

# □ 1. Project Overview

□ **Title:** Deploy a Note-Taking Website on AWS EC2 with Backup Strategy

□ **Objective:**

Develop a **note-taking web application** using Python Flask and MariaDB, deploy it on **AWS EC2 (Red Hat 9)**, and implement a **daily backup strategy** to an external EBS volume.

---

## □ 2. Project Details

□ **Main Features:**

- Write and save notes through a web interface
- Store notes with **timestamp** in MariaDB
- Display all notes (latest first)
- Backup database daily to a separate mounted volume under `/oracle`

□ **Technology Stack:**

- **Python Flask** for web app
  - **MariaDB** as the database
  - **Gunicorn** as WSGI server
  - **EC2 (Red Hat Enterprise Linux 9)** as hosting environment
  - **EBS volume** for backups
- 

## □ 3. EC2 Setup

✓ Created **t2.micro EC2 instance** with:

- RHEL 9

- Opened ports: **22 (SSH), 80 (HTTP), 5000 (Flask testing)**
  - Connected using **MobaXterm**.
- 

## ❑ 4. Software Installation

### ❑ Installed:

```
sudo yum update -y
sudo yum install python3 mariadb-server -y
pip3 install flask pymysql gunicorn
```

### ❑ Started and enabled MariaDB:

```
sudo systemctl enable mariadb
sudo systemctl start mariadb
```

---

## ❑ 5. MariaDB Configuration

### ❑ Created database and table:

```
CREATE DATABASE notesdb;
USE notesdb;

CREATE TABLE notes (
    id INT AUTO_INCREMENT PRIMARY KEY,
    content MEDIUMTEXT,
    timestamp DATETIME
);
```

### ❑ Issue Faced:

- ❑ Initially faced **Access denied for user 'root'@'localhost'**.

### ❑ Solution:

✓ Ran:

```
ALTER USER 'root'@'localhost' IDENTIFIED BY 'YourPassword';
FLUSH PRIVILEGES;
```

## ❑ Issue Faced:

❑ Arabic text caused the app to crash.

## ❑ Solution:

✓ Changed table encoding:

```
ALTER TABLE notes CONVERT TO CHARACTER SET utf8mb4 COLLATE  
utf8mb4_unicode_ci;
```

---

## ❑ 6. Flask Application Code

❑ **app.py** simplified:

```
python

from flask import Flask, request, render_template
import pymysql
from datetime import datetime

app = Flask(__name__)

conn = pymysql.connect(host='localhost', user='root',
password='YourPassword', database='notesdb',
charset='utf8mb4')

@app.route('/', methods=['GET', 'POST'])
def index():
    cursor = conn.cursor()
    if request.method == 'POST':
        note = request.form['note']
        timestamp = datetime.now().strftime('%Y-%m-%d %H:%M:
%S')
        cursor.execute("INSERT INTO notes (content,
timestamp) VALUES (%s, %s)", (note, timestamp))
        conn.commit()
        cursor.execute("SELECT content, timestamp FROM notes
ORDER BY id DESC")
        notes = cursor.fetchall()
        return render_template('index.html', notes=notes)

if __name__ == '__main__':
    app.run(host='0.0.0.0', port=5000)
```

---

## ❑ 7. Issues Faced in Deployment

❑ **Issue:** `gunicorn.errors.HaltServer: Worker failed to boot.`

✓ **Solution:** Checked database credentials and ensured MariaDB was running.

---

❑ **Issue:** Port 5000 in use.

✓ **Solution:** Killed processes using:

```
ps aux | grep gunicorn  
kill -9 <PID>
```

---

❑ **Issue:** Website not accessible externally despite running on 0.0.0.0

✓ **Solution:** Opened port 5000 in EC2 Security Groups.

---

❑ **Issue:** Arabic input caused crashes

✓ **Solution:** Converted table and column to `utf8mb4`.

---

## ❑ 8. Running with Gunicorn

❑ Started using:

```
nohup python3 -m gunicorn -w 4 -b 0.0.0.0:5000 app:app &
```

❑ **Verified:**

- curl <http://127.0.0.1:5000> worked
  - Accessed via EC2 public IP: `http://54.175.63.29:5000`
-

## ❑ 9. Backup EBS Volume Setup

### ✓ Steps Followed:

1. Created new EBS volume
2. Attached to EC2
3. Formatted:

```
sudo mkfs.ext4 /dev/xvdb
```

4. Mounted:

```
sudo mkdir /oracle  
sudo mount /dev/xvdb /oracle
```

5. Persisted mount in /etc/fstab:

```
ini  
CopyEdit  
UUID=e06ba662-44ee-44b8-821a-0fed50e16c4f /oracle ext4  
defaults,nofail 0 2
```

### ❑ Verified:

```
df -h
```

---

## ❑ 10. Backup Script

### ❑ /oracle/backup\_mariadb.sh

```
#!/bin/bash  
mysqldump -u root -pYourPassword notesdb >  
/oracle/notesdb_backup_$(date +%F).sql
```

### ✓ Made executable:

```
chmod +x /oracle/backup_mariadb.sh
```

### ❑ Tested manually: Backup files appeared in /oracle.

---

## ☐ 11. Cron Job for Daily Backup

### ☐ Added to crontab:

```
0 2 * * * /oracle/backup_mariadb.sh
```

✓ Runs daily at **2AM**.

---

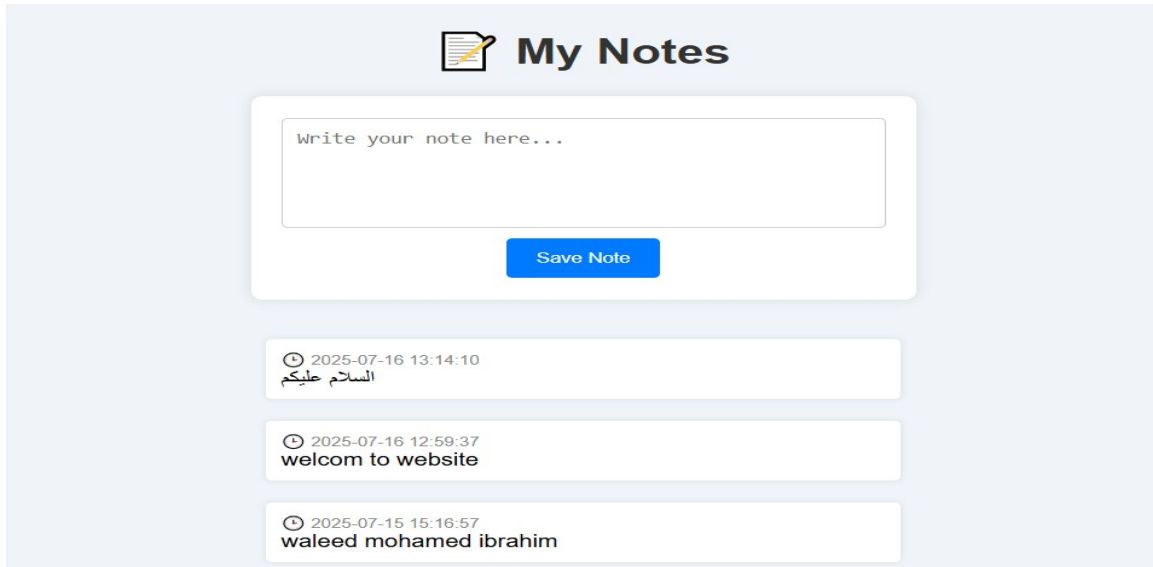
## ☐ 12. Final Verification

- ☐ Web app accessible via public IP
  - ☐ Notes saved to database and persisted
  - ☐ Arabic and English supported
  - ☐ Backup script tested and daily schedule confirmed
  - ☐ Volume mounted and data stored under /oracle
- 

## ☐ 13. Deliverables Checklist ☐

- ✓ Source code (app.py, index.html)
- ✓ Screenshots of app running
- ✓ MariaDB schema and tables
- ✓ Mounted volume config (/oracle)
- ✓ Backup files created
- ✓ **Full Documentation with issues & solutions (this file)**

THE OUTPUT OF WEBSITE :



## Coding to show this website :

### ❑ 1. لتأكيد التشغيل EC2 instance عرض الـ

```
hostname -I
```

```
output : 172.31.24.38
```

### ❑ 2. OS فحص حالة السيرفر والـ

```
uname -a
```

```
cat /etc/redhat-release
```

```
output :
```

```
[root@ip-172-31-24-38 ec2-user]# uname -a
cat /etc/redhat-release
Linux ip-172-31-24-38.ec2.internal 6.12.0-55.21.1.el10_0.x86_64 #1 SMP PREEMPT_DYNAMIC Mon Jul 7 02:36:42 EDT 2025 x86_64 GNU/Linux
Red Hat Enterprise Linux release 10.0 (Coughlan)
```

### ❑ 3. التحقق من تشغيل التطبيق (Gunicorn)

```
ps aux | grep gunicorn
```

```
output :
```

```
[root@ip-172-31-24-38 ec2-user]# ps aux | grep gunicorn
root      1767  0.0  2.4 31044 24204 ?        S    12:58   0:00 python3 -m gunicorn -w 4 -b 0.0.0.0:5000 app:app
root      1847  0.0  3.0 40344 30128 ?        S    13:06   0:00 python3 -m gunicorn -w 4 -b 0.0.0.0:5000 app:app
root      1849  0.0  3.0 40344 30128 ?        S    13:06   0:00 python3 -m gunicorn -w 4 -b 0.0.0.0:5000 app:app
root      2134  0.0  3.0 40344 30132 ?        S    13:34   0:00 python3 -m gunicorn -w 4 -b 0.0.0.0:5000 app:app
root      2191  0.0  3.0 40344 30132 ?        S    13:54   0:00 python3 -m gunicorn -w 4 -b 0.0.0.0:5000 app:app
root      2449  0.0  0.1  3848  1868 pts/3    S+   14:16   0:00 grep --color=auto gunicorn
```

#### □ 4. التأكد من المجلدات و الـ mount point

```
df -h
```

output :

```
[root@ip-172-31-24-38 ec2-user]# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/xvda3       9.8G  2.2G  7.6G  23% /
devtmpfs         4.0M    0  4.0M   0% /dev
tmpfs            476M    0  476M   0% /dev/shm
tmpfs            191M  3.7M  187M   2% /run
tmpfs            1.0M    0   1.0M   0% /run/credentials/systemd-journald.service
/dev/xvda2      200M  8.4M  192M   5% /boot/efi
tmpfs            1.0M    0   1.0M   0% /run/credentials/getty@tty1.service
tmpfs            1.0M    0   1.0M   0% /run/credentials/serial-getty@ttyS0.service
tmpfs           96M   4.0K   96M   1% /run/user/1000
/dev/xvddb       7.8G   36K   7.4G   1% /backup
```

#### □ 5. عرض محتوى /backup

```
ls -lh /backup
```

output :

```
[root@ip-172-31-24-38 ec2-user]# ls -lh /backup
total 28K
-rwxr-xr-x. 1 root root 105 Jul 15 15:01 backup_mariadb.sh
drwx----- 2 root root 16K Jul 15 14:55 lost+found
-rw-r--r-- 1 root root 2.2K Jul 15 15:05 notesdb_backup_2025-07-15.sql
-rw-r--r-- 1 root root 2.3K Jul 16 13:31 notesdb_backup_2025-07-16.sql
```

#### □ 6. للباك اب crontab إظهار

```
crontab -l
```



outout :

```
[root@ip-172-31-24-38 ec2-user]# crontab -l
0 2 * * * /oracle/backup_mariadb.sh
```

## □ 7. عرض MariaDB databases & tables

```
mysql -u root -p -e "SHOW DATABASES;"
```

```
mysql -u root -p -e "USE notesdb; SHOW TABLES; DESCRIBE notes;"
```

```
[root@ip-172-31-24-38 ec2-user]# mysql -u root -p -e "SHOW DATABASES;"
mysql -u root -p -e "USE notesdb; SHOW TABLES; DESCRIBE notes;"
Enter password:
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| notesdb |
| performance_schema |
| sys |
+-----+
Enter password:
+-----+
| Tables_in_notesdb |
+-----+
| notes |
+-----+
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id | int(11) | NO | PRI | NULL | auto_increment |
| content | mediumtext | YES | | NULL | |
| timestamp | datetime | YES | | NULL | |
+-----+-----+-----+-----+-----+-----+
[root@ip-172-31-24-38 ec2-user]#
```

Source code for website

```

[root@ip-172-31-24-38 notepad]# cat /root/notepad/app.py
from flask import Flask, request
import pymysql
from datetime import datetime

app = Flask(__name__)

# Database connection
conn = pymysql.connect(
    host='localhost',
    user='notepad',
    password='123456',
    db='notesdb',
    cursorclass=pymysql.cursors.DictCursor
)

@app.route('/', methods=['GET', 'POST'])
def index():
    with conn.cursor() as cursor:
        if request.method == 'POST':
            note = request.form['note']
            timestamp = datetime.now().strftime('%Y-%m-%d %H:%M:%S')
            sql = "INSERT INTO notes (content, timestamp) VALUES (%s, %s)"
            cursor.execute(sql, (note, timestamp))
            conn.commit()

            cursor.execute("SELECT * FROM notes ORDER BY id DESC")
            notes = cursor.fetchall()

    html = '''
    <html>
    <head>
        <title>My Notes App</title>
        <style>
            body {

```

```
        display: flex;
        justify-content: center;
        align-items: center;
        flex-direction: column;
        min-height: 100vh;
    }
    h1 { color: #333; }
    form {
        background: white;
        padding: 20px;
        border-radius: 8px;
        box-shadow: 0 0 10px rgba(0,0,0,0.1);
        width: 400px;
        text-align: center;
    }
    textarea {
        width: 100%;
        height: 100px;
        padding: 10px;
        border-radius: 4px;
        border: 1px solid #ccc;
        resize: none;
        font-size: 14px;
    }
    button {
        margin-top: 10px;
        padding: 10px 20px;
        background: #007BFF;
        color: white;
        border: none;
        border-radius: 4px;
        cursor: pointer;
    }
    button:hover {
        background: #0056b3;
    }
}
```

```

        .note {
            background: white;
            margin-top: 20px;
            padding: 10px;
            border-radius: 4px;
            box-shadow: 0 0 5px rgba(0,0,0,0.1);
            width: 400px;
        }
        .timestamp {
            font-size: 12px;
            color: #888;
        }
    </style>
</head>
<body>
    <h1>📝 My Notes</h1>
    <form method="POST">
        <textarea name="note" placeholder="Write your note here..."></textarea><br>
        <button type="submit">Save Note</button>
    </form>
    ...

    for n in notes:
        html += f'''
            <div class="note">
                <div class="timestamp">🕒 {n['timestamp']}</div>
                <div>{n['content']}</div>
            </div>
            ...

        html += '''
    </body>
</html>
'''

    return html

if __name__ == '__main__':
    app.run(host='0.0.0.0')

```