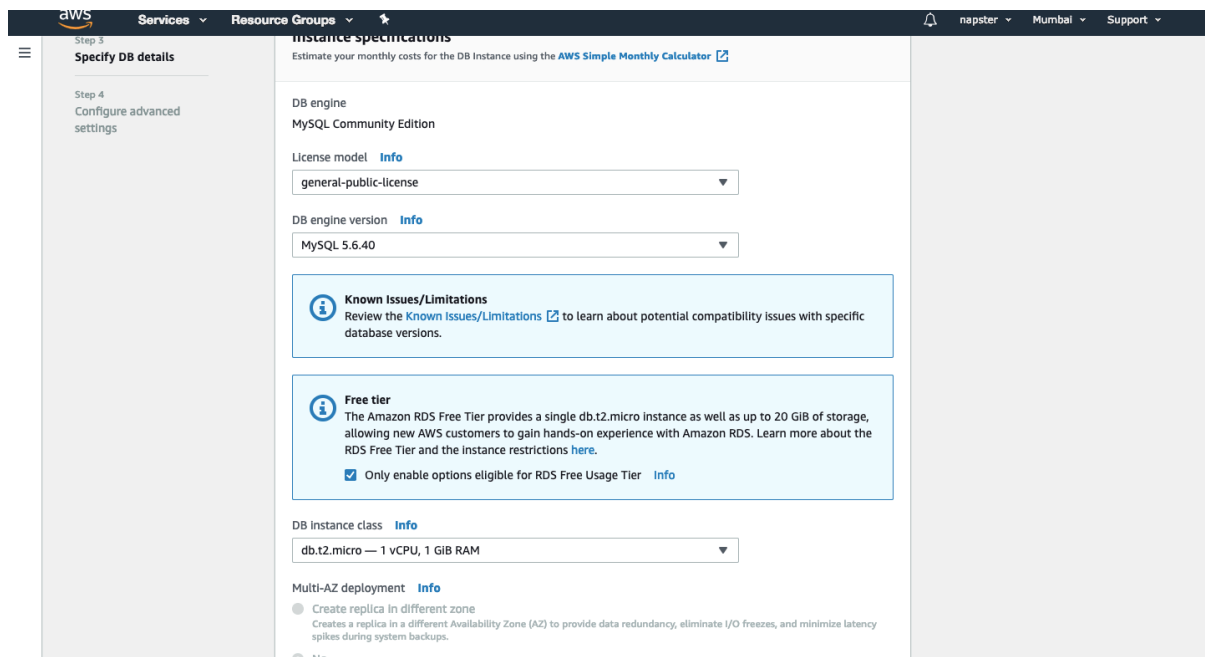
Step 2 : Click on “Create Database” and select one of the database type. It is recommended to chose MySQL, MariaDB or PostgreSQL if you’re in Free Tier. In this lab session we will choose, MySQL.



Step 3 : Click on next and select “Dev/Test Environment” and click on next



Step 4 : Click on the Free Tier eligible for free tier configuration or you can choose any instance type



Step 5 : Allocate storage for the database, min – 20GiB

Step 6 : Now, tag the DBInstance and give master username and password

The screenshot shows the 'Settings' section of the AWS RDS console. It includes three main input fields: 'DB Instance identifier' with the value 'TestDB', 'Master username' with the value 'admin', and 'Master password' (masked with dots). A 'Confirm password' field is also present and masked. Below the password fields, there are three buttons: 'Cancel', 'Previous', and 'Next'. The 'Next' button is highlighted in orange.

Step 7 : Now choose the custom VPC in your Env if any otherwise go with default one and then choose a new private subnet for DB. Also select to create new Security Group.

The screenshot shows the 'Configure advanced settings' page for a database instance, specifically the 'Network & Security' section. It features several configuration options: 'Virtual Private Cloud (VPC)' set to 'VPC_01 (vpc-0057f4c36df34f2a9)', 'Subnet group' set to 'Create new DB Subnet Group', 'Public accessibility' set to 'No', 'Availability zone' set to 'ap-south-1a', and 'VPC security groups' set to 'Create new VPC security group'. The 'No' radio button for public accessibility and the 'Create new VPC security group' radio button are selected.

Step 8 : Configure the Database

Database options

Database name [Info](#)

Note: if no database name is specified then no initial MySQL database will be created on the DB Instance.

Port [Info](#)
TCP/IP port the DB instance will use for application connections.

DB parameter group [Info](#)

Option group [Info](#)

IAM DB authentication [Info](#)

Enable IAM DB authentication
Manage your database user credentials through AWS IAM users and roles.

Disable

And keep everything default and hit create database

Maintenance

Auto minor version upgrade [Info](#)

Enable auto minor version upgrade
Enables automatic upgrades to new minor versions as they are released. The automatic upgrades occur during the maintenance window for the DB instance.

Disable auto minor version upgrade

Maintenance window [Info](#)
Select the period in which you want pending modifications or patches applied to the DB instance by Amazon RDS.

Select window

No preference

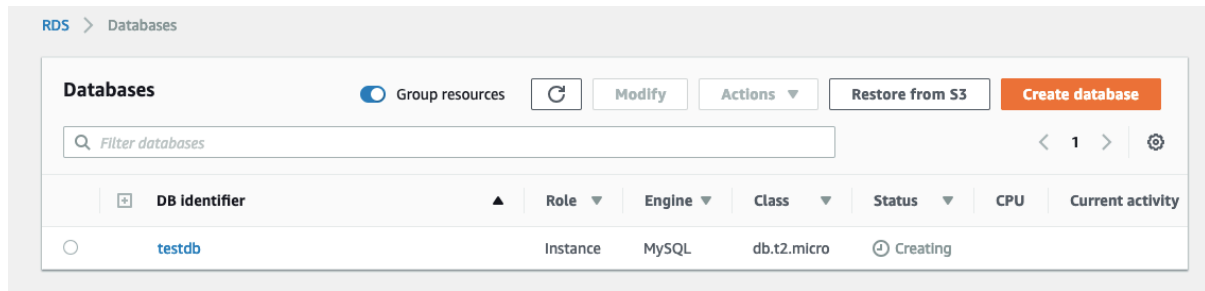
Deletion protection

Enable deletion protection
Protects the database from being deleted accidentally. While this option is enabled, you can't delete the database.

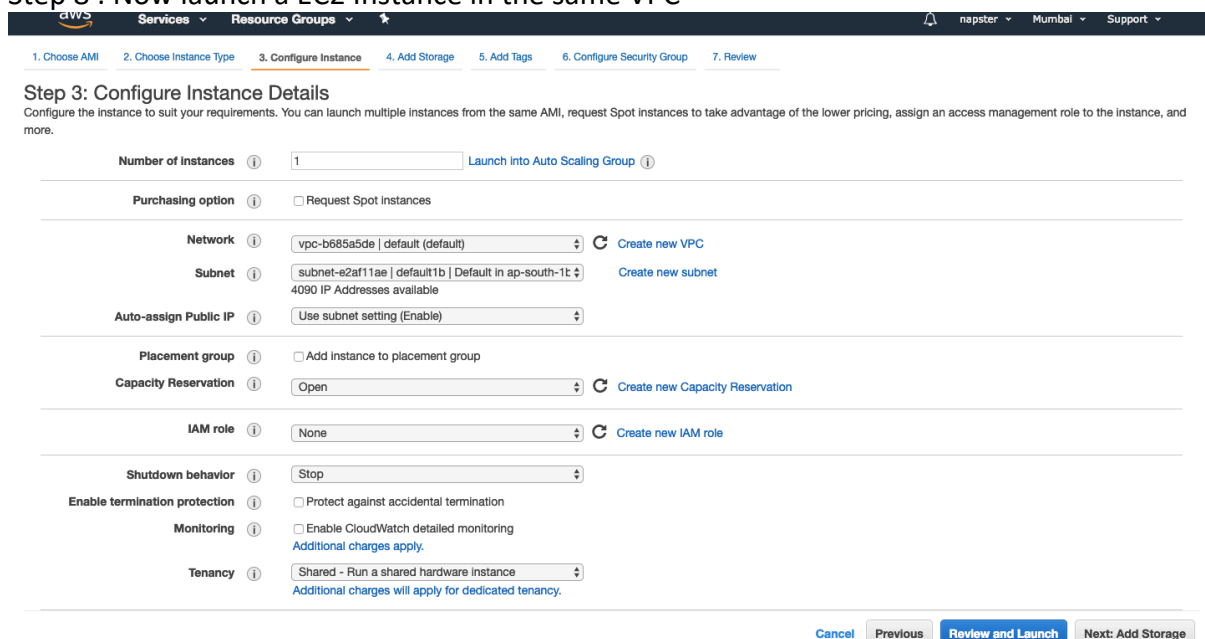
i Amazon RDS requires permissions to manage AWS resources on your behalf. By clicking Launch DB Instance, you grant permission for Amazon RDS to create a service-linked role in AWS IAM that contains the required permissions. [Learn more.](#)

Cancel

It will take some time to launch the Database Instance. Wait till the status becomes Available

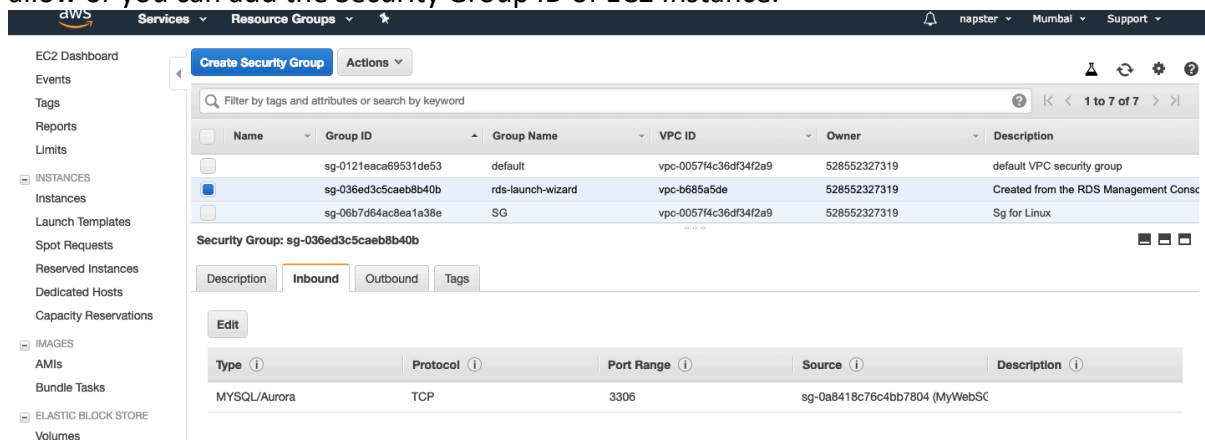


Step 8 : Now launch a EC2 Instance in the same VPC



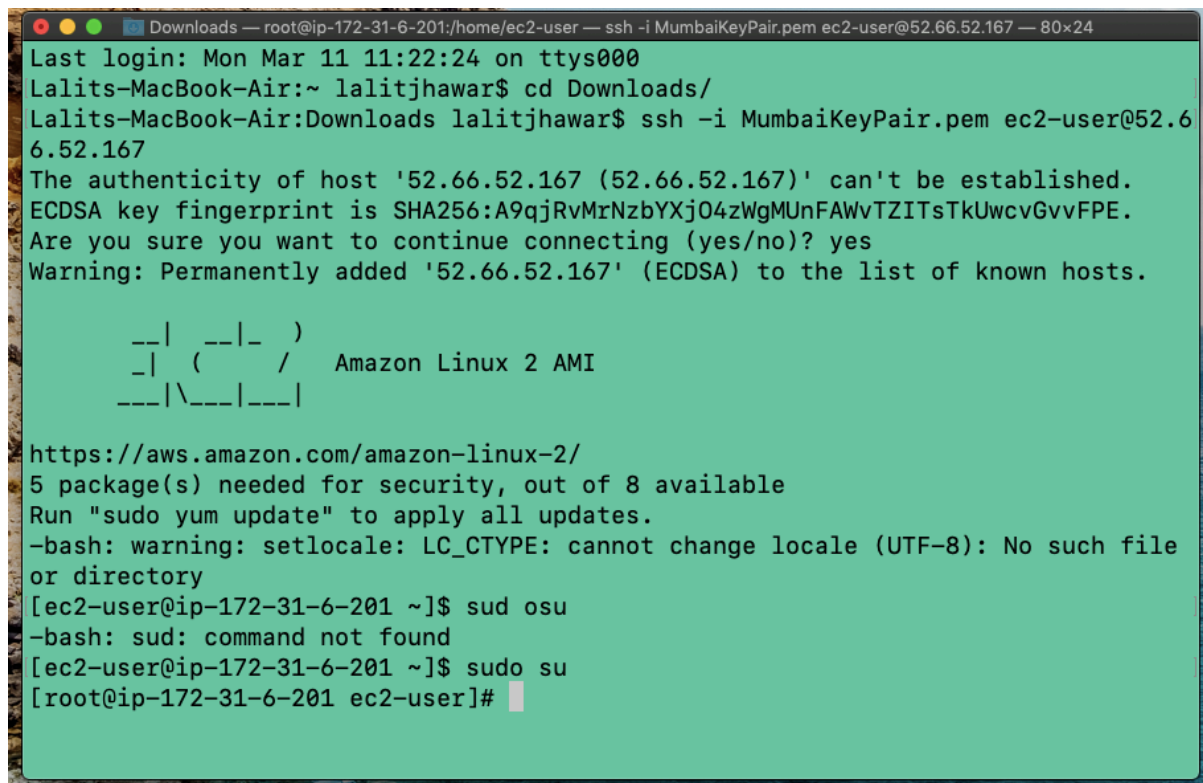
Step 9 : Now you must be having two security group : 1 for EC2 Instance and Another of RDS Instance that both are created automatically.

Here you need to copy the EC2 Instance Private IP and add a route in RDS Security Group to allow or you can add the Security Group ID of EC2 Instance.





Step 10 : Now SSH to your EC2 Instance.



Step 1 1: Now Install PHP and Apache server to test our connection.

Command : **"yum install httpd php php-mysql -y"**

```
Dependency Installed:
apr.x86_64 0:1.6.3-5.amzn2.0.2
apr-util.x86_64 0:1.6.1-5.amzn2.0.2
apr-util-bdb.x86_64 0:1.6.1-5.amzn2.0.2
generic-logos-httpd.noarch 0:18.0.0-4.amzn2
httpd-filesystem.noarch 0:2.4.37-1.amzn2.0.1
httpd-tools.x86_64 0:2.4.37-1.amzn2.0.1
libzip010-compat.x86_64 0:0.10.1-9.amzn2.0.5
mailcap.noarch 0:2.1.41-2.amzn2
mod_http2.x86_64 0:1.11.1-1.amzn2
php-cli.x86_64 0:5.4.16-45.amzn2.0.6
php-common.x86_64 0:5.4.16-45.amzn2.0.6
php-pdo.x86_64 0:5.4.16-45.amzn2.0.6

Complete!
[root@ip-172-31-6-201 ec2-user]#
```

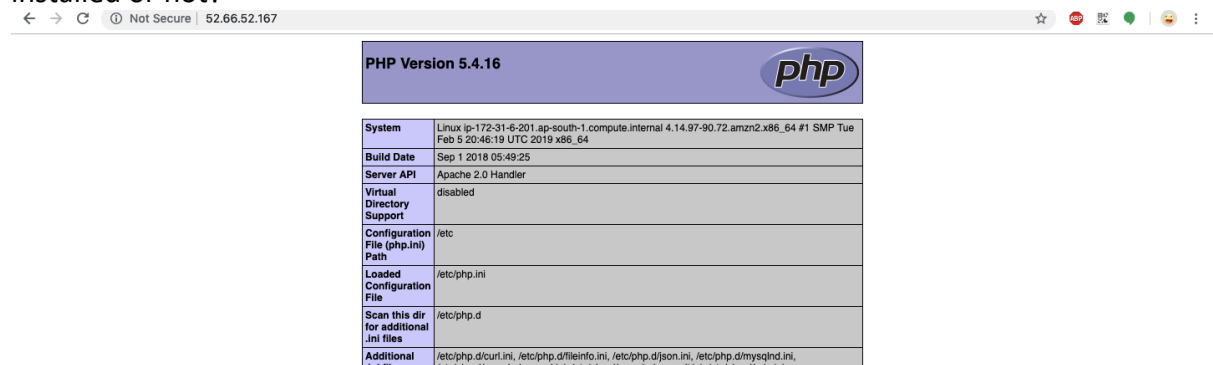
Step 12 : Once installed, start the server and create test PHP page

Command : **"service httpd start"**

"echo "<?php phpinfo();?>" > /var/www/html/index.php"

```
Complete!
[root@ip-172-31-6-201 ec2-user]# service httpd start
Redirecting to /bin/systemctl start httpd.service
[root@ip-172-31-6-201 ec2-user]# echo "<?php phpinfo();?>" > /var/www/html/index
.php
[root@ip-172-31-6-201 ec2-user]#
```

Step 13 : Now copy the Instance Public IP and check whether Apache and PHP has successfully installed or not?



If you see this output, then it means – Apache and PHP has been successfully installed.

Step 14 : Now we will create one PHP File that connects to our database.

Code:

```
<?php
$username = "awsreconnect";
$password = "awsreconnect";
$hostname = "awsreconnect.cfjmmzvhexdp.ap-south-1.rds.amazonaws.com";
$dbname = "awsreconnect";

$dbhandle = mysql_connect($hostname, $username, $password) or
die("Unable to connect to MySQL");
echo "Connected to MySQL<br>";
$selectd = mysql_select_db("$dbname",$dbhandle) or
die("Unable to connect to MySQL DB - check the database name
and try again.");
?>
```

Replace username, password, hostname and db name with actual paramters

Note : Hostname is your RDS Endpoint.

The screenshot shows the Amazon RDS console interface. On the left is a navigation menu with options like Dashboard, Databases, Performance Insights, Snapshots, Automated backups, Reserved instances, Subnet groups, Parameter groups, Option groups, Events, Event subscriptions, and Notifications. The main content area is titled 'Amazon RDS' and shows a 'Summary' section for a database instance named 'testdb'. The summary includes: DB Name (testdb), CPU (1.67%), Info (Available), Class (db.t2.micro), Role (Instance), Current activity, Engine (MySQL), and Region & AZ (ap-south-1b). Below the summary are tabs for 'Connectivity & security', 'Monitoring', 'Logs & events', 'Configuration', 'Maintenance & backups', and 'Tags'. The 'Connectivity & security' tab is active, showing details for 'Endpoint & port', 'Networking', and 'Security'. The 'Endpoint & port' section shows the endpoint as 'testdb.cfjmmzvhexdp.ap-south-1.rds.amazonaws.com' and the port as '3306'. The 'Networking' section shows the availability zone as 'ap-south-1b', VPC as 'default (vpc-b685a5de)', and subnet group as 'default'. The 'Security' section shows VPC security groups as 'rds-launch-wizard-1 (sg-0e6a7cec1fd10c231) (active)', public accessibility as 'Yes', and certificate authority as 'default'.

Here you'll find the RDS Endpoint copy and paste in the code.

```
.php
[root@ip-172-31-6-201 ec2-user]# cd /var/www/html/
[root@ip-172-31-6-201 html]# vi database.php
```



```
Downloads — root@ip-172-31-6-201:/var/www/html — ssh -i MumbaiKeyPair.pem ec2-user@52.66.52.167 — 80x24
<?php
$username = "admin";
$password = "admin123";
$hostname = "testdb.cfjmmzvhexdp.ap-south-1.rds.amazonaws.com";
$dbname = "testdb";

$dbhandle = mysql_connect($hostname, $username, $password) or die("Unable to connect to MySQL");
echo "Connected to MySQL<br>";
$selectd = mysql_select_db("$dbname",$dbhandle) or die("Unable to connect to MySQL DB - check the database name and try again.");
?>
```

Now save the file with ESC+Collan+wq

Step 15 : Now go to browser and after your Instance public Ip write "ip/database.php"

