A guide for setting up a MySQL database on an Ubuntu EC2 instance, creating RDS MySQL instances, and performing database migration using AWS Database Migration Service (DMS). Here is a detailed breakdown:

Part 1: Installing MySQL on Ubuntu

1. **Launch One Ec2 Instance.**
2. **Security Group Configuration**
    - Ensure the security group includes HTTP, SSH, RDP, and MySQL/Aurora.
3. **Launch EC2 Instance**
    - Switch to the sudo user:

      Sudo su
    - Update the packages

      ```
      sudo apt update
      ```
    - Install MySQL Server:

      ```
      sudo apt install mysql-server
      ```
    - Check MySQL status:

      ```
      sudo service mysql status
      ```

Part 2: MySQL User Setup

1. **Login to MySQL:**

   ```
   mysql
   ```
2. **Create New User:**

   ```
   CREATE USER 'newuser'@'%' IDENTIFIED BY 'password';
   ```
3. **Grant Privileges:**

   ```
   GRANT ALL PRIVILEGES ON *.* TO 'newuser'@'%' WITH GRANT OPTION;
   FLUSH PRIVILEGES;
   EXIT;
   ```

Part 3: MySQL Configuration

1. **Edit MySQL Configuration:**

   ```
   vi /etc/mysql/mysql.conf.d/mysqld.cnf
   ```

   - Set line numbers:

     ```
     :se nu
     ```

   - Change `bind-address` to:

     ```
     0.0.0.0
     ```

   - Save and quit:

     ```
     :wq!
     ```

2. **Restart MySQL Service:**

   ```
   sudo service mysql restart
   ```

3. **Check MySQL Status:**

   ```
   sudo service mysql status
   ```

Part 4: Connect with MySQL Workbench or DBeaver

- **Connection Details:**
  - Name: (Try To Specify name)
  - Hostname: `EC2 instance public IP address`
  - Port: `3306`
  - Username: `newuser`
  - Password: `password`

RDS MySQL Setup

1. **Create RDS MySQL:**
    - **Steps:**
        1. Select Standard
        2. Select MySQL
        3. Template: Free Tier
        4. DB Identifier: `name the database`
        5. Master Username: `select autogenerated password`
        6. Storage: gp2
        7. Public Access: Yes (for MySQL Workbench connection)
        8. Security Group: Include SSH, RDP, HTTP, MySQL/Aurora
        9. Default Subnet: Selected
        10. Backup: Uncheck enable automated backup
        11. Create RDS

2. **Connect with MySQL Workbench or DBeaver**

    Connection Details:

    - Name: (Try To Specify name)
    - Hostname: `RDS endpoint`
    - Port: `3306`
    - Username: `admin`
    - Password: (Try to use `autogenerated password`)

Creating Endpoints in DMS

1. **Source Endpoint:**
    - Name:
    - Source Engine: Selected engine
    - Access Endpoint: Provide access information manually
    - Details:
        - Server Name: `EC2 instance IP address`
        - Port: `3306`
        - Username: `newuser`
        - Password: `password`

2. **Target Endpoint:**
    - Name:
    - Target Engine: Selected engine (RDS DB)
    - Access Endpoint: Provide access information manually
    - Details:
        - Server Name: `RDS endpoint`
        - Port: `3306`
        - Username: `admin`
        - Password: `autogenerated password`
        -

Replication Instance

- **Create Replication Instance:**
    - Name:
    - Description:
    - Instance Class: `DMS.t3.medium`
    - High Availability: Single AZ

Creating Migration Task

1. **Task Setup:**
    - Task Identifier: `name`
    - Replication Instance: Selected instance
    - Source Database Endpoint: Selected source endpoint
    - Target Database Endpoint: Selected target endpoint
    - Migration Type: Migrate existing data
    - Table Mapping: Add new selection rule for schema
    - Pre-Migration Assessment: Uncheck turn on pre-migration assessment

2. **After Creating:**
    - Check the manual process in MySQL Workbench.

This document provides a comprehensive step-by-step guide for setting up MySQL on an EC2 instance, configuring RDS, and performing database migrations using AWS DMS.