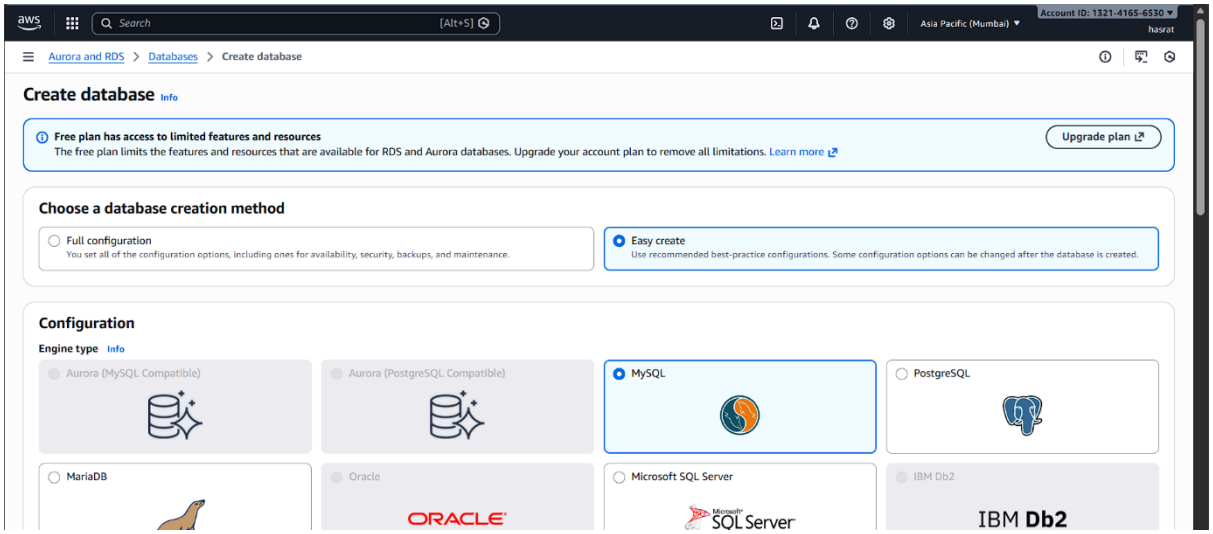
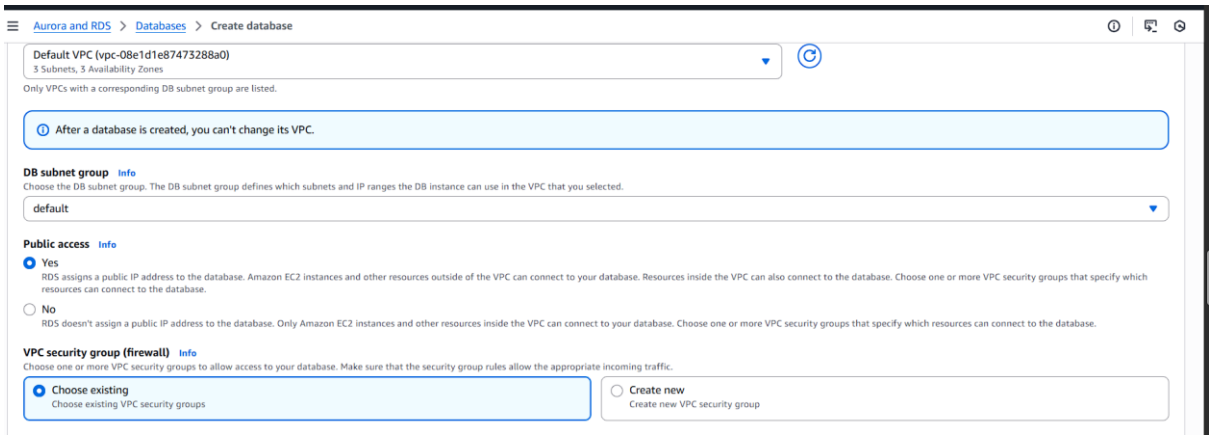
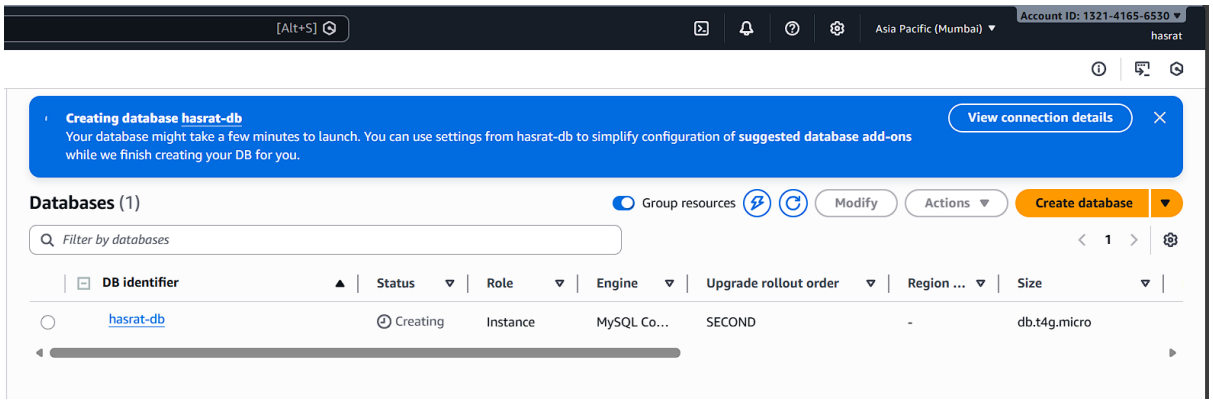
1. Search Aurora and and Create MySQL database



2. Select subnet , give public Access and choose VPC security group and create database

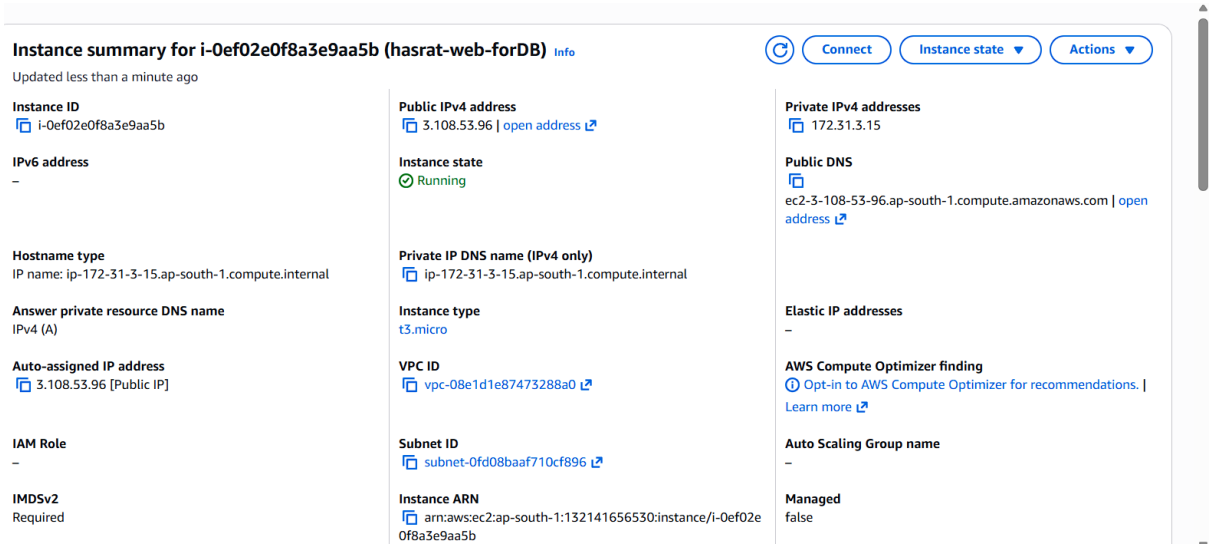


3. Database is created:

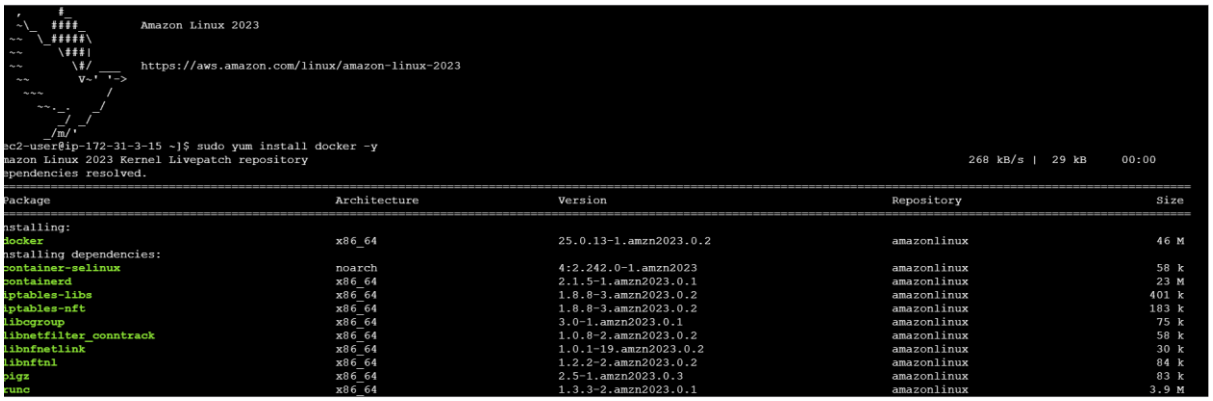


Connecting EC2 with Database

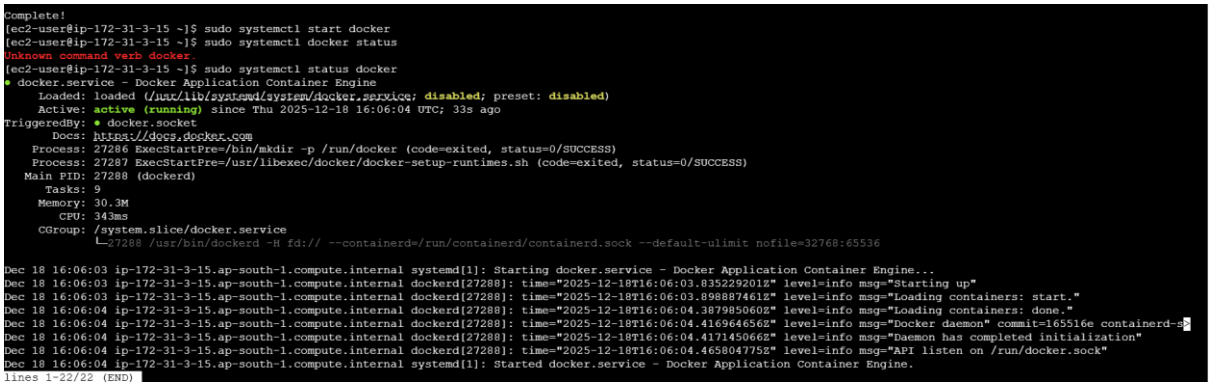
4. Create EC2 instance and connect



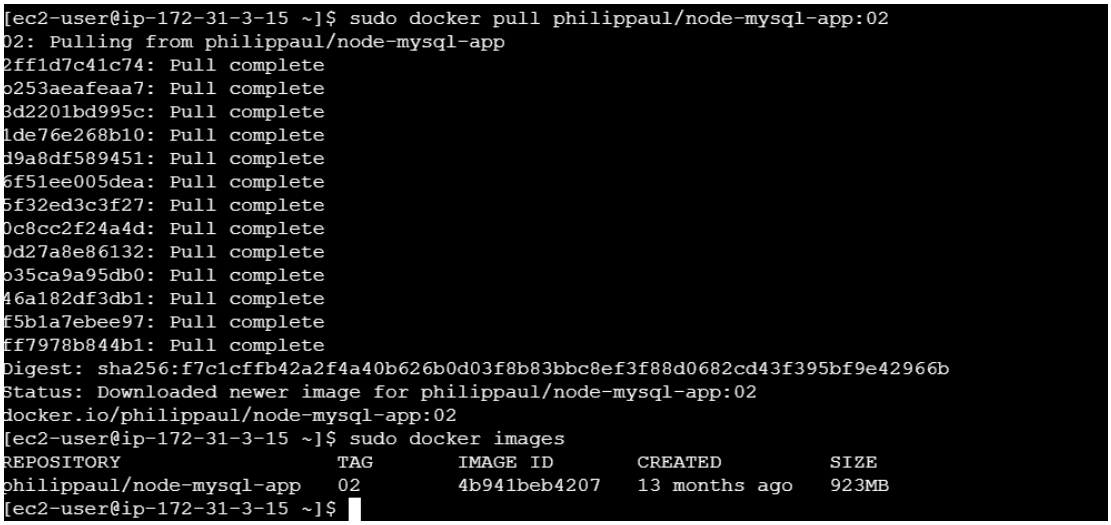
5. Install Docker in the Instance



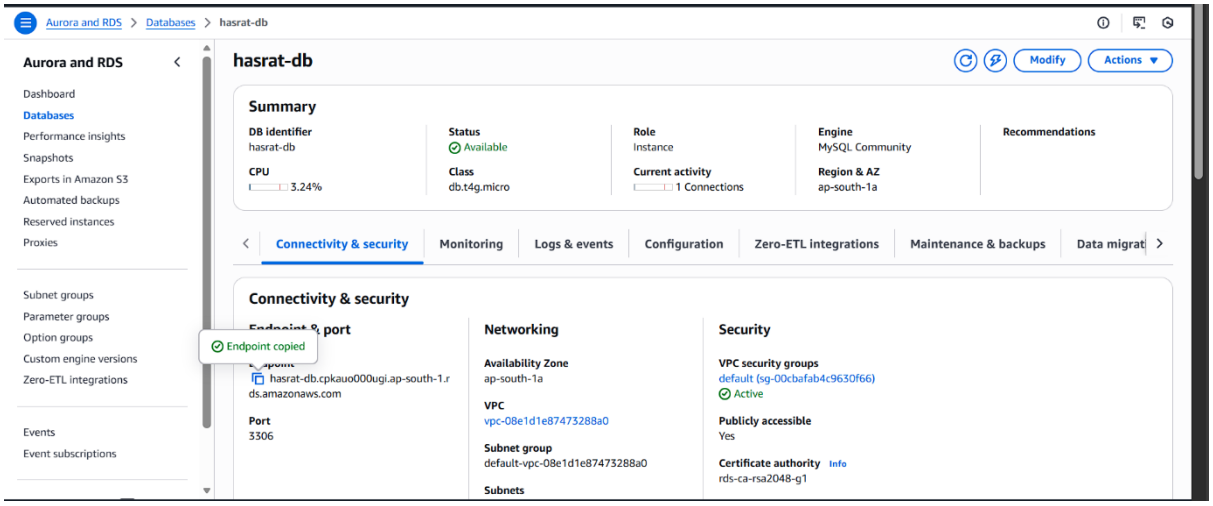
6. Starting docker and checking status



7. Pulling Docker Image named : phillippaul/node-mysql-app:02
(simple web app for database)



8. Copy endpoint of Database we created in AWS for using as DB_HOST for connecting docker image with database



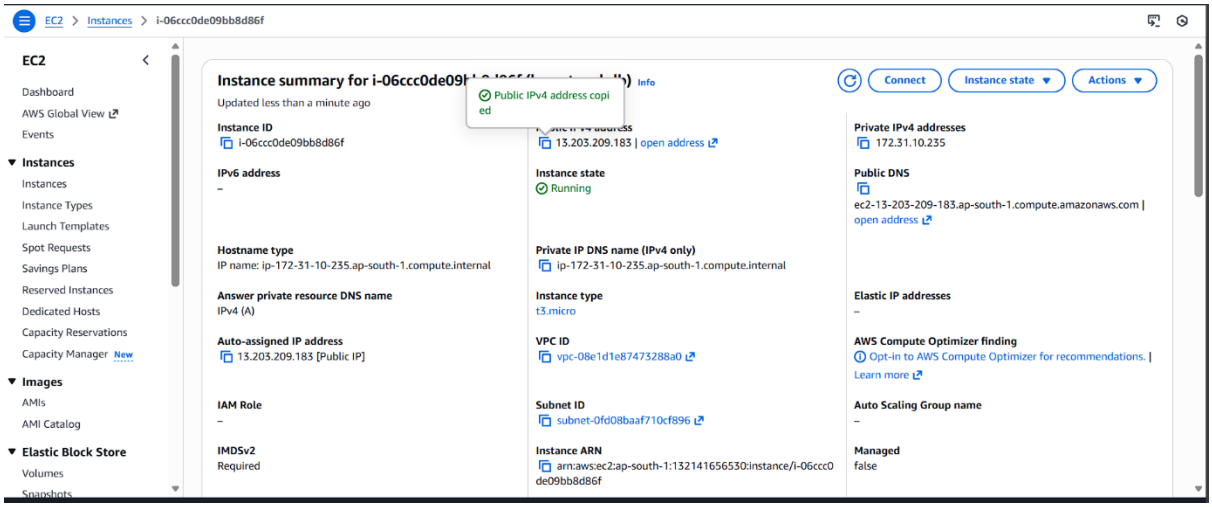
9. Connect the docker image and Database with the following command

```
[ec2-user@ip-172-31-3-15 ~]$ sudo docker run --rm -p 80:3000 -e DB_HOST="hasrat-db.cpkauo000ugi.ap-south-1.rds.amazonaws.com" -e DB_USER="hasrat-db" -e DB_PASSWORD="hasrat123" -d philippaul/node-mysql-app:02
aef06cd492672a8e765d6025d692003d4f66d127b272cfd66c9f976bc1ce3c6
[ec2-user@ip-172-31-3-15 ~]$
```

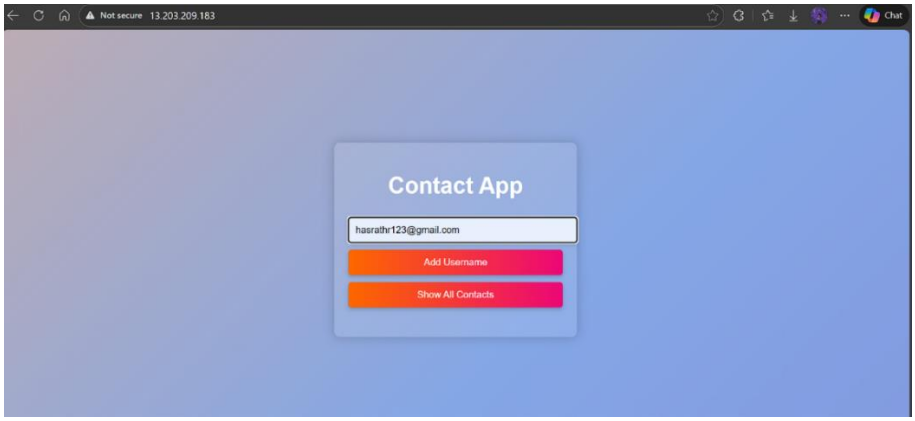
10. Checking status if the web server and database both are running properly

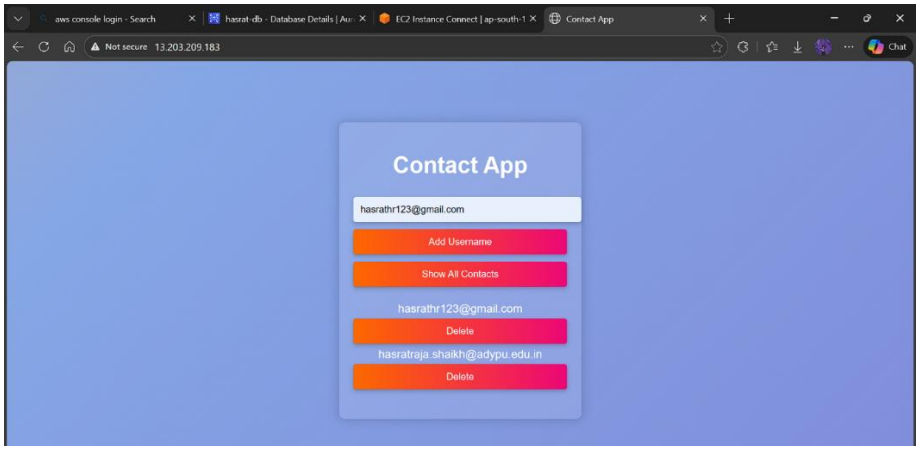
```
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
Last login: Thu Dec 18 17:55:57 2025 from 13.233.177.4
[ec2-user@ip-172-31-10-235 ~]$ sudo docker run --rm -p 80:3000 -e DB_HOST="hasrat-db.cpkauo000ugi.ap-south-1.rds.amazonaws.com" -e DB_USER="admin" -e DB_PASSWORD="hasrat123" -d philippaul/node-mysql-app:02
b6d7c173862a2babe12ceea2ae7a97ee728e026457c4007c52f6fb6c25898dc9
[ec2-user@ip-172-31-10-235 ~]$ sudo docker ps
CONTAINER ID   IMAGE                        COMMAND                  CREATED          STATUS          PORTS                                   NAMES
b6d7c173862a   philippaul/node-mysql-app:02   "docker-entrypoint.s…"   10 seconds ago   Up 8 seconds    0.0.0.0:80->3000/tcp, :::80->3000/tcp   interesting_almeida
[ec2-user@ip-172-31-10-235 ~]$ sudo docker logs -f interesting_almeida
Server is running on http://localhost:3000
Database "my_app_db" is ready.
Using database "my_app_db"
Table "contacts" is ready.
```

11. Now copy instance public IP address and paste it on Browser URL to check if the Web app is running

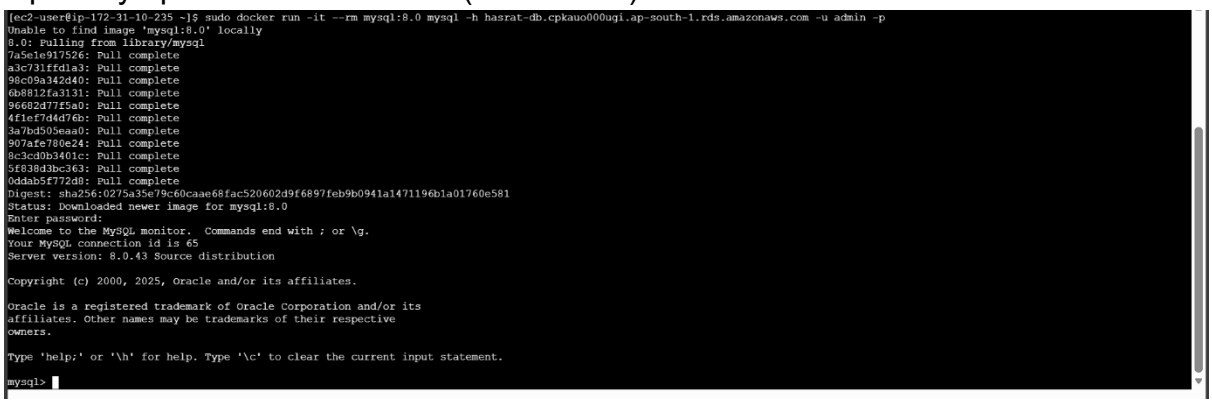


12. Add username in the web app

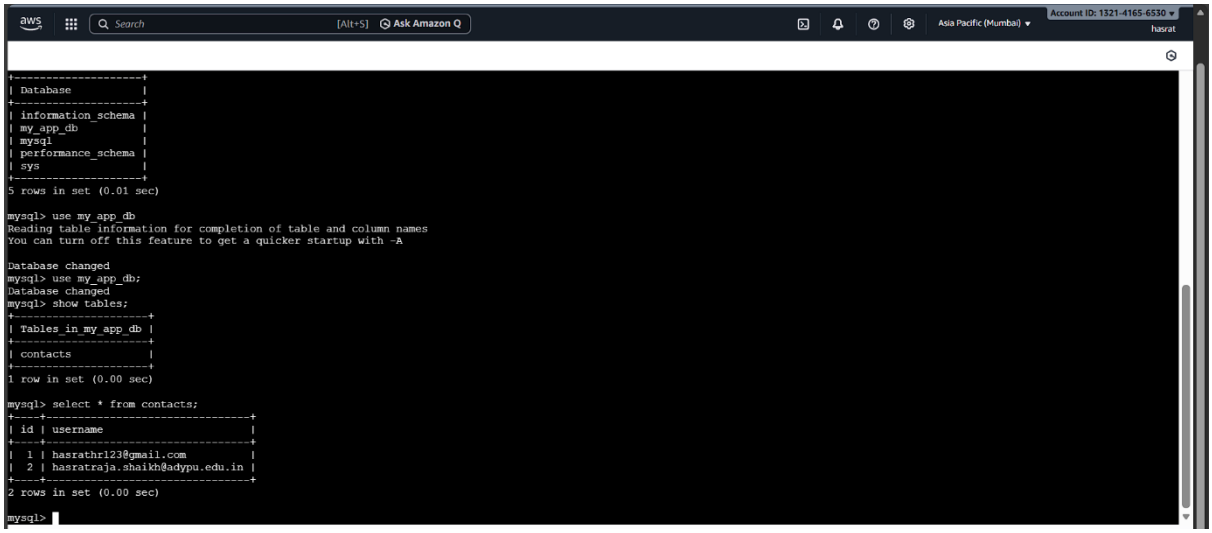




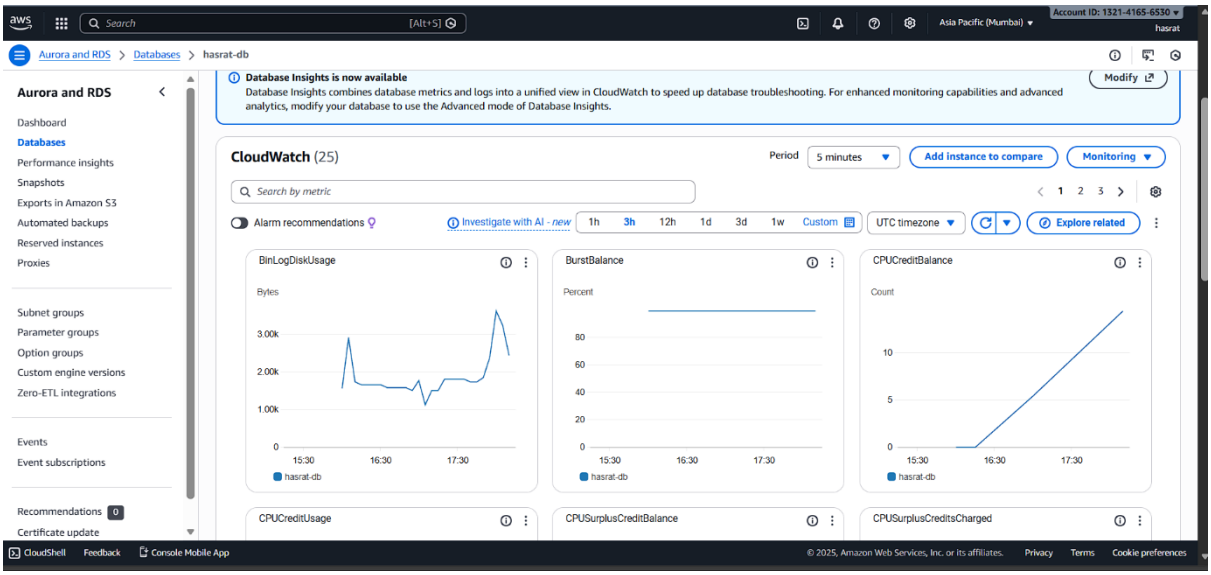
13. Open mysql to check if the data (User name) is stored in the RDS:



14. We see the Data is stored successfully



X


Search
[Alt+S]
Asia Pacific (Mumbai)
Account ID: 1321-4165-6530
hasrat
Aurora and RDS > Databases > hasrat-db
Aurora and RDS
Dashboard
Databases
Performance insights
Snapshots
Exports in Amazon S3
Automated backups
Reserved instances
Proxies
Subnet groups
Parameter groups
Option groups
Custom engine versions
Zero-ETL integrations
Events
Event subscriptions
Recommendations 0
Certificate update
Database Insights is now available
Database Insights combines database metrics and logs into a unified view in CloudWatch to speed up database troubleshooting. For enhanced monitoring capabilities and advanced analytics, modify your database to use the Advanced mode of Database Insights.
Modify
CloudWatch (25)
Period
5 minutes
Add instance to compare
Monitoring
Search by metric
Alarm recommendations
Investigate with AI - new
1h
3h
12h
1d
3d
1w
Custom
UTC timezone
Explore related
BinLogDiskUsage
Bytes
3.00k
2.00k
1.00k
0
15:30
16:30
17:30
hasrat-db
BurstBalance
Percent
80
60
40
20
0
CPUCreditBalance
Count
10
5
CPUCreditUsage
CPUSurplusCreditBalance
CPUSurplusCreditsCharged
CloudShell
Feedback
Console Mobile App
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences