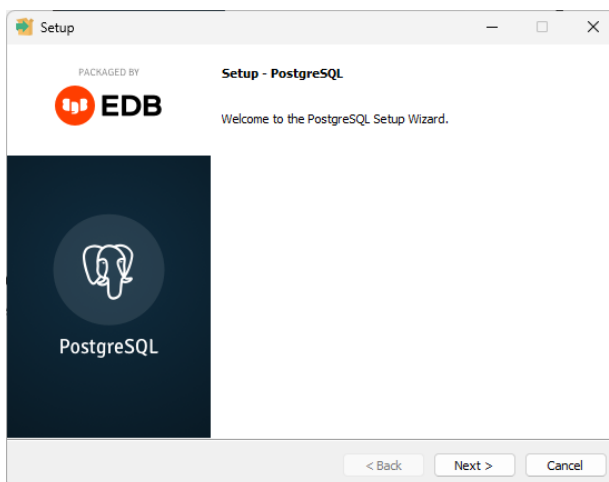
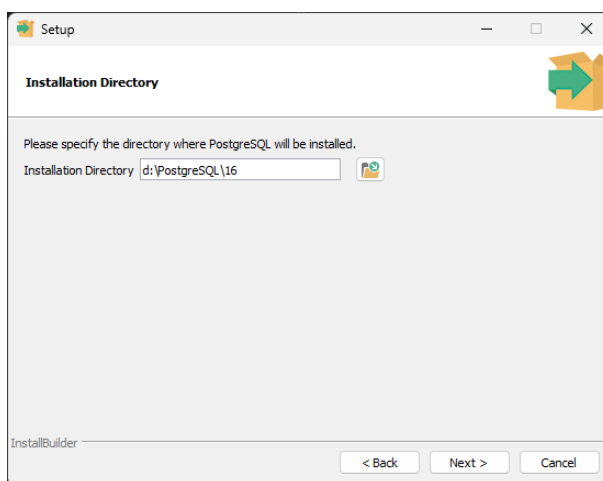


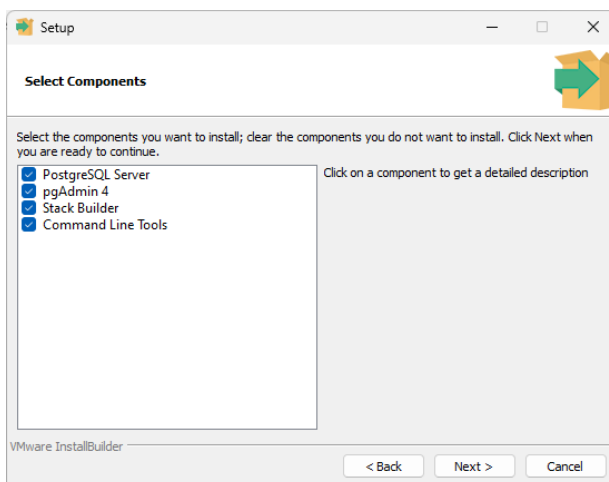
## Instalare Server PostgreSQL v.16:



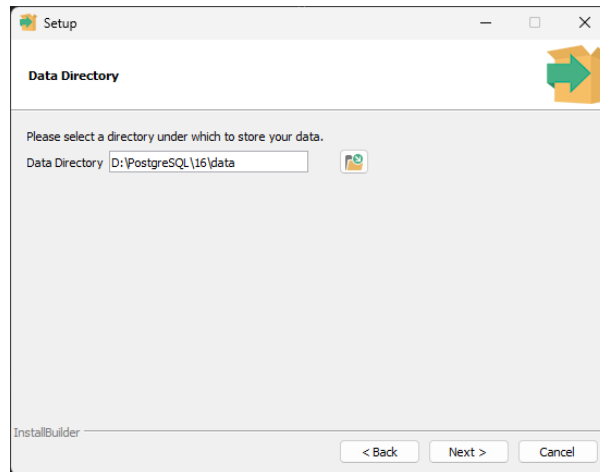
Se alege calea unde se instaleaza Serverul de PostgreSQL (by default este C:\Program Files, dar de preferat este sa fie pe o alta partitie diferita de C:\ din motive care tin de unele drepturi de acces si administrative, cum ar fi reinstalările de Windows):



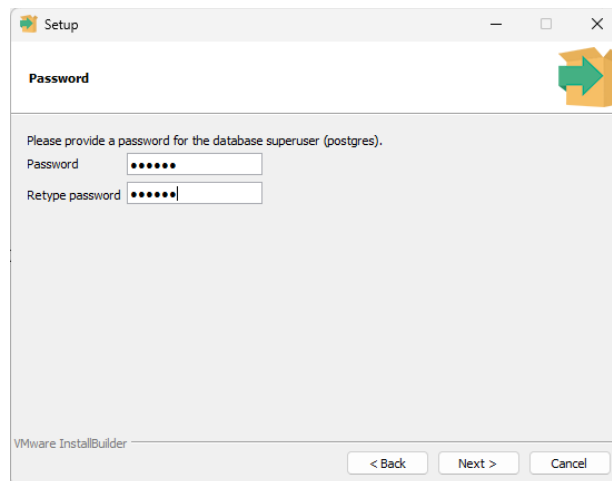
Se bifeaza componentele pentru instalare:



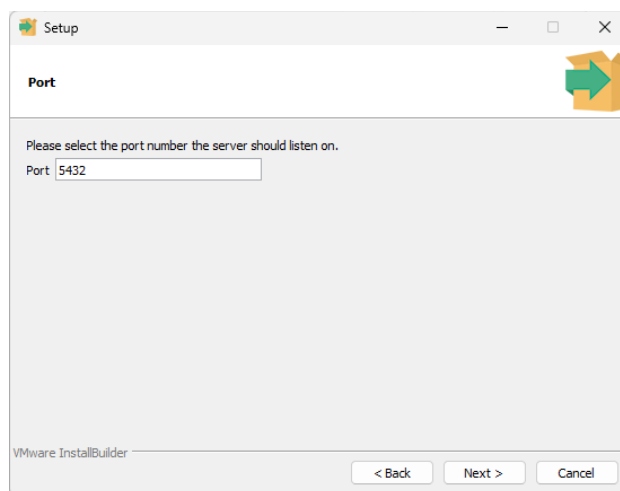
Se alege locatia bazei de date PostgreSQL (by default este in C:\Program Files, dar de preferat este sa fie pe o alta partitie diferita de C:):



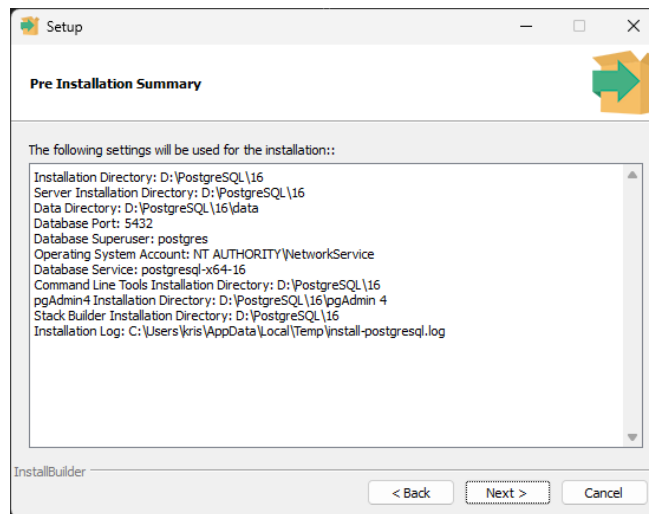
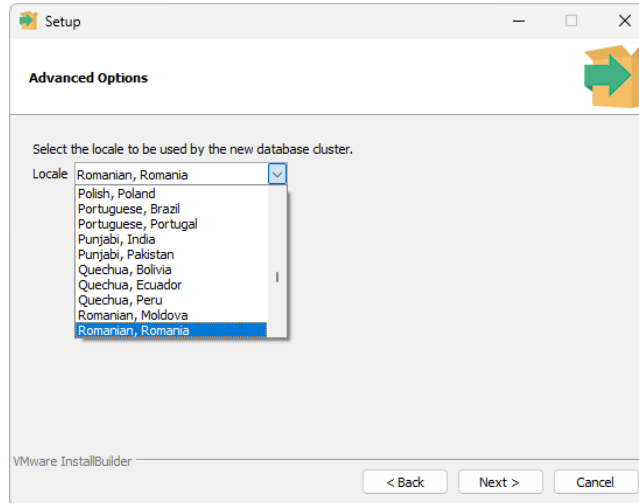
Se alege o parola pentru administratorul bazei de date PostgreSQL (user default: postgres)



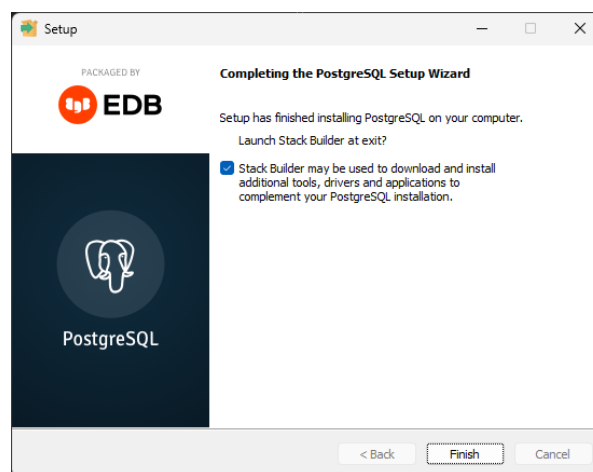
Portul serverului default este 5432:

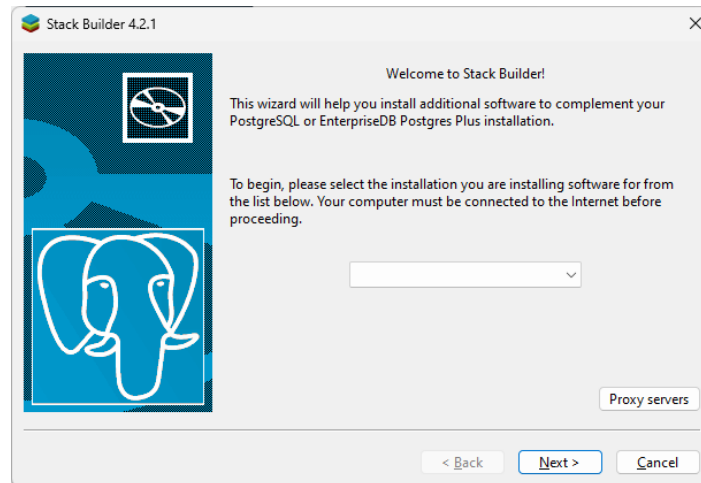


Se aleg parametrii care definesc setarile userului de regiune/limba etc.:

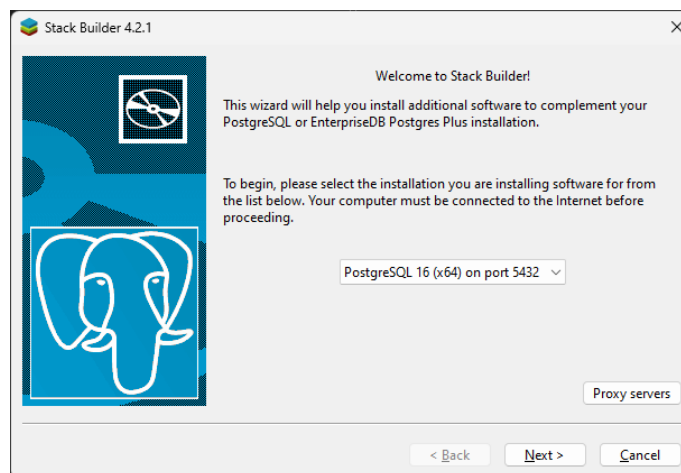


**PG Agent (job scheduler) se instaleaza prin Stack Builder, bifat la inceput de instalare PostgreSQL:**

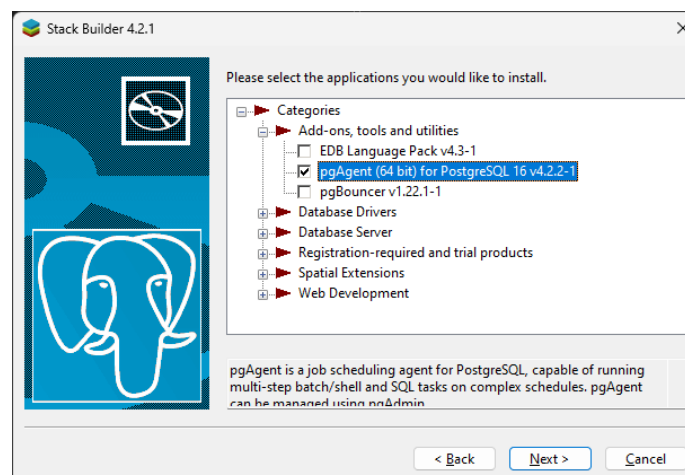


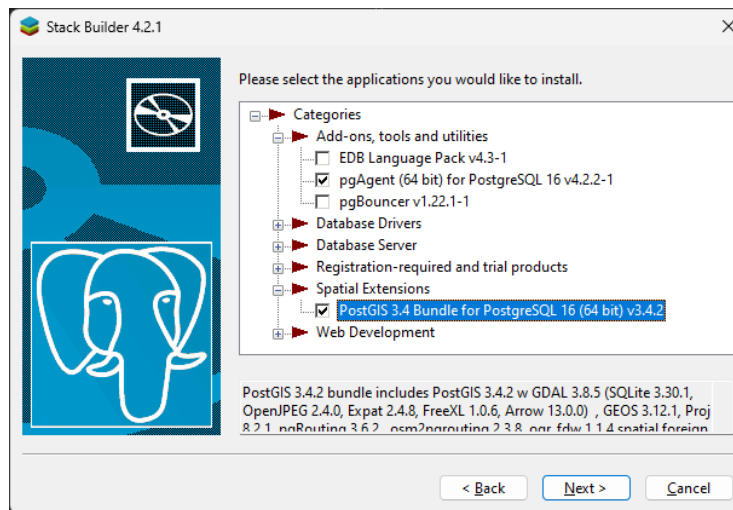


Se alege serverul de PG:

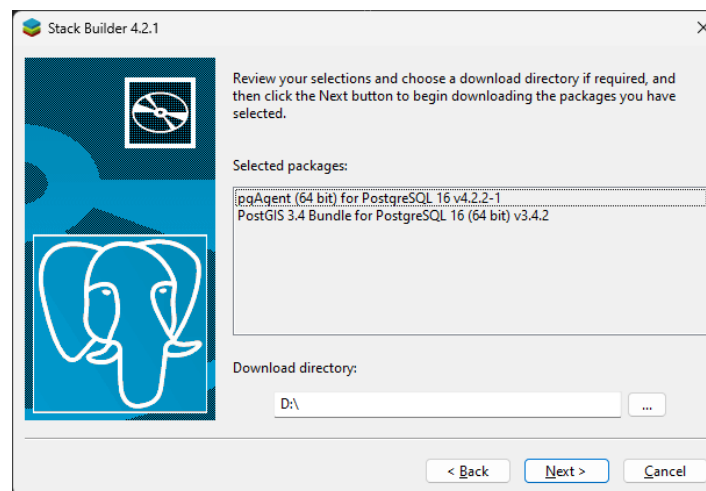


Se bifeaza pgAgent sub *Add-ons, tools and utilities*:

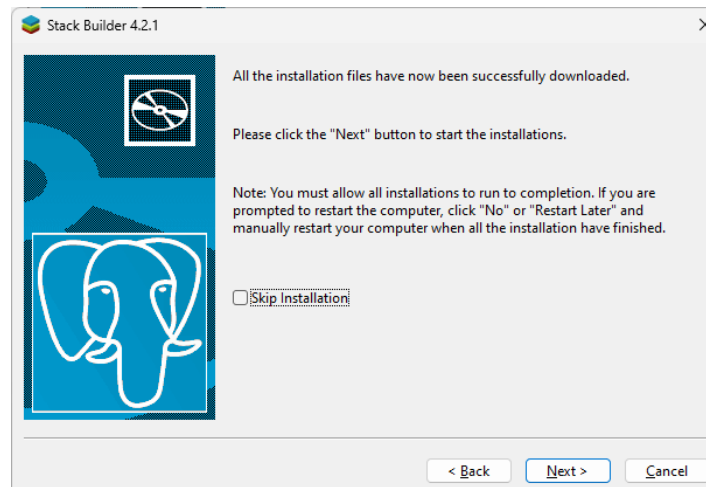




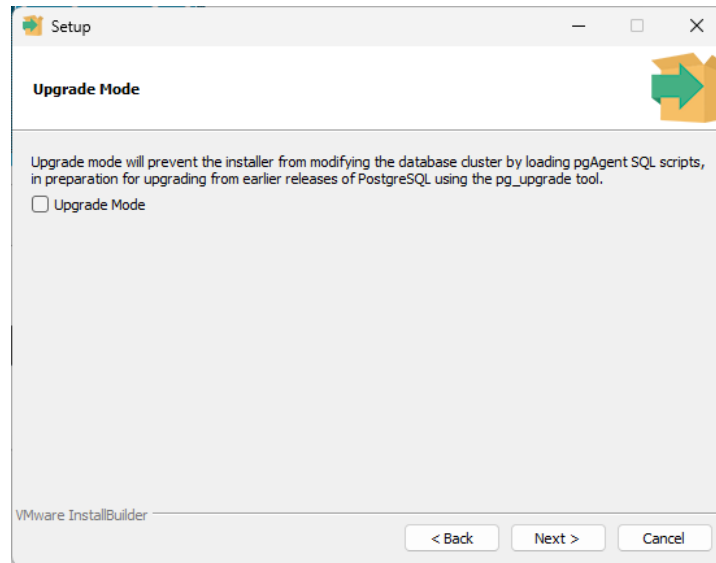
**Nota:** aveti nevoie de conexiune la internet pe serverul unde se face instalarea:



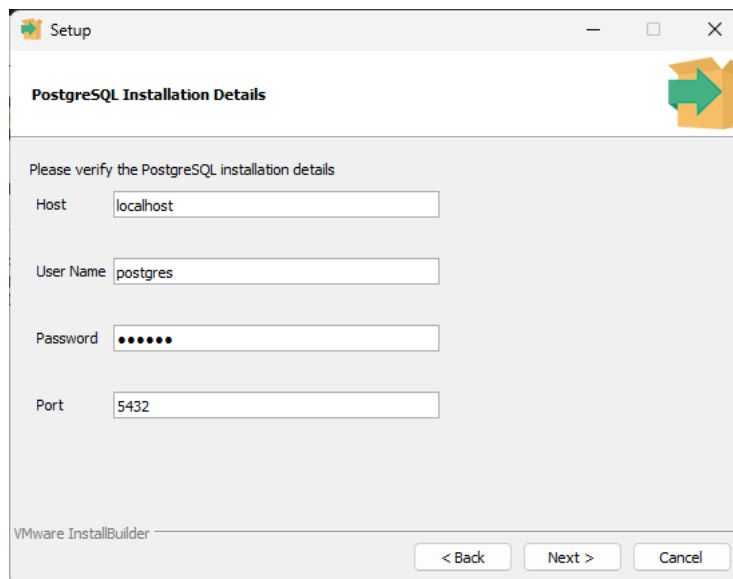
Next:



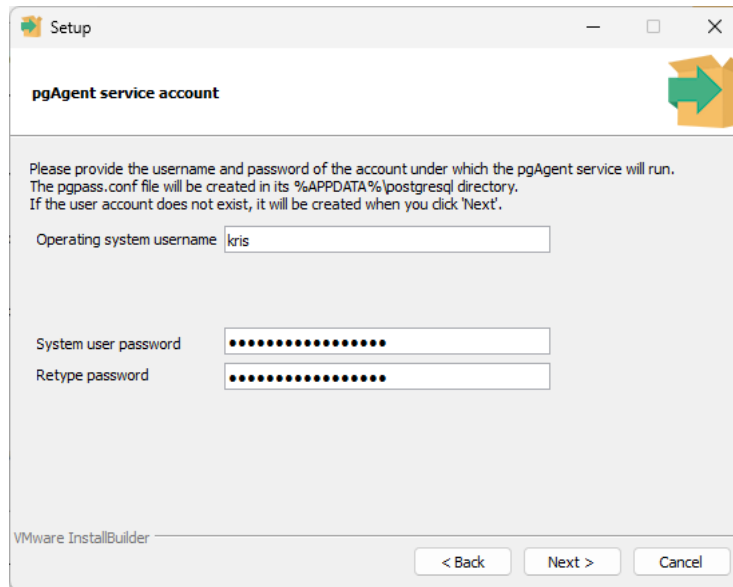
Next:



Se completeaza cu datele serverului si user-ul administrator al bazei de date PG (postgres):



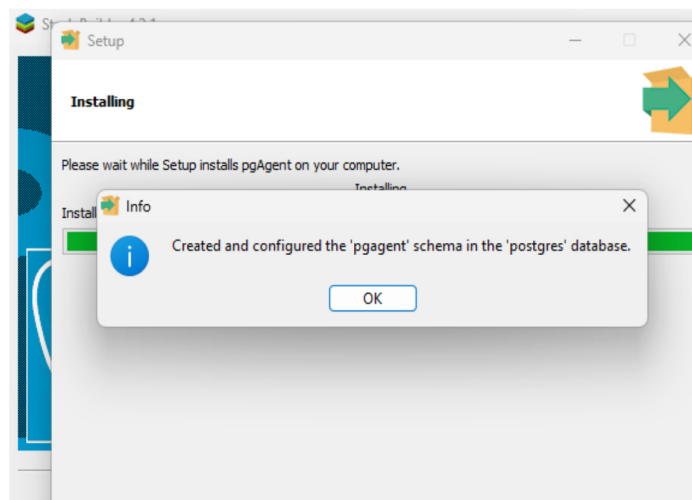
Se alege un user de sistem sub care pgAgent is va crea si rula serviciul:

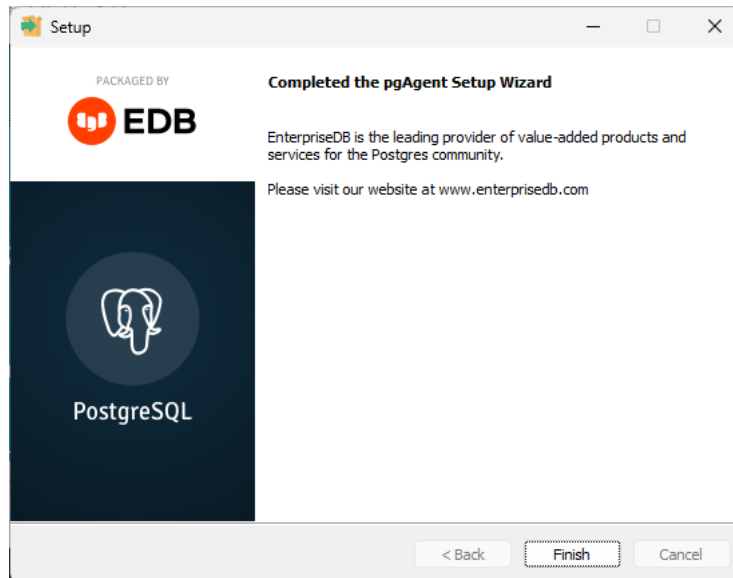


**Nota:** Au fost cazuri pe diferite sisteme de operare unde dacă s-a ales opțiunea de a crea din această fereastră un user nou, sub care să ruleze serviciul, nu s-a creat structura %APPDATA% sub care să pornească acest serviciu.

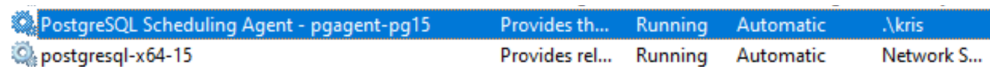
**Recomandarea noastră este să folosiți un user existent cu drepturi de administrator care a fost logat pe acel sistem (și care să nu fie cu drepturi de administrator moștenite adică făcând parte de fapt dintr-un grup separat, grup care are este pus la rândul său în Grupul de Administratori).**

**Dacă userul ales de Dvs. nu are parola, trebuie setată o parola, este obligatorie pentru pgAgent.**





Dupa instalare se recomanda un restart si se va verifica ca cele 2 servicii sa fie pornite:



Urmatorul pas este configurarea celor doua fisiere de config: **postgresql.conf** si **pg\_hba.conf** care se gasesc in directorul de instalare (ex.: d:\PostgreSQL\16\data)

- a. **Postgresql.conf** este fisierul de parametri de pornire Postgres si este exclusiv pentru server.

Fisierul text **postgresql.conf din kit** ([https://download2.winmentor.ro/WMEnterprise/Tools/01%20-%20Postgre Instalare/PostgreSQL%20v16/03.Fisiere%20config%20postgres/](https://download2.winmentor.ro/WMEnterprise/Tools/01%20-%20Postgre%20Instalare/PostgreSQL%20v16/03.Fisiere%20config%20postgres/)) **contine setarile pentru buna functionare a Postres, cu observatia ca datele referitoare la configuratie hardware a serverului trebuie insa descrisa dupa instalare (si la fiecare modificare de configuratie).**

El se de copie din kitul nostru in directorul unde ati facut Dvs. instalarea, peste fisierul default de la instalare (ex.: d:\PostgreSQL\16\data\postgresql.conf).Urmatorul pas este sa accesati link-ul <https://pgtune.leopard.in.ua/#/> in ideea de a obtine parametrii specifici serverului Dvs.. Apasa butonul „Generate”. În partea dreapta vor apărea câțiva parametri și valorile propuse.

**Recomandarea initiala** ar fi generarea configurarea cu optiunea DBType = Online Transaction Processing (oltp).

De asemenea pot fi luate in calcul si optiunile Mixed type of application (server comun pentru BD si Aplicatie) sau Data Warehouse (dw) (in functie de modul de folosire al aplicatiei, predominant rapoarte, etc). **Exemplu, se aleg datele corespunzatoare serverului si versiunii PG de instalat:**



**Nota:**

La parametrul **Number of Connections** se pune 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*”.

```

postgresql.conf
File Edit View postgresql.conf
# Add settings for extensions here

# DB Version: 16
# 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
huge_pages = try
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 actualizati în fișierul de configurare *postgresql.conf* dat de noi pe care tocmai l-ati copiat peste cel creat by default la instalarea Postgres.

**Nota:**

La “Total Memory (RAM)” **daca** aveati 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 tabele 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 curatate 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 tabelelor 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) tabelelor 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 îngreuneaza operațiile 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.

- b. **Pg\_hba.conf** este fișierul de autentificare a clienților PG și conține ip-uri care au acces la serverul de postgres, trebuie editat, de ex:

# IPv4 local connections (0.0.0.0/0 permite oricarui IP sa 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** urmasorii 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\16\data !

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

### **Extensie HTTP pt. vers PostgreSQL 15:**

Urmatorul pas este copierea unei **Extensii PG pentru Request http (folosit de ex. pentru comunicarea cu WMEWebService)** din fisierul tip arhiva aflat in directorul PG\_HTTP:

[https://download2.winmentor.ro/WMEnterprise/Tools/01%20-%20Postgre\\_Instalare/PostgreSQL%20v16/06.PG\\_agent%20si%20PG\\_http/PG\\_HTTP/](https://download2.winmentor.ro/WMEnterprise/Tools/01%20-%20Postgre_Instalare/PostgreSQL%20v16/06.PG_agent%20si%20PG_http/PG_HTTP/)

Acesta versiune de extensie http este compatibila doar cu versiunea **15** de PostgreSQL.

a. se copie directoarele/fisierele in directoarele corespunzatoare din directorul unde este instalat PostgreSQL (bin, lib, share, ssl)

b. Se instaleaza DBever :

[https://download2.winmentor.ro/WMEnterprise/Tools/01%20-%20Postgre\\_Instalare/PostgreSQL%20v16/05.DBever%20-%20pentru%20interogare%20BD/dbeaver-ce-23.1.1-x86\\_64-setup.exe](https://download2.winmentor.ro/WMEnterprise/Tools/01%20-%20Postgre_Instalare/PostgreSQL%20v16/05.DBever%20-%20pentru%20interogare%20BD/dbeaver-ce-23.1.1-x86_64-setup.exe)

c. din DBever (conectat la serverul PG) se executa:

```
set search_path to public;  
CREATE EXTENSION http;
```

## **INSTRUCTIUNI instalare WME-PG**

### **1. Creati folder de WME dupa cum stiti de la instalarea de WME-ORACLE.**

Copiatii dbparams.ini (trebuie configurat de Dvs.) si Installer-ul WME impreuna cu WinMService si Nethasp.ini (trebuie configurat de Dvs.) in folderul creat de Dvs de WME. Link kit:

[https://download2.winmentor.ro/WMEnterprise/Tools/01%20-%20Postgre\\_Instalare/PostgreSQL%20v16/04.Installer%20WME%20-PG/](https://download2.winmentor.ro/WMEnterprise/Tools/01%20-%20Postgre_Instalare/PostgreSQL%20v16/04.Installer%20WME%20-PG/)

Se copie dll-urile din [https://download2.winmentor.ro/WMEnterprise/Tools/01%20-%20Postgre\\_Instalare/PostgreSQL%20v16/02.Dll-ri%20pentru%20clientul%20de%20postgres/](https://download2.winmentor.ro/WMEnterprise/Tools/01%20-%20Postgre_Instalare/PostgreSQL%20v16/02.Dll-ri%20pentru%20clientul%20de%20postgres/)

Rulati WMInstaller\_PG.exe in directorul creat de Dvs. (alaturi de PGmartor.ZIP si PGservice.ZIP) pentru a crea structura bazei de date WME si configurarea sa, asa cum este si la instalarea de Oracle.

### **2. Pe statiile client nu se intaleaza PostgreSQL. Pe statie client PG:**

a. Se copie dll-urile din [https://download2.winmentor.ro/WMEnterprise/Tools/01%20-%20Postgre\\_Instalare/PostgreSQL%20v16/02.Dll-ri%20pentru%20clientul%20de%20postgres/](https://download2.winmentor.ro/WMEnterprise/Tools/01%20-%20Postgre_Instalare/PostgreSQL%20v16/02.Dll-ri%20pentru%20clientul%20de%20postgres/)

b. se configureaza nethasp.ini, Protect.dat si dbparams.ini in dir. de WME.

c. Se instaleaza driverul de cheie: <http://ftp.winmentor.ro/WMEnterprise/Tools/hasp/>

d. Se dezarhiveaza din DLL\_pt\_AnafConnector.zip (32b sau 64b), chkVIES.zip in dir. de WME.

e. Se copie executabilele de WME.

### **3. Copiati exe noi de WME din dir. cu Versiunea xx.xxx, cand este cazul !**

<https://download2.winmentor.ro/WMEnterprise/Versiuni%20WME/>

**Engage!**