# **Migrate An On-Prem Database | AWS RDS**

1. Set up EC2 Instance:

Create an EC2 instance and assign an instance name.

Connect to the instance via command line:  
bash  
Copy code  
ssh -i "newkeypair.pem" ubuntu@ec2-54-91-41-189.compute-1.amazonaws.com

1. Install and Configure MySQL:

Update and install MySQL server:  
bash  
Copy code  
sudo apt-get update

sudo apt-get install mysql-server

Verify MySQL installation:  
bash  
Copy code  
mysql --version

sudo systemctl status mysql

sudo systemctl start mysql

Access MySQL and create a test database:  
sql  
Copy code  
sudo mysql -u root -p

CREATE DATABASE test;

USE test;

CREATE TABLE customer (

personid INT AUTO\_INCREMENT PRIMARY KEY,

firstname VARCHAR(50) NOT NULL,

lastname VARCHAR(50) NOT NULL,

age INT

);

INSERT INTO customer (firstname, lastname, age) VALUES

('John', 'Doe', 30),

('Jane', 'Smith', 25),

('Emily', 'Johnson', 40);

1. Install and Configure Percona XtraBackup:

Install XtraBackup:  
bash  
Copy code  
sudo apt install percona-xtrabackup

xtrabackup --version

Create Backup Directory:  
bash  
Copy code  
mkdir -p ~/s3-restore/backup

Take MySQL Backup:  
bash  
Copy code  
sudo xtrabackup --backup --user=root --password=1502 --stream=xbstream --target-dir=~/s3-restore/backup | split -d --byte=100MB - ~/s3-restore/backup/backup.xbstream

1. Install AWS CLI on EC2:

Install and configure the AWS CLI:  
bash  
Copy code  
sudo apt install unzip curl -y

curl "https://awscli.amazonaws.com/awscli-exe-linux-x86\_64.zip" -o "awscliv2.zip"

unzip awscliv2.zip

sudo ./aws/install

aws --version

aws configure

Check S3 Buckets:  
bash  
Copy code  
aws s3 ls

1. Upload Backup to S3:
   * Create a new S3 bucket (e.g., akshay-1209333).

Copy backup files to the S3 bucket:  
bash  
Copy code  
aws s3 cp ~/s3-restore/backup/ s3://akshay-1209333 --recursive

1. Restore Backup to RDS:
   * In RDS Console, select Restore from S3 and choose your bucket name.
   * Select Engine and configure settings:
     + DB Instance Class: db.t3.small
     + DB Cluster Identifier: database-test
     + Master Username: admin
     + Password: admin12345
     + Public Access: Yes
2. Configure Security Settings:
   * Ensure the security group has MySQL/Aurora inbound rule set to allow access from anywhere.

This process will set up a secure, scalable AWS RDS instance with data migrated from an on-premises MySQL database.