**AWS Databases: RDS, MySQL & High Availability - Interactive Guide**

**📌 Table of Contents**

1. **🔍 Introduction to Databases**
   * Structured vs. Unstructured vs. In-Memory
2. **🏗️ Setting Up RDS MySQL**
   * Subnet Groups & Multi-AZ Deployment
3. **🖥️ Connecting via SQL Workbench**
   * Data Insertion & Queries
4. **🔄 Failover & High Availability**
   * Testing Automatic Failover
5. **🚀 Next Steps**
   * DynamoDB, API Gateway & More

**1. 🔍 Introduction to Databases**

**📂 Types of Databases**

| **Type** | **Use Case** | **AWS Service** | **Example** |
| --- | --- | --- | --- |
| **Structured** | User credentials, transactions | **RDS (MySQL, PostgreSQL)** | Amazon login system |
| **Unstructured** | Images, videos, logs | **S3, DynamoDB** | Product images on Amazon |
| **In-Memory** | High-speed caching | **ElastiCache (Redis)** | Shopping cart data |

**💡 Key Insight:**

* **RDS** = For structured data (e.g., user accounts).
* **DynamoDB** = For NoSQL (e.g., product catalogs).
* **ElastiCache** = For low-latency reads (e.g., session data).

**2. 🏗️ Setting Up RDS MySQL**

**Step 1: Create a DB Subnet Group**

* **Why?** To deploy databases across **multiple Availability Zones (AZs)**.
* **Navigate to:**

RDS → Subnet Groups → Create DB Subnet Group

* **Select:**
  + VPC + **3 public subnets** (for testing).
  + **Production Tip:** Use private subnets for security.

**Step 2: Launch RDS MySQL**

* **Configurations:**
  + **Engine:** MySQL
  + **Multi-AZ Deployment:** ✅ (For failover)
  + **Instance Class:** db.t3.medium (2 vCPU, 4GB RAM)
  + **Storage:** 20GB (GP2)
  + **Credentials:** admin + Strong password
  + **Connectivity:** Publicly accessible (for testing)

**⚠️ Warning:**

* **Backups & Encryption** disabled for demo (enable in production).
* **Monitoring:** Turn on **Enhanced Monitoring**.

**3. 🖥️ Connecting via SQL Workbench**

**Step 1: Install Tools**

1. **On Windows EC2:**
   * Install **MySQL Workbench** + **Visual C++ Redistributable**.
   * Disable firewall (firewall.cpl → Turn off).
2. **Connect to RDS:**
   * **Endpoint:** Copy from RDS console (e.g., mydb.123.us-east-1.rds.amazonaws.com).
   * **Port:** 3306
   * **Username:** admin

**Step 2: Insert & Query Data**

sql

Copy

Download

*-- Create a table*

CREATE TABLE movies (

id INT AUTO\_INCREMENT PRIMARY KEY,

title VARCHAR(255),

year INT

);

*-- Insert data*

INSERT INTO movies (title, year) VALUES

('Inception', 2010),

('The Dark Knight', 2008);

*-- Query data*

SELECT \* FROM movies;

**🎬 Demo:** Data syncs instantly between RDS and Ubuntu app!

**4. 🔄 Failover & High Availability**

**Testing Automatic Failover**

1. **Trigger Failover:**
   * **RDS Console → Databases → Select DB → Actions → Reboot with failover**.
2. **Observe:**
   * **Ubuntu app** briefly disconnects (~2 mins).
   * Traffic shifts to **Secondary AZ** (IP changes).

bash

nslookup mydb.123.us-east-1.rds.amazonaws.com

**📌 Pro Tip:**

* Use **ElastiCache** to cache queries during failover.