**Nginx:**

sudo apt-get install nginx # to install nginx

cat /etc/nginx/sites-available/todo\_app.conf #nginx config

sudo systemctlnginxad nginx

cat /var/log/nginx/access.log #to see nginx logs

**to access app server from web server:**

ssh -i YashwanthCloudZen.pem [ubuntu@10.0.129.153](mailto:ubuntu@10.0.129.153)

**App Server:**

cd TodoApp

cat app.py # to see python code

cat templates/index.html #to see HTML code

cat run\_app.sh #automation script to run python code

cat /etc/systemd/system/flaskapp.service #systemd configuration to start flask app at system start time.

**To access RDS from App server:**

mysql -u admin -p -h cloudzendb.ccrusy86s4fy.us-east-1.rds.amazonaws.com

**SQL commands:**

use CloudZenDatabase;

show databases;

**Create Table:**

CREATE TABLE tasks (id INT AUTO\_INCREMENT PRIMARY KEY, description TEXT NOT NULL, is\_done BOOLEAN DEFAULT FALSE, created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP);

see data:

select \* from tasks;

**For screenshot:**

sudo apt-get update

sudo apt-gevenv venv nginx

sudo apt-get install python3-pip python3-venv

sudo apt-get install mysql-client

pip install flask pymysql

python -m venv venv

Python code:

app = Flask(\_\_name\_\_)

# MySQL DB Config

DB\_CONFIG = {

'host': 'cloudzendb.ccrusy86s4fy.us-east-1.rds.amazonaws.com',

'user': 'admin',

'password': 'Admin123#',

'database': 'CloudZenDatabase',

'cursorclass': pymysql.cursors.DictCursor # To return rows as dictionaries

}

def get\_db\_connection():

return pymysql.connect(\*\*DB\_CONFIG)

@app.route('/')

def index():

conn = get\_db\_connection()

with conn.cursor() as cursor:

cursor.execute("SELECT \* FROM tasks ORDER BY created\_at DESC")

tasks = cursor.fetchall()

conn.close()

return render\_template('index.html', tasks=tasks)

@app.route('/add', methods=['POST'])

def add\_task():

desc = request.form.get('description')

if desc:

conn = get\_db\_connection()

with conn.cursor() as cursor:

cursor.execute("INSERT INTO tasks (description) VALUES (%s)", (desc, ))

conn.commit()

conn.close()

return redirect(url\_for('index'))

@app.route('/complete/<int:task\_id>')

def complete\_task(task\_id):

conn = get\_db\_connection()

with conn.cursor() as cursor:

cursor.execute("UPDATE tasks SET is\_done = TRUE WHERE id = %s", (task\_id ,))

conn.commit()

conn.close()

return redirect(url\_for('index'))

@app.route('/delete/<int:task\_id>')

def delete\_task(task\_id):

conn = get\_db\_connection()

with conn.cursor() as cursor:

cursor.execute("DELETE FROM tasks WHERE id = %s", (task\_id,))

conn.commit()

conn.close()

return redirect(url\_for('index'))

if \_\_name\_\_ == '\_\_main\_\_':

app.run(host="0.0.0.0", port=5000, debug=True)

html

app = Flask(\_\_name\_\_)

# MySQL DB Config

DB\_CONFIG = {

'host': 'cloudzendb.ccrusy86s4fy.us-east-1.rds.amazonaws.com',

'user': 'admin',

'password': 'Admin123#',

'database': 'CloudZenDatabase',

'cursorclass': pymysql.cursors.DictCursor # To return rows as dictionaries

}

def get\_db\_connection():

return pymysql.connect(\*\*DB\_CONFIG)

@app.route('/')

def index():

conn = get\_db\_connection()

with conn.cursor() as cursor:

cursor.execute("SELECT \* FROM tasks ORDER BY created\_at DESC")

tasks = cursor.fetchall()

conn.close()

return render\_template('index.html', tasks=tasks)

@app.route('/add', methods=['POST'])

def add\_task():

desc = request.form.get('description')

if desc:

conn = get\_db\_connection()

with conn.cursor() as cursor:

cursor.execute("INSERT INTO tasks (description) VALUES (%s)", (desc, ))

conn.commit()

conn.close()

return redirect(url\_for('index'))

@app.route('/complete/<int:task\_id>')

def complete\_task(task\_id):

conn = get\_db\_connection()

with conn.cursor() as cursor:

cursor.execute("UPDATE tasks SET is\_done = TRUE WHERE id = %s", (task\_id ,))

conn.commit()

conn.close()

return redirect(url\_for('index'))

@app.route('/delete/<int:task\_id>')

def delete\_task(task\_id):

conn = get\_db\_connection()

with conn.cursor() as cursor:

cursor.execute("DELETE FROM tasks WHERE id = %s", (task\_id,))

conn.commit()

conn.close()

return redirect(url\_for('index'))

if \_\_name\_\_ == '\_\_main\_\_':

app.run(host="0.0.0.0", port=5000, debug=True)

ubuntu@ip-10-0-129-153:~/TodoApp$ ^C

ubuntu@ip-10-0-129-153:~/TodoApp$ cat templates/index.html

<!doctype html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>TODO App</title>

<style>

body { font-family: sans-serif; max-width: 600px; margin: auto; }

.done { text-decoration: line-through; color: gray; }

.task { display: flex; justify-content: space-between; margin: 5px 0; }

</style>

</head>

<body>

<h1>📝 TODO List</h1>

<form action="/add" method="post">

<input type="text" name="description" placeholder="Enter a task..." required>

<button type="submit">Add</button>

</form>

<ul>

{% for task in tasks %}

<li class="task">

<span class="{{ 'done' if task.is\_done }}">{{ task.description }}</span>

<span>

{% if not task.is\_done %}

<a href="/complete/{{ task.id }}">✅</a>

{% endif %}

<a href="/delete/{{ task.id }}">🗑️</a>

</span>

</li>

{% endfor %}

</ul>

</body>

</html>

ubuntu@ip-10-0-129-153:~/TodoApp$ cat templates/index.html

<!doctype html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>TODO App</title>

<style>

body { font-family: sans-serif; max-width: 600px; margin: auto; }

.done { text-decoration: line-through; color: gray; }

.task { display: flex; justify-content: space-between; margin: 5px 0; }

</style>

</head>

<body>

<h1>📝 TODO List</h1>

<form action="/add" method="post">

<input type="text" name="description" placeholder="Enter a task..." required>

<button type="submit">Add</button>

</form>

<ul>

{% for task in tasks %}

<li class="task">

<span class="{{ 'done' if task.is\_done }}">{{ task.description }}</span>

<span>

{% if not task.is\_done %}

<a href="/complete/{{ task.id }}">✅</a>

{% endif %}

<a href="/delete/{{ task.id }}">🗑️</a>

</span>

</li>

{% endfor %}

</ul>

</body>

</html>