

1. CLIENT WME-PG

DBParams.ini

[DB]

POSTGRES=1

[POSTGRESQL]

NumeServerul1=19x.xx.xx.11

NumeServerul2=19x.xx.xx.22

NumeServerul3=19x.xx.xx.33

DBName=postgres

ServerName=NumeServerul1

- NumeServerul1,2,3 sunt serverele la care vreți să vă logați din WME;
- ServerName=NumeServerul1 este serverul curent pe care lucrați;
- IP-urile trebuie setate la cele N aliasuri de server și trebuie să fie IP server de Postgres dorit.

Important: DBName="postgres" numele bazei de date ESTE CASE-SENSITIVE.

Nota: Pentru cei care vor să testeze pe același server versiuni diferite de PostgreSQL se pot instala și configura pe porturi diferite ca de ex:

Serverul1=19x.xx.xx.11:5432 *port default*

Serverul2=19x.xx.xx.11:5433 *port custom*

Același IP la server dar porturi diferite conform datelor din instalările de versiuni diferite de PostgreSQL, de ex PGv.14.0 pe 19x.xx.xx.11:5432 și PGv.14.1 pe 19x.xx.xx.11:5433.

Pentru acest caz trebuie două directoare pentru instalare de **ServerWME**, fiecare cu dbparams.ini configurat corespunzător serverului de PostgreSQL.

Deoarece în POSTGRES nu există ceva similar cu TNSNames.ORA, toate datele de logare sunt în acest fișier INI.

DBParams.ini se pune în folderul cu WMEnterprise.

2. SERVER WME-PG

Postgresql.conf este fișierul de parametri de pornire Postgres și este numai pentru server.

Fișierul text *postgresql.conf* din kit conține setările pentru buna funcționare a Postgres, cu observația că

datele referitoare la configurație hardware a serverului trebuie însă descrise după instalare (și la fiecare modificare de configurație).

Accesați link <https://pgtune.leopard.in.ua/#/> în ideea de a obține parametrii specifici serverului vostru.

Apasa butonul „Generate”. În partea dreaptă vor apărea câțiva parametri și valorile propuse.

Recomandarea inițială ar fi generarea configurației cu opțiunea DBType = Online Transaction Processing (oltp).

De asemenea pot fi luate în calcul și opțiunile Mixed type of application (server comun pentru BD și Aplicație) sau Data Warehouse (dw) (în funcție de modul de folosire al aplicației, predominant

rapoarte, etc).

Home How it works

PGTune

Parameters of your system

DB version what is this?
15

OS Type what is this?
Windows

DB Type what is this?
Online transaction processing system

Total Memory (RAM) what is this?
32 GB

Number of CPUs what is this?
8

Number of Connections what is this?
100

Data Storage what is this?
SSD storage

Generate

postgresql.conf ALTER SYSTEM

Add/modify this settings in **postgresql.conf** and restart database

```
# DB Version: 15
# OS Type: windows
# DB Type: oltp
# Total Memory (RAM): 32 GB
# CPUs num: 8
# Connections num: 100
# Data Storage: ssd

max_connections = 100
shared_buffers = 8GB
effective_cache_size = 24GB
maintenance_work_mem = 2047MB
checkpoint_completion_target = 0.9
wal_buffers = 16MB
default_statistics_target = 100
random_page_cost = 1.1
work_mem = 20971kB
min_wal_size = 2GB
max_wal_size = 8GB
max_worker_processes = 8
max_parallel_workers_per_gather = 4
max_parallel_workers = 8
max_parallel_maintenance_workers = 4
```

Copy configuration

Nota:

La parametrul **Number of Connections** se punea 100 default: “**max_connections=100**” (se da **restart la servicii PostGre**) dar pentru un tuning mai bun acest paramentru se poate schimba in functie de nr. de utilizatori WME-PG, astfel:

- a. Intre 1 si 5 utilizatori: **max_connections=40**
- b. Intre 5 si 10 utilizatori: **max_connections=60**
- c. Intre 10 si 20 utilizatori: **max_connections=80**
- d. Intre 20 si 50 utilizatori: **max_connections=100**
- e. Peste 50 utilizatori: **max_connections=100** si se va stabili de la caz la caz, in functie de nr. de utilizatori peste 50.

Copy Configuration si editati **Postgresql.conf** la final de lista de parametri, dupa linia “**# Add settings for extensions here**”.

```

-----
# CUSTOMIZED OPTIONS
#-----

# Add settings for extensions here

# DB Version: 15
# OS Type: windows
# DB Type: oltp
# Total Memory (RAM): 32 GB
# CPUs num: 8
# Connections num: 100
# Data Storage: ssd

max_connections = 100
shared_buffers = 8GB
effective_cache_size = 24GB
maintenance_work_mem = 2047MB
checkpoint_completion_target = 0.9
wal_buffers = 16MB
default_statistics_target = 100
random_page_cost = 1.1
work_mem = 20971kB
min_wal_size = 2GB
max_wal_size = 8GB
max_worker_processes = 8
max_parallel_workers_per_gather = 4
max_parallel_workers = 8
max_parallel_maintenance_workers = 4
    
```

Acești parametri vor fi actualizați în fișierul de configurare *postgresql.conf* dat de noi pe care tocmai l-ați copiat peste cel creat by default la instalarea Postgres.

Nota:

La “Total Memory (RAM)” **daca** aveți pus 32GB **aparea** sugestia la parametrul: `maintenance_work_mem = 2GB`.

Schimbati acest parametru `maintenance_work_mem = 2047MB` **daca este cazul.**

Indiferent de cat RAM aveti pe server si vi se sugereaza la acest parametru, acesta **nu poate depasi 2047MB.**

In PostgreSQL, managementul tabelor temporare este diferit de Oracle, in sensul ca acestea sunt create pentru fiecare sesiune noua. Din acest motiv tabela de sistem `pg_attribute` si indecsii acesteia, in anumite cazuri cresc intr-un mod alarmant. Aceasta crestere a acestei tabeli afecteaza in timp performanta bazei de date PostgreSQL.

Pentru a ajuta la menținerea performanței bazei de date și a eficienței spațiului prin eliminarea rândurilor care nu mai sunt necesare trebuie facuta o mentenanta periodica a BD numita **VACUUM**.

Trebuie precizat exista **AUTOVACUUM** care este activat implicit în PostgreSQL și poate fi configurat folosind mai mulți parametri în fișierul *postgresql.conf*.

1. `autovacuum`: activează sau dezactivează procesul de fundal de autovacuum.
2. `autovacuum_vacuum_threshold`: determină numărul minim de rânduri moarte care trebuie să fie prezente într-un tabel înainte de a fi curatat. Valoarea implicită este 50.
3. `autovacuum_analyze_threshold`: determină numărul minim de rânduri active care trebuie să fie prezente într-un tabel înainte de a fi analizat. Valoarea implicită este 50.
4. `autovacuum_vacuum_scale_factor`: un multiplicator care determină câte rânduri moarte sunt necesare pentru a declanșa un vacuum în funcție de dimensiunea tabelului. Valoarea implicită este 0,2.

5. `autovacuum_analyze_scale_factor`: multiplicator care determină câte rânduri active sunt necesare pentru a declanșa o analiză în funcție de dimensiunea tabelului. Valoarea implicită este 0,1.

6. `autovacuum_vacuum_cost_delay`: timpul (în milisecunde) în care AUTOVACUUM îl va aștepta înainte de a începe o operație de VACUUM. Valoarea implicită este 20.

7. `autovacuum_vacuum_cost_limit`: numărul maxim de rânduri care pot fi curățate într-o singură operațiune de VACUUM. Valoarea implicită este 200.

EXEMPLU:

`autovacuum = on`

`autovacuum_vacuum_threshold = 100`

`autovacuum_analyze_threshold = 100`

`autovacuum_vacuum_scale_factor = 0.5`

`autovacuum_analyze_scale_factor = 0.2`

`autovacuum_vacuum_cost_delay = 50`

`autovacuum_vacuum_cost_limit = 500`

Este important să configurați aceste setări pentru a vă asigura că vacuum și analiza corect rulează eficient și nu provoacă o încărcare prea mare a bazei de date. De asemenea, este o idee bună să monitorizați activitatea autovacuumului și să faceți VACUUM manual al tabelor care nu sunt întreținute în mod adecvat de către autovacuum.

VACUUM

În PostgreSQL, ori de câte ori rândurile dintr-un tabel sunt șterse (DELETE), rândul (tuplul) existent este marcat ca mort (nu va fi eliminat fizic) și în timpul unei actualizări (INSERT), marchează tuplul de ieșire corespunzător ca mort și inserează un nou tuplu, deoarece în PostgreSQL operațiunile UPDATE = DELETE + INSERT.

Aceste tupluri moarte consumă spațiu de stocare inutil și, în cele din urmă, aveți o bază de date PostgreSQL umflată.

VACUUM recuperează spațiu și îl face disponibil pentru reutilizare, dar spațiul suplimentar nu este returnat sistemului de operare este doar păstrat disponibil pentru reutilizare în cadrul aceluiași tabel.

Balonarea (umflare) tabelor cu tuple moarte afectează serios performanța interogărilor PostgreSQL, deoarece tabelele și indecșii sunt stocate ca matrice de pagini cu dimensiune fixă. Ori de câte ori o interogare solicită rânduri, instanța PostgreSQL încarcă aceste pagini în memorie, iar rândurile moarte provoacă îngreunarea operațiilor I/O pe disc în timpul încărcării datelor.

VACUUM FULL rescrie întregul conținut al tabelului într-un fișier de disc nou, fără spațiu suplimentar, permițând ca spațiul neutilizat să fie returnat sistemului de operare.

Acest mod este mult mai lent și necesită o blocare exclusivă pe fiecare tabel în timp ce este procesat.

- a. **Pg_hba.conf** este fișierul de autentificare a clienților PG și conține ip-uri care au acces la serverul de postgre, trebuie editat, de ex:

IPv4 local connections (0.0.0.0/0 permite oricărui IP să se conecteze la serverul de PG) :

```
host all all 0.0.0.0 md5
```

Note:

- a. O problema care merita atentie la importul de date in PG, este dimensiunea foarte mare a fisierelor de tip LOG.
- b. De retinut ca fisierele .log pot fi sterse manual sau de un job care ruleaza la un anumit interval de timp.

Dar pentru importul datelor sau pentru alte operatiuni care genereaza fisiere de log foarte mari, ar fi indicat ca in fisierul **postgresql.conf** parametrul sa fie pe **off**:

logging_collector = off (necesita restart la BD) , astfel incat sa nu se mai genereze fisiere tip log.

Dupa terminarea operatiunilor de import date ar fi bine sa se genereze totusi fisiere log, de exemplu cate un fisier pentru fiecare zi a saptamanii cu suprascrierea lor la fiecare 7 zile. Pentru aceasta trebuie configurati in **postgresql.conf** urmatorii parametri astfel:

```
log_destination = 'stderr'  
logging_collector = on  
log_directory = 'log'  
log_filename = 'postgresql-%a.log'  
log_truncate_on_rotation = on  
log_rotation_age = 1d  
log_rotation_size = 0
```

Daca cineva considera ca nu sunt necesare fisierele log, atunci se poate seta in **postgresql.conf** doar:

```
logging_collector = off
```

- c. Atat **Postgresql.conf** cat si **Pg_hba.conf** se configureaza si se pun in dir.:
\\PostgreSQL\15\data !

Se restarteaza serviciul de postgres sau restart server, dupa orice modificare a acestor fisiere de config!