

# Introducere Oracle database folosind Docker

Puşcaşu Felix

10 Decembrie 2019

## Contents

<b>1</b>	<b>Introducere</b>	<b>2</b>
<b>2</b>	<b>Instalare Docker</b>	<b>2</b>
<b>3</b>	<b>Configurare docker desktop</b>	<b>3</b>
<b>4</b>	<b>Pornirea şi oprirea serverului</b>	<b>3</b>
<b>5</b>	<b>Metode de conectare şi utilizare</b>	<b>5</b>
5.1	Direct in terminal . . . . .	5
5.2	Acces direct Sql*Plus . . . . .	5
5.3	SQL Developer . . . . .	6
<b>6</b>	<b>Crearea unui user</b>	<b>7</b>
<b>7</b>	<b>Concluzie</b>	<b>7</b>

# 1 Introducere

Acest referat are scopul descrierii instalării unei baze de date Oracle 12c pe mașina locală și o mică introducere în gestiunea utilizatorilor într-o schemă de baze de date. Avantajul izolării acestui server într-un container docker ajută în două moduri:

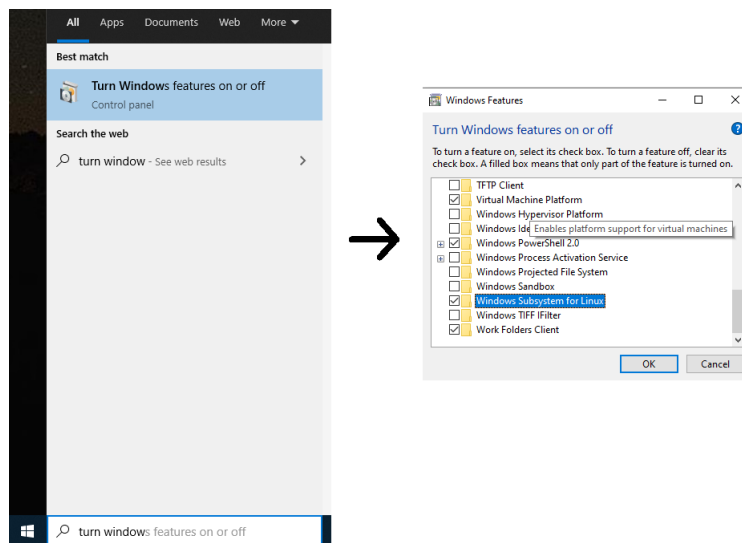
- Se garantează faptul că serverul se va comporta exact la fel indiferent unde este instalat.
- Putem avea mai multe baze de date în paralel, sau alte servicii în paralel, și datorită izolării într-un container avem o siguranță că diferitele procese nu vor afecta baza de date instalată. Pe scurt, este ca și cum am avea un alt computer care are ca singur scop rularea server-ului.

# 2 Instalare Docker

Docker Desktop poate fi downloadat pentru [Windows](#) sau [MacOS](#). Deși docker poate fi instalat prin mai multe metode sau doar diferite componente din el, pentru scopul nostru vom folosi *Docker Desktop*. [Aici](#) se găsesc instrucțiuni pentru diferite distribuții linux.

După ce docker este instalat, puteți verifica versiunea cu linia: `docker --version` și puteți testa rapid dacă totul funcționează folosind comanda: `docker run hello-world`

**Nota:** Pentru windows, docker poate să depindă de serviciul WSL. Acesta se activează căutând "Turn windows features on or off", și activând Windows Subsystem for Linux, după care instalăm și un update pentru acest serviciu de la site-ul oficial Microsoft, [aici](#)



### 3 Configurare docker desktop

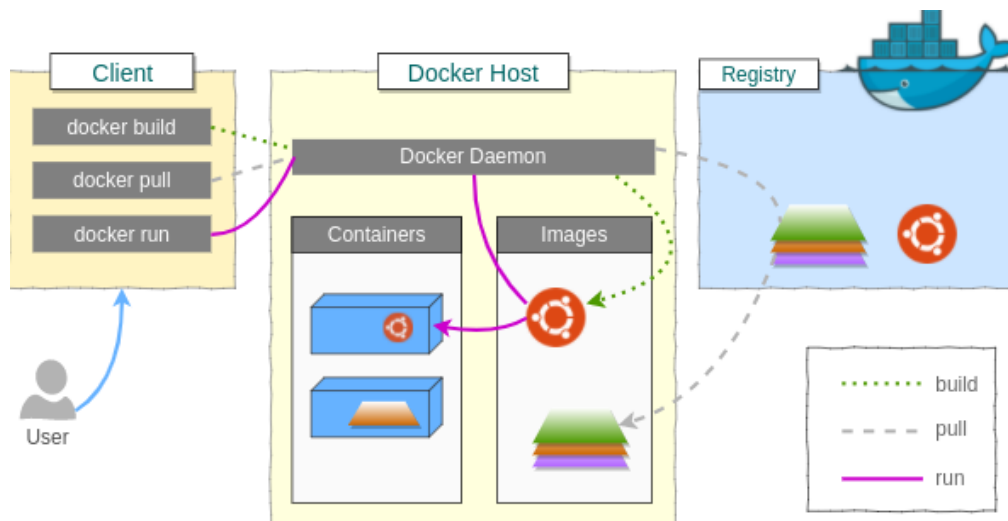
Pentru a putea folosi containerul de docker cu baza de date Oracle, va trebui să avem un cont pe [dockerhub](https://hub.docker.com/) pentru a putea descărca imaginea docker. Ne logăm și pe aplicația *Docker Desktop* în colțul din dreapta sus. Pentru cine folosește varianta din linia de comandă, trebuie să vă logați folosind comanda `docker login`

După care intrăm pe link-ul [imaginii de docker](https://docs.docker.com/get-docker/), apăsăm proceed to checkout și get content pentru a avea acces la imagine. În pagina la care am fost redirecționați avem mai multe instrucțiuni pentru a configura serverul. Mai multe detalii concrete sunt în secțiunea următoare.

### 4 Pornirea și oprirea serverului

În diagrama următoare avem la un nivel înalt arhitectura docker. Observăm că docker funcționează ca un proxy între noi și serverul nostru.

Putem face o analogie cu programul **npm**: Docker administrează și controlează imagini (care sunt un fel de "snapshot"-uri în timp a unor mașini virtuale), la fel cum programul **npm** administrează pachete pentru runtime-ul nodeJS. Doar că în plus față de **npm**, imaginile pot fi rulate în interiorul unor containere cu care noi putem comunica și trimite comenzi.



De asemenea, trebuie să înțelegem diferența dintre cele 3 comenzi de docker: *run*, *start*, *stop*

- **Run:** Pe baza unei imagini va crea un container cu anumite setări și va fi și pornit. Prin setări ne referim de exemplu la: să facem disponibile diferite porturi, sau a avea fișierele bazei de date defapt pe host, și nu în interiorul containerului pentru cazul în care vom vrea să copiem ușor datele.
- **Start:** Pornim un container care a fost deja creat și este oprit.
- **Stop:** Oprim un container care rulează în acel moment.

De exemplu, comanda de mai jos ne va crea un container care va rula o bază de date, dar vom putea interacționa cu ea numai trimițând comenzi prin intermediul docker client, neavând niciun port partajat către mașina locală. În același timp, neavând un *volume* atașat acestui container pentru path-ul */ORCL* (din mașina virtuală) un folder din mașina host, nu vom putea accesa direct din host fișierele ce conțin datele bazei de date.

```
$ > docker run -d -it --name Oracle-DB store/oracle/database-enterprise:12.2.0.1
```

De menționat, numele **Oracle-DB** poate fi schimbat cu ce nume dorim noi pentru container, iar **12.2.0.1** reprezintă ultima versiune a imaginii. Pe măsura ce Oracle actualizează software-ul, putem modifica și noi versiunea imaginii, deși nu este necesar.

Vom folosi în schimb această linie de comandă pentru a ne crea imaginea de docker:

```
$ > docker run -d -it --name Oracle-DB -p 1521:1521  
store/oracle/database-enterprise:12.2.0.1
```

Opțiunea *-p* va face forward la portul 1521 pentru a ne conecta de exemplu cu SQL Developer la baza de date. Mai multe detalii despre configurarea containerului găsiți pe pagina de dockerhub a imaginii.

În continuare, pentru a opri / porni serverul:

- **docker start Oracle-DB** # Porneste container-ul
- **docker stop Oracle-DB** # Opreste container-ul

## 5 Metode de conectare și utilizare

### 5.1 Direct in terminal

Dacă vrem să accesăm linia de comandă exact ca și cum am utiliza mașina virtuală, va trebui să folosim comanda

```
$ > docker exec -it Oracle-DB bash
```

### 5.2 Acces direct Sql\*Plus

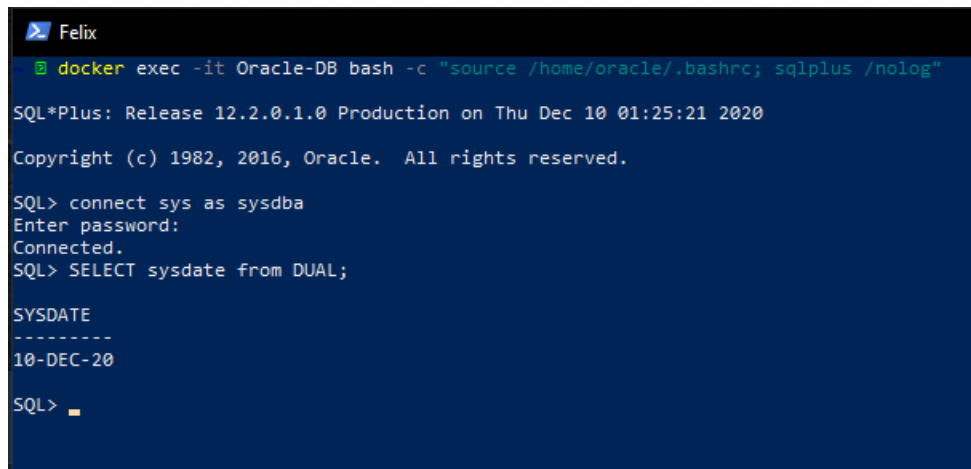
Pentru a accesa direct interfața Sql\*Plus din interiorul container-ului avem comanda

```
$ > docker exec -it Oracle-DB bash -c "source /home/oracle/.bashrc; sqlplus /nolog"
```

Care este echivalentă cu a rula cele doua comenzi:

```
$ > docker exec -it Oracle-DB bash
$ > source /home/oracle/.bashrc; sqlplus /nolog
```

Pentru variantele de conectare prin linia de comanda, odată ajunși în consola de SQLPlus, rulăm următoarele comenzi pentru a ne autentifica:



```
Felix
docker exec -it Oracle-DB bash -c "source /home/oracle/.bashrc; sqlplus /nolog"

SQL*Plus: Release 12.2.0.1.0 Production on Thu Dec 10 01:25:21 2020

Copyright (c) 1982, 2016, Oracle. All rights reserved.

SQL> connect sys as sysdba
Enter password:
Connected.
SQL> SELECT sysdate from DUAL;

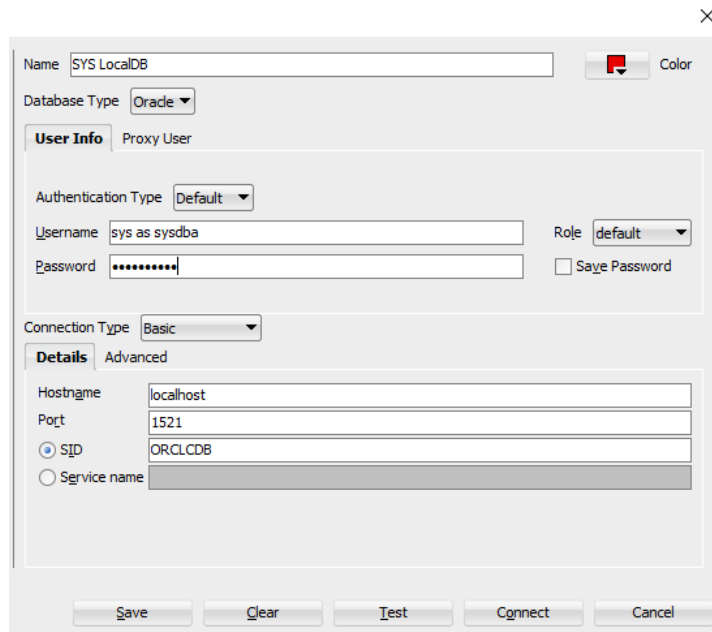
SYSDATE
-----
10-DEC-20

SQL> ■
```

### 5.3 SQL Developer

SQL Developer poate fi download-at de pe site-ul oficial la acest [link](#).

După instalare, creăm o nouă conexiune având următoarele proprietăți:



- Username: **sys as sysdba** (Pentru user-ul system)
- Password: **Oradoc\_db1** Parola default
- Hostname: **localhost**
- Port: **1521**
- SID: **ORCLCDB**

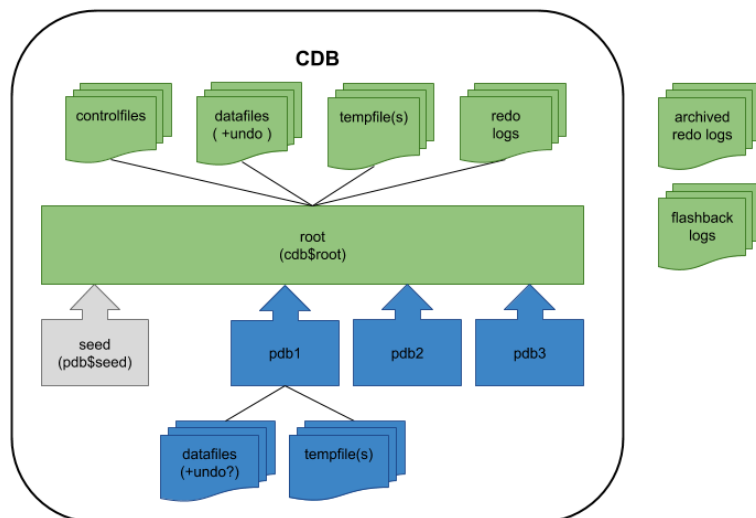
**Nota:** Observați că această conexiune este pentru userul *SYS*, ceea ce nu este recomandat. Însă trebuie să intrăm măcar o dată cu acest user pentru a crea alte conturi pentru accesul la baza de date.

## 6 Crearea unui user

Odată conectați cu user-ul *sys* (prin oricare din metodele expuse mai devreme), putem crea un alt user cu toate privilegiile, pentru a putea gestiona în mod normal o bază de date. Putem folosi următoarele comenzi în PL/SQL:

```
CREATE USER c##some_username IDENTIFIED BY MyPassword;  
GRANT ALL PRIVILEGES TO some_username;
```

**Nota:** Numele de utilizatori trebuie să înceapă cu prefixul **c##** datorită unui feature pe nume *pluggable databases*. Pe scurt, o bază de date împreună cu toate mecanismele ei se numește un *CDB*: Container database, care poate conține mai multe *PDB*: pluggable database, ele în schimb sunt mai mici fiind axate mai mult doar pe a stoca informația și de a putea fi mutate / copiate. Acest prefix stabilește că utilizatorul creat poate avea acces la toate bazele de date: cea *CDB*, dar și toate *PDB* care au fost atașate și vor fi atașate. O schemă se poate vedea mai jos:



## 7 Concluzie

Crearea unei baze de date nu este un proces ușor, însă o înțelegere fermă asupra mediului de lucru ne poate ajuta să înțelegem mai bine factorii ce pot influența performanța și modul de utilizare. Acest referat are scopul de a facilita acest proces de învățare, aducând într-un loc informații privind diferite aspecte.