**To install MySQL, Mail utilils and take MYSQL backup on Ubuntu 20.04**

**1. Update the Package Index**

Open a terminal and update the package index to ensure you have the latest repository information:

sudo apt update

**2. Install MySQL Server**

Install the MySQL server package by running:

sudo apt install mysql-server

This will install MySQL along with its dependencies.

**3. Start the MySQL Service**

Ensure the MySQL service is running with the following command:

sudo systemctl start mysql

To enable MySQL to start automatically on system boot, run:

sudo systemctl enable mysql

**4. Secure the MySQL Installation**

MySQL comes with a security script to enhance the default security settings. Run this script:

sudo mysql\_secure\_installation

During this process, you will:

* Set a **root password**.
* Remove anonymous users.
* Disallow root login remotely.
* Remove test databases.
* Reload privilege tables.

Answer the prompts according to your security preferences.

**5. Log In to MySQL**

Once installation is complete, you can log in to MySQL as the root user:

sudo mysql -u root -p

You will be prompted for the root password you set during the secure installation step.

**6. Verify MySQL Installation**

To check the status of the MySQL service and confirm it is running, use the following command:

sudo systemctl status mysql

### ****7. Change the MySQL Root Password****

Once you're logged in, you can change the password using the following SQL command:

ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql\_native\_password BY 'new\_password';

### ****8. Flush Privileges****

After altering the password, make sure to flush the privileges so that MySQL reloads the updated user credentials:

FLUSH PRIVILEGES;

To create a table in a MySQL

### 1. ****Log in to MySQL****

Open your terminal and log in to MySQL:

mysql -u root -p

Enter your root password when prompted.

### 2. ****Create the Database****

If you haven't already created the test database, do so with the following command:

CREATE DATABASE test;

### 3. ****Use the Database****

Switch to the test database:

USE test;

### 4. ****Create the Table****

Now, create a table named contacts with columns for id, name, contact\_number, and region. You can use the following SQL command:

CREATE TABLE contacts (

id INT AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(100) NOT NULL,

contact\_number VARCHAR(15) NOT NULL,

region VARCHAR(100) NOT NULL

);

### 5. ****Insert Sample Data****

You can insert some sample data into the contacts table using the following SQL commands:

INSERT INTO contacts (name, contact\_number, region) VALUES

('John Doe', '123-456-7890', 'North'),

('Jane Smith', '987-654-3210', 'South'),

('Alice Johnson', '555-123-4567', 'East'),

('Bob Brown', '555-987-6543', 'West');

### 6. ****Verify the Data****

To ensure the data has been inserted correctly, you can query the contacts table:

SELECT \* FROM contacts;

### 7. ****Exit MySQL****

After you’re done, you can exit the MySQL shell:

EXIT;

INSTALL MAILUTILS

**1. Install mailutils**

First, install the mailutils package to enable email notifications. Open a terminal and run the following command:

sudo apt update

sudo apt install mailutils

During the installation, you may be prompted to configure the email system. Choose an appropriate option based on your needs.

**2. Create a Backup Script**

Create a bash script to perform the backup of the test database and send a notification email upon success. Use a text editor to create the script:

nano /path/to/backup\_test\_db.sh

**3. Add the Backup Commands to the Script**

Add the following commands to the script:

#!/bin/bash

# Variables

DB\_NAME="test"

DB\_USER="your\_username"

DB\_PASS="your\_password"

BACKUP\_DIR="/path/to/backup/directory"

DATE=$(date +%Y-%m-%d\_%H-%M-%S)

BACKUP\_FILE="$BACKUP\_DIR/${DB\_NAME}\_backup\_$DATE.sql"

LOG\_FILE="/path/to/log/file.log"

EMAIL="your\_email@example.com"

# Create backup

mysqldump -u $DB\_USER -p$DB\_PASS $DB\_NAME > $BACKUP\_FILE

# Check if the backup was successful

if [ $? -eq 0 ]; then

echo "[$DATE] Backup successful: $BACKUP\_FILE" >> $LOG\_FILE

# Send email notification

echo -e "Subject: MySQL Backup Successful\n\nBackup for $DB\_NAME completed successfully on $DATE." | mail -s "MySQL Backup Notification" $EMAIL

else

echo "[$DATE] Backup failed for $DB\_NAME" >> $LOG\_FILE

# Send email notification about failure

echo -e "Subject: MySQL Backup Failed\n\nBackup for $DB\_NAME failed on $DATE." | mail -s "MySQL Backup Notification" $EMAIL

fi

**4. Modify the Script**

Make sure to replace the following placeholders with your actual values:

* your\_username: Your MySQL username.
* your\_password: Your MySQL password.
* /path/to/backup/directory: The directory where you want to save the backup.
* /path/to/log/file.log: The path where you want to save the log file.
* your\_email@example.com: Your email address to receive notifications.

**5. Make the Script Executable**

Make the script executable by running:

chmod +x /path/to/backup\_test\_db.sh

**6. Test the Backup Script**

Run the script to test if it works correctly:

/path/to/backup\_test\_db.sh

**7. Set up a Cron Job (Optional)**

If you want to automate the backup process, you can set up a cron job to run the script at a specified interval (e.g., daily). Open the crontab file:

crontab -e

Add the following line to run the backup script every day at midnight:

0 0 \* \* \* /path/to/backup\_test\_db.sh