

Linux Project

Final web interface:



Write a Note

Save Note

Saved Notes

2026-01-29 20:06:11

final check

2026-01-29 19:44:58

Second note test

2026-01-29 19:15:03

Terminal note test

Source code for the web app (Python):

```
GNU nano 8.1
from flask import Flask, request, redirect, render_template_string
import mysql.connector

app = Flask(__name__)

# Database configuration
db_config = {
    "host": "localhost",
    "user": "noteuser",
    "password": "notepass",
    "database": "notesdb"
}

HTML_TEMPLATE = """
<!DOCTYPE html>
<html>
<head>
    <title>Notes App</title>
</head>
<body>
    <h2>Write a Note</h2>
    <form method="POST">
        <textarea name="content" rows="4" cols="50" required></textarea><br><br>
        <button type="submit">Save Note</button>
    </form>

    <hr>

    <h2>Saved Notes</h2>
    {% for note in notes %}
    <div>
        <small>{{ note[2] }}</small>
        <p>{{ note[1] }}</p>
        <hr>
    </div>
    {% endfor %}
</body>
</html>

```

```

@app.route("/", methods=["GET", "POST"])
def index():
    conn = mysql.connector.connect(**db_config)
    cursor = conn.cursor()

    if request.method == "POST":
        content = request.form["content"]
        cursor.execute(
            "INSERT INTO notes (content) VALUES (%s)",
            (content,)
        )
        conn.commit()
        return redirect("/")

    cursor.execute(
        "SELECT id, content, created_at FROM notes ORDER BY created_at DESC"
    )
    notes = cursor.fetchall()

    cursor.close()
    conn.close()

    return render_template_string(HTML_TEMPLATE, notes=notes)

if __name__ == "__main__":
    app.run(host="0.0.0.0", port=80, debug=True)

```

Screenshots of the running app on EC2:

```

[root@ip-172-31-30-79 notes-app]# sudo python3 app.py
* Serving Flask app 'app'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment.
Use a production WSGI server instead.
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:80
* Running on http://172.31.30.79:80
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 109-076-865

```

MariaDB schema and tables:

```
MariaDB [(none)]> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| notesdb |
| performance_schema |
| sys |
+-----+
5 rows in set (0.000 sec)
```

```
MariaDB [(none)]> USE notesdb;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A
```

Database changed

```
MariaDB [notesdb]> SHOW TABLES;
```

```
+-----+
| Tables_in_notesdb |
+-----+
| notes |
+-----+
1 row in set (0.000 sec)
```

```
MariaDB [notesdb]> DESCRIBE notes;
```

```
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id | int(11) | NO | PRI | NULL | auto_increment |
| content | text | NO | | NULL | |
| created_at | timestamp | YES | | current_timestamp() | |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.001 sec)
```

```
MariaDB [notesdb]> SELECT * FROM notes;
```

```
+---+-----+-----+
| id | content | created_at |
+---+-----+-----+
| 1 | Terminal note test | 2026-01-29 19:15:03 |
| 2 | Second note test | 2026-01-29 19:44:58 |
| 3 | final check | 2026-01-29 20:06:11 |
+---+-----+-----+
3 rows in set (0.000 sec)
```

Configuration of mounted volume:

```
[root@ip-172-31-30-79 ~]# lsblk
NAME                MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
nvme0n1             259:0    0   10G  0 disk
├─nvme0n1p1         259:1    0    1M  0 part
├─nvme0n1p2         259:2    0  200M  0 part /boot/efi
└─nvme0n1p3         259:3    0   9.8G  0 part /
nvme1n1             259:4    0   10G  0 disk
```

Evidence of database backup:

```
[root@ip-172-31-30-79 ~]# df -h | grep backup  
/dev/nvme1n1 9.8G 32K 9.3G 1% /backup
```

```
[root@ip-172-31-30-79 ~]# ls -lh /backup  
total 20K  
drwx-----. 2 root root 16K Jan 29 20:03 lost+found  
drwxr-xr-x. 2 root root 4.0K Jan 29 20:04 mariadb
```

```
[root@ip-172-31-30-79 ~]# ls -lh /backup/mariadb  
total 4.0K  
-rw-r--r--. 1 root root 2.1K Jan 29 20:04 notesdb_2026-01-29_20-04-26.sql
```

```
[root@ip-172-31-30-79 ~]# head /backup/mariadb/notesdb_*.sql  
/*M!999999\ - enable the sandbox mode */  
-- MariaDB dump 10.19 Distrib 10.11.15-MariaDB, for Linux (x86_64)  
--  
-- Host: localhost Database: notesdb  
-- -----  
-- Server version 10.11.15-MariaDB  
  
/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;  
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;  
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
```

Deploy a Note-Taking Web Application on AWS EC2 with Backup Strategy

1. Objective

The objective of this project is to deploy a simple note-taking web application on an AWS EC2 instance using **Python**, connect it to a **MariaDB** database, and implement a **backup solution** using an additional **Amazon EBS volume**.

The application allows users to submit notes via a web interface, stores them with timestamps, and displays them in reverse chronological order.

2. System Overview

- **Cloud Provider:** Amazon Web Services (AWS)
 - **Compute Service:** EC2
 - **Operating System:** Red Hat Enterprise Linux 10
 - **Web Application:** Python (Flask)
 - **Database:** MariaDB
 - **Storage:** Amazon EBS (root volume + backup volume)
-

3. EC2 Instance Setup

- **Instance Type:** t2.micro
- **Operating System:** Red Hat Enterprise Linux 10
- **Security Group Configuration:**
 - Port **22** – SSH access
 - Port **80** – HTTP access
- **Authentication:** SSH key pair

The EC2 instance was successfully created and accessed using SSH.

4. Web Application Development

The web application was developed using **Python Flask**.

Features:

- A simple web form allowing users to write and submit notes
- Notes are stored in a MariaDB database
- Each note is automatically saved with a timestamp
- Saved notes are displayed below the input form
- The most recent note appears at the top

Application Behavior:

- HTTP POST is used to submit notes
 - HTTP GET is used to display existing notes
 - HTML is embedded directly inside the Python application
-

5. MariaDB Configuration

MariaDB was installed and configured on the EC2 instance.

Database Setup:

- **Database Name:** notesdb
- **Table Name:** notes

Table Schema:

```
CREATE TABLE notes (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    content TEXT NOT NULL,  
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP  
);
```

The web application connects to the database and performs insert and select operations successfully.

6. Backup Volume Setup

An additional **Amazon EBS volume** was created to store database backups.

Volume Configuration:

- **Volume Type:** General Purpose SSD (gp3)
- **Size:** 10 GiB
- **Encryption:** Enabled
- **Availability Zone:** Same as EC2 instance

Volume Mounting:

- The volume was formatted using ext4
- Mounted at:

/backup

Mount verification:

```
df -h | grep backup
```

7. MariaDB Backup Process

A backup process was implemented using mysqldump.

Backup Command:

```
mysqldump -u root -p notesdb > /backup/mariadb/notesdb_YYYY-MM-DD_HH-MM-SS.sql
```

Backup Verification:

```
ls -lh /backup/mariadb
```

This confirms that the database backup is successfully stored on the mounted EBS volume.

8. Testing and Validation







The following tests were performed:

- Verified application access via port 80
- Submitted multiple notes and confirmed persistence
- Confirmed timestamps are stored and displayed
- Verified newest notes appear first
- Verified database backups are created and stored under /backup
- Confirmed mounted volume availability

All tests passed successfully.

9. Deliverables

The following deliverables are provided:

-  Python source code for the web application
 -  Running application on EC2
 -  MariaDB database schema and tables
 -  Mounted backup volume at /backup
 -  Database backup files stored in /backup/mariadb
 -  This documentation (README)
-

10. Conclusion

This project successfully demonstrates the deployment of a web application on AWS EC2, database integration using MariaDB, and the implementation of a backup strategy using Amazon EBS.

The system meets all functional and technical requirements outlined in the project specification.