🚀 CRASHING POSTGRESQL BY ADJUSTING CONFIGURATION TO LOWER ITS TOLERANCE

STEP 1: LOGIN AS SUPERUSER INTO POSTGRESQL

1. Open Command Prompt as Administrator *(Win + R, type cmd, press Ctrl + Shift + Enter)*
2. *Connect to PostgreSQL with superuser (postgres):*

```
psql -U postgres -d location_db
```

3. If prompted, *enter the password for postgres.*

STEP 2: APPLY WEAK DATABASE CONFIGURATION

Run these SQL commands in psql:

```
ALTER SYSTEM SET max_connections = 5;

ALTER SYSTEM SET work_mem = '256kB';

ALTER SYSTEM SET shared_buffers = '8MB';

ALTER SYSTEM SET autovacuum = 'off';

ALTER SYSTEM SET checkpoint_timeout = '30s';

SELECT pg_reload_conf();
```

These changes weaken PostgreSQL, making it easier to crash under heavy load.

STEP 3: FIND YOUR POSTGRESQL SERVICE NAME

1. Open Command Prompt as Administrator
2. *Run:*

```
sc query state= all | findstr /I "postgres"
```

3. *You'll see something like*:

```
SERVICE_NAME: postgresql-x64-17

DISPLAY_NAME: postgresql-x64-17
```

4. Your service name is postgresql-x64-17 (or whatever is returned).

STEP 4: RESTART POSTGRESQL FOR CHANGES

1. Open Command Prompt as Administrator
2. Run:

```
net stop postgresql-x64-17

net start postgresql-x64-17
```

This restarts PostgreSQL with reduced settings.

STEP 5: RUN THE REQUEST FLOODING TEST

Run your flood script:

```
python main.py
```

STEP 6: CHECK IF OVERLOADED OR CRASHED

- The API may become unresponsive or return errors (e.g., 500 status codes).
- If there are too many active queries, the database is overloaded.
- If requests are failing or extremely slow, PostgreSQL is struggling.

To verify a complete crash, try a query like:

```
SELECT * FROM location_db;
```

If you see:

```
server closed the connection unexpectedly

This probably means the server terminated abnormally

before or while processing the request.

The connection to the server was lost. Attempting reset: Succeeded.
```

This strongly indicates PostgreSQL truly crashed or terminated.

STEP 7: RESET POSTGRESQL IF IT CRASHES

If PostgreSQL has become too slow or crashed:

In psql:

```
ALTER SYSTEM RESET ALL;
```

```
SELECT pg_reload_conf();
```

Then restart:

```
net stop postgresql-x64-17

net start postgresql-x64-17
```

Everything should revert to normal.

FINAL SUMMARY

- PostgreSQL is now very limited, making it easier to crash under load.
- If it doesn't crash, further lower max_connections (e.g., 3).
- Monitor pg_stat_activity and logs for errors.
- A connection error or abnormal termination means you've successfully crashed PostgreSQL.