AWS RDS and DynamoDB User Guide

This guide provides step-by-step instructions for creating, configuring, and managing AWS RDS MySQL databases and DynamoDB tables.

Table of Contents

AWS RDS MySQL Database Setup

Prerequisites

- AWS Account with appropriate permissions
- Access to AWS Console
- MySQL client installed locally (for database connections)

Step 1: Navigate to RDS Service

1. In the AWS Console, search for "RDS"
2. Click on "Aurora RDS" or navigate to the RDS dashboard
3. Ensure you're in the correct region (e.g., eu-north-1)

Step 2: Create Database

1. Click **"Databases"** in the left navigation panel
2. Click **"Create database"** button
3. Configure the following settings:

Engine Configuration

- **Engine type**: MySQL
- **Edition**: MySQL Community

Templates

- Select **"Free tier"** (creates single AZ deployment)

Settings

- **DB instance identifier**: `mySQLDB-01`
- **Master username**: `admin`
- **Master password**: Choose "Self managed" and set password (e.g., `Raviv12345!`)

Connectivity

- **Public access**: Enable (Yes)
- Leave other settings as default

4. Click **"Create database"**

> **Note**: Database creation typically takes 5-10 minutes to complete.

Database Security Configuration

Step 1: Access Security Groups

1. Navigate to your newly created RDS instance
2. Click on the database instance name
3. Go to **"Connectivity & security"** tab
4. Click on the **Security group** link

Step 2: Configure Inbound Rules

1. Select the security group
2. Click **"Edit inbound rules"**
3. **Delete all existing rules**
4. Click **"Add rule"**
5. Configure new rule:
   - **Type**: MySQL/Aurora
   - **Port**: 3306 (auto-populated)

- **Source**: My IP (recommended for security)

6. Click **"Save rules"**

> **Security Note**: Using "My IP" restricts access to your current IP address only, which is more secure than allowing all traffic.

Connecting to RDS MySQL Database

Step 1: Obtain Connection Details

1. From the RDS console, copy the **Endpoint** of your database
2. Format: `mysqldb-01.cf024kmmgpuw.eu-north-1.rds.amazonaws.com`

Step 2: Connect Using MySQL Client

```bash
mysql -h mysqldb-01.cf024kmmgpuw.eu-north-1.rds.amazonaws.com -u admin -p
```

When prompted, enter your password: `Raviv12345!`

Expected Connection Output

```
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 30
Server version: 8.0.41 Source distribution
Copyright (c) 2000, 2025, Oracle and/or its affiliates.

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

Database Operations

Step 1: View Existing Databases

```sql
SHOW DATABASES;
```

Expected output:

```
+--------------------+
| Database           |
+--------------------+
| information_schema |
| my_app_db          |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
```

Step 2: Create New Database

```sql
CREATE DATABASE db01;
```

Step 3: Verify Database Creation

```sql
SHOW DATABASES;
```

Step 4: Use Database and Create Table

```sql
USE db01;

CREATE TABLE users (
    id INT AUTO_INCREMENT PRIMARY KEY,
    username VARCHAR(50) NOT NULL UNIQUE,
    email VARCHAR(100) NOT NULL UNIQUE,
    password_hash VARCHAR(255) NOT NULL,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

Step 5: Insert Sample Data

```sql
INSERT INTO users (username, email, password_hash) VALUES
('user_beta', 'beta@example.com', 'hashed_password_for_beta');
```

Step 6: Query Data

```sql
SELECT * FROM users;
```

Expected output:

```
+----+-----------+-----------------+-----------------------+---------------------+
| id | username  | email           | password_hash         | created_at          |
+----+-----------+-----------------+-----------------------+---------------------+
|  1 | user_beta | beta@example.com | hashed_password_for_beta | 2025-06-29 18:13:44 |
+----+-----------+-----------------+-----------------------+---------------------+
```

Step 7: Exit MySQL Client

```sql
EXIT;
```

Cleanup and Monitoring

Delete RDS Database

1. Return to AWS RDS Console
2. Select your database instance
3. Click **"Actions"** → **"Delete"**
4. **Uncheck** "Create final snapshot" (to fully delete)
5. Type **"delete me"** in the confirmation field
6. Click **"Delete"**

Monitor with CloudWatch

1. Navigate to **CloudWatch** service
2. Go to **"Logs"** → **"Log groups"**
3. Check for any existing logs related to your RDS instance
4. Review logs for troubleshooting if needed

AWS DynamoDB Setup

Step 1: Navigate to DynamoDB

1. In AWS Console, search for "DynamoDB"
2. Access the DynamoDB Dashboard
3. Note: DAX (DynamoDB Accelerator) provides caching capabilities

Step 2: Create Table

1. Click **"Tables"** in the left navigation
2. Click **"Create table"**
3. Configure table settings:
   - **Table name**: `ProductCatalog`
   - **Partition key**: `ProductID` (String)
4. Click **"Create table"**

DynamoDB Operations

Step 1: Create Items

1. Navigate to your created table
2. Click **"Actions"** → **"Create item"**

Item 1:

json

```
{
 "ProductID": {
  "S": "101"
 },
 "Price": {
  "N": "0"
 },
 "ProductTitle": {
  "S": "Devops course 2026"
 },
 "StartDate": {
  "S": "2026-01-01"
 }
}
```

Item 2:

```json
{
  "ProductID": {
    "S": "102"
  },
  "ProductTitle": {
    "S": "Full stack course 2026"
  },
  "StartDate": {
    "S": "2026-01-01"
  }
}
```

Step 2: Explore and Query Items

1. Click **"Explore items"**
2. Use **"Scan"** to view all items
3. Use **"Add filters"** to create specific queries
4. Click **"Run"** to execute filters

Step 3: Delete Table

1. Select the table
2. Click **"Delete"**
3. Type **"confirm"** in the confirmation field
4. Click **"Delete table"**

Best Practices

RDS Security

- Always use security groups to restrict database access
- Use strong passwords and consider AWS Secrets Manager
- Enable encryption at rest and in transit for production environments
- Regularly update database engine versions

DynamoDB Optimization

- Choose appropriate partition keys to avoid hot partitions
- Use composite keys when needed for better data distribution
- Monitor capacity metrics and adjust provisioned throughput
- Consider using DynamoDB Streams for real-time processing

Cost Management

- Delete unused resources promptly
- Use free tier resources for learning and testing
- Monitor billing dashboard regularly
- Set up billing alerts for cost control

Monitoring and Logging

- Enable CloudWatch monitoring for both RDS and DynamoDB
- Set up alerts for important metrics (CPU usage, connection count, etc.)
- Regularly review logs for performance optimization
- Use AWS X-Ray for distributed tracing when applicable

Troubleshooting Common Issues

RDS Connection Issues

- Verify security group rules allow your IP
- Check if database is publicly accessible
- Ensure correct endpoint and port (3306 for MySQL)
- Verify username and password

DynamoDB Issues

- Check IAM permissions for DynamoDB operations
- Verify table exists and is in ACTIVE state
- Ensure correct partition key values
- Monitor throttling metrics

General AWS Issues

- Verify you're in the correct AWS region
- Check service limits and quotas

- Review IAM policies and permissions
- Use AWS CLI for programmatic access when needed

This guide covers the essential operations for AWS RDS and DynamoDB based on the provided lesson transcript. For production deployments, additional security and performance considerations should be implemented.