AWS

chmod 400 2401084.pem

Amazon Linux 2

ssh -i 2401084.pem ec2-user@ec2-54-196-251-37.compute-1.amazonaws.com

Ubuntu

ssh -i 2401084.pem ubuntu@ec2-54-196-251-37.compute-1.amazonaws.com

Debian

ssh -i 2401084.pem admin@ec2-54-196-251-37.compute-1.amazonaws.com

CentOS / RHEL / Rocky Linux

ssh -i 2401084.pem centos@ec2-54-196-251-37.compute-1.amazonaws.com

Universal workflow for RDS and S3

Quick checklist (do this before starting)

- Your AWS account has permission to create RDS and S3 (IAM).
- Install MySQL Workbench on your laptop.
- Install AWS CLI (optional but helpful) and configure with aws configure.
- Download the dataset zip from the URL you gave and extract classicmodels.sql on your laptop.
- Know your laptop public IP (go to https://ifconfig.me or https://whatismyip.akamai.com).

Step 1: Prepare your environment

✅ Prerequisites

- AWS Account (Free Tier)
- MySQL Workbench installed

  👉 Download: https://dev.mysql.com/downloads/workbench/

- Dataset:

  👉 Download and unzip:

  https://www.mysqltutorial.org/wp-content/uploads/2023/10/mysqlsampledatabase.zip

  Inside, you'll get classicmodels.sql.

🏗️ Step 2: Create the RDS MySQL instance

1️⃣ Go to

AWS Console → RDS → Databases → Create database

- **Database creation method:** Standard create
- **Engine options:** MySQL
- **Version:** MySQL 8.0.35 (default is fine)
- **Templates:** Free Tier

2️⃣ Settings

- **DB instance identifier:** mysql-lab-01
- **Master username:** admin
- **Master password:** (set your own, e.g., Admin@12345)
- ✅ Check "Show password" so you don't forget it.

3️⃣ Instance configuration

- **DB instance class:** db.t3.micro
- **Storage type:** General Purpose SSD (gp3)
- **Allocated storage:** 20 GB (default is fine)

- **Disable storage autoscaling**

4️⃣ Connectivity

- **Compute resource:** Don’t connect to EC2
- **Public access:** ✅ Yes (Enable temporarily)
- **VPC security group:** Choose “Create new”
  - Name: mysql-lab-01-sg
- **Port:** 3306

5️⃣ Additional configuration

- **Initial database name:** classicmodels
- **Backup retention:** 0 days (disable backups)
- **Monitoring, Performance Insights:** Disable all
- **Deletion protection:** Uncheck

6️⃣ Add tags

Key	Value
name	your_name
project	mysql-lab-01
roll_no	your_roll_number
date	today’s date
teacher_name	Prof. Meera Kumar

✅ Click **Create database**

🕒 Step 3: Wait until status = “Available”

Once available:

- Open your instance details → **Connectivity & security tab**
- Copy **Endpoint** — looks like:

```
mysql-lab-01-1234abcd.us-east-1.rds.amazonaws.com
```

-
- Note **Port = 3306**

🔐 Step 4: Allow your computer to connect

1. Go to **EC2 → Security Groups → mysql-lab-01-sg**
2. In **Inbound rules → Edit inbound rules → Add rule**
   - **Type:** MySQL/Aurora
   - **Port:** 3306
   - **Source:** My IP
   - ✅ Click **Save rules**

🧩 Step 5: Connect via MySQL Workbench

Open MySQL Workbench → click **+ (New Connection)**

Fill these:

Field	**Value**
Connection Name	mysql-lab-01
Hostname	*your RDS endpoint*
Port	3306
Username	admin
Password	Store in vault (Admin@12345)

Click **Test Connection** → ✅ should say "Successfully made the MySQL connection."

If it fails → check:

- Correct username/password
- Public access is **Yes**
- Security group inbound rule = **My IP**

💾 Step 6: Import the dataset

1. Open your classicmodels.sql file from downloaded zip
2. In MySQL Workbench:
    - Go to **Server → Data Import**
    - Select **Import from Self-Contained File**
    - Choose classicmodels.sql
    - Default Target Schema: classicmodels
    - Click **Start Import**

✅ When done, tables will appear under your database:

- customers
- employees
- orders
- payments
- etc.

📤 Step 7: Export the database

1. In Workbench → **Server → Data Export**
2. Select classicmodels schema
3. Choose **Export to Self-Contained File**
4. File name: classicmodels_export_01.sql
5. Click **Start Export**

☁️ Step 8: Create an S3 bucket

1. Go to **AWS → S3 → Create bucket**
2. Bucket name: yourrollno-bucket-01
    - Example: 2401084-bucket-01
3. **Region:** us-east-1
4. Uncheck “Block all public access”

5. Add tags:

Key	Value
name	your_name
roll_no	your_roll_number
project	mysql-lab-01
date	today’s date
teacher_name	Prof. Meera Kumar

Click **Create bucket**

📁 Step 9: Upload and verify

1. Open the bucket → **Upload**
    - Upload classicmodels_export_01.sql
2. Download it again to confirm it works.

✅ Take screenshots:

- RDS instance details
- Endpoint
- Security group inbound rule
- MySQL Workbench connection
- Imported tables
- Exported file
- S3 upload
- S3 download

🧹 Step 10: Cleanup (after screenshots)

1. Delete S3 bucket (Empty → Delete)
2. Delete RDS instance:
    - Uncheck “Create final snapshot”
    - Confirm deletion

📄 Step 11: Documentation

Create a **Google Doc** named:

📘 "AWS MySQL Lab 01 - RDS + S3 Integration - "

Include:

1. Your name, roll number, project
2. Stepwise screenshots
3. Final remarks

Instance

Question 1: Create EC2 Instance + Tag + SSH Login + Screenshot Documentation

---

🧩

Step 1: Log in to AWS Management Console

1. Go to: https://aws.amazon.com/console/
2. Click **Sign in to Console**.
3. Use your **student AWS credentials** (root or IAM user as provided).
4. Once logged in, **search for "EC2"** in the AWS search bar → click **EC2.**

Step 2: Launch an EC2 Instance

1. Click **"Launch Instance"** button.
2. Fill the following details:

Name and Tags

Key	Value
Name	(Your Name or Roll No)

Key	Value
project	Cloud-Intro
roll_no	(Your Roll Number)
date	(Today’s Date)
teacher_name	Prof. Meera Kumar

💡 You can add all tags here directly using **“Add additional tags”** → +Add tag.

⚙️

Step 3: Choose Amazon Machine Image (AMI)

- Under **Application and OS Images**, choose:
    - **Ubuntu Server 22.04 LTS (HVM), SSD Volume Type**

(Ensure architecture: **64-bit (x86)**)

⚙️

Step 4: Choose Instance Type

- Select: **t2.micro** (Free tier eligible).

🔑

Step 5: Create Key Pair

1. Under **Key pair (login)** section → click **Create new key pair**.
2. Name: your **roll_number** (e.g., CS101)
3. Type: **RSA**
4. File format:
    - .pem (for Mac/Linux)
    - .ppk (for Windows PuTTY users)
5. Click **Create key pair** → it will automatically download to your system.

⚠️ **Keep it safe** — you’ll need it for SSH login.

🔒

Step 6: Network Settings

1. Under **Network Settings**, leave defaults:
    - VPC: default
    - Subnet: default
2. Tick “**Allow SSH traffic from My IP**”.

Step 7: Configure Storage

You’ll see default root volume as /dev/sda1.

Volume Type	Size	Description
Root Volume	10 GB	OS storage
New Volume	8 GB	Additional disk

To add:

1. Under **Storage**, click **“Add new volume”**
2. Size = 8
3. Volume type = gp3
4. Leave default device name (e.g., /dev/sdf)

Step 8: Review and Launch

1. Double-check details:
    - AMI: Ubuntu 22.04 LTS
    - Instance Type: t2.micro
    - Storage: 10 GB + 8 GB
    - Tags: Added
    - Key pair: your_roll_number.pem
2. Click **Launch Instance**.

Step 9: View Instance Details

Once launched:

1. Click **View Instances**.

2. You’ll see your instance listed.
3. Wait until **Status = Running** and **Status Check = 2/2 passed**.

Take a **screenshot** of:

- Instance Name
- Instance ID
- Instance State = Running
- Public IPv4 address
- Key name
- Tags visible

Step 10: Connect via SSH

For

Mac/Linux

:

1. Open Terminal.
2. Navigate to your .pem file location:

```
cd ~/Downloads
```

1.
2. Change file permissions:

```
chmod 400 your_roll_number.pem
```

1.
2. Copy your **Public IPv4 DNS** from AWS console (Example: ec2-3-108-45-12.ap-south-1.compute.amazonaws.com)
3. SSH into instance:

```
ssh -i your_roll_number.pem ubuntu@<Public_DNS>
```

1.
2. Example:

```
ssh -i CS101.pem ubuntu@ec2-13-233-144-12.ap-south-1.compute.amazon
aws.com
```

1.
2. On success, you'll see:

```
Welcome to Ubuntu 22.04 LTS (GNU/Linux 5.15.0-1031-aws x86_64)
ubuntu@ip-172-31-32-123:~$
```

📸 **Take screenshot** of:

- Terminal showing successful login.
- Instance prompt (ubuntu@ip-xxxx:~$)

📀

Step 11: Verify Attached Volumes

Inside your instance terminal:

```
lsblk
```

You should see:

```
NAME   MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
xvda   202:0    0   10G  0 disk
└─xvda1 202:1    0   10G  0 part /
xvdf   202:80   0    8G  0 disk
```

📸 Take screenshot of this too.

📄

Step 12: Create Google Doc for Submission

1. Open **Google Docs**.
2. Title: AWS EC2 Practical – Roll No: <Your Roll Number>
3. Include sections:

1️⃣ Instance Creation Details

- Screenshot of instance summary page.
- Screenshot showing tags.
- Screenshot showing key pair.

2️⃣ SSH Login Proof

- Screenshot of terminal login.

3️⃣ Disk Verification

- Screenshot of lsblk command.

4️⃣ Tags Summary

- Copy/paste tags table.

1. Add your **Name, Roll Number, Date, Teacher Name** at the bottom.
2. Save the document.
3. Rename file as:

```
AWS_EC2_Practical_<RollNumber>.docx
```

1.
2. Upload or share as per instructions.

💡 Optional (if you want to make it professional)

Run:

```
sudo apt update
```

Then take a screenshot showing it working — it proves your login is interactive.

```
lsblk              # to check attached disks
sudo apt update       # to update the system
sudo mkdir /data      # optional: create mount directory
```

Migration: