

Despliegue de servidor Oracle y creación de Database link

1. Instalación de base de oracle 18c

Vamos a instalar la base de datos de oracle 18c en un Ubuntu Server 20.04 LTS.

Principalmente vamos a descargar Oracle Database 18c Express Edition:

<https://www.oracle.com/technetwork/database/database-technologies/express-edition/downloads>

Se nos descargara un paquete .rpm que tendremos que pasar a .deb utilizando el comando `alien` 'sudo alien --scripts -d oracle-database-xe-18c-1.0-1.x86_64.rpm' tardara un rato. Yo lo he ejecutado en el anfitrión ya que si lo ejecutamos desde la MV tiraría de muchos recursos y iría mas lento.

Una vez pasado a .deb utilizo el comando scp para pasarlo a la maquina virtual 'sudo scp oracle-database-xe-18c_1.0-2amd64.deb abraham@192.168.122.193:~'

Ahora instalamos en la MV los paquetes necesarios para la instalación 'sudo apt install libaio1 unixodbc'

Creamos el script chkconfig requerido utilizamos 'sudo nano /sbin/chkconfig' y ponemos esto en el dichero:

```
1 #!/bin/bash
2 # Oracle 11gR2 XE installer chkconfig hack for Ubuntu
3 file=/etc/init.d/oracle-xe
4 if [[ ! `tail -n1 $file | grep INIT` ]]; then
5 echo >> $file
6 echo '### BEGIN INIT INFO' >> $file
7 echo '# Provides: OracleXE' >> $file
8 echo '# Required-Start: $remote_fs $syslog' >> $file
9 echo '# Required-Stop: $remote_fs $syslog' >> $file
10 echo '# Default-Start: 2 3 4 5' >> $file
11 echo '# Default-Stop: 0 1 6' >> $file
12 echo '# Short-Description: Oracle 11g Express Edition' >> $file
13 echo '### END INIT INFO' >> $file
14 fi
15 update-rc.d oracle-xe defaults 80 01
```

Cambiamos los permisos del fichero chkconfig usando el comando **'sudo chmod 755 /sbin/chkconfig'**

Establecemos los parámetros del kernel **'sudo nano /etc/sysctl.d/60-oracle.conf'** y ponemos esto en el fichero:

```
1 # Oracle 18c XE kernel parameters
2 fs.file-max=6815744
3 net.ipv4.ip_local_port_range=9000 65000
4 kernel.sem=250 32000 100 128
5 kernel.shmmax=536870912 |
```

Hacemos **'sudo service procps start'** para cargar los parámetros del núcleo y podemos verificarlos con **'sudo sysctl -q fs.file-max'**

Configuramos el punto de montaje para Oracle. Creamos el fichero **'sudo nano /etc/rc2.d/S01shm_load'** y introducimos esto en el fichero:

```
1 #!/bin/sh
2 case "$1" in
3 start) mkdir /var/lock/subsys 2>/dev/null
4         touch /var/lock/subsys/listener
5         rm /dev/shm 2>/dev/null
6         mkdir /dev/shm 2>/dev/null
7         mount -t tmpfs shmfs -o size=2048m /dev/shm ;;
8 *) echo error
9     exit 1 ;;
10 esac
```

Cambiamos los permisos del fichero **'sudo chmod 755 /etc/rc2.d/S01shm_load'** y ahora reiniciamos la maquina virtual.

Vamos a la instalación de oracle utilizando **'sudo dpkg --install oracle-database.....deb'**

Editamos el fichero y añadimos el LISTENER_PORT

'sudo nano /etc/sysconfig/oracle-xe-18c.conf'

```
#This is a configuration file to setup the Oracle Database.
#It is used when running '/etc/init.d/oracle-xe-18c configure'.

# LISTENER PORT used Database listener, Leave empty for automatic port assignment
LISTENER_PORT=1521

# EM_EXPRESS_PORT Oracle EM Express URL port
EM_EXPRESS_PORT=5500

# Character set of the database
CHARSET=AL32UTF8

# Database file directory
# If not specified, database files are stored under Oracle base/oradata
DBFILE_DEST=

# SKIP Validations, memory, space
SKIP_VALIDATIONS=false
```

Arrancamos el script como root **'sudo /etc/init.d/oracle-xe-18c configure'** la primera vez nos dara un error de listener pero lo volvemos a ejecutar y nos funcionara, establecemos la contraseña de SYS.

Configuramos las variables de entorno en nuestro fichero .bashrc:

nano ~/.bashrc

```
#Las siguientes variables las hemos obtenido examinando el fichero /etc/init.d/oracle-xe-18c configure
# DB defaults
export ORACLE_HOME=/opt/oracle/product/18c/dbhomeXE
export ORACLE_SID=XE
export TEMPLATE_NAME=XE_Database.dbc
export PDB_NAME=XEPDB1
export LISTENER_NAME=LISTENER
export NUMBER_OF_PDBS=1
export CREATE_AS_CDB=true
export LD_LIBRARY_PATH=/opt/oracle/product/18c/dbhomeXE/lib/

# General exports and vars
export PATH=$ORACLE_HOME/bin:$PATH
LSNR=$ORACLE_HOME/bin/lsnrctl
SQLPLUS=$ORACLE_HOME/bin/sqlplus
NETCA=$ORACLE_HOME/bin/netca
DBCA=$ORACLE_HOME/bin/dbca
ORACLE_OWNER=oracle
RETVAL=0
CONFIG_NAME="oracle-xe-18c.conf"
CONFIGURATION="/etc/sysconfig/$CONFIG_NAME"
ORACLE_HOME_NAME="OraHomeXE"
MINIMUM_MEMORY=1048576
MAXIMUM_MEMORY=2097152
MINIMUM_MEMORY_STR="1GB"

MINIMUM_SPACE=4718592
MINIMUM_SPACE_STR="4.5GB"
```

Ejecutamos los cambios ejecutando nuestro perfil: `. ~/.profile`

Agregamos a nuestro usuario al grupo dba '`sudo usermod -a -G dba abraham`' asignamos una contraseña al usuario oracle con '`sudo passwd oracle`' nos conectamos como oracle con '`su oracle`' nos saldra una consola `/sh` tendremos que cambiarlo a `/bash` en el fichero `/etc/passwd`

```
oracle:x:54321:54321::/home/oracle:/bin/bash
```

Una vez lo cambiemos iniciamos otra vez con oracle y cargamos el `.bashrc`

```
abraham@oracle:~$ su oracle
Password:
oracle@oracle:/home/abraham$ source .bashrc
oracle@oracle:/home/abraham$
```

Esto lo hacemos para cargar las variables de entorno para oracle.

Ponemos `sudo su` para entrar como root

Ahora copiamos el archivo init que estara en una carpeta de oracle con unos numeros que se le asignan.

```
'cp /opt/oracle/admin/ora18c/pfile/init.ora.XXXXXXX
/opt/oracle/product/18c/dbhomeXE/dbs/initXE.ora'
```

```
Cambiamos el propietario 'chown oracle:oinstall
/opt/oracle/product/18c/dbhomeXE/dbs/initXE.ora'
```

Arrancamos la BD `sqlplus / as sysdba` y ponemos `startup` y si nos sale algo como esto es que todo va bien:

Ya estaría listo!!!

```
abraham@oracle:~$ sqlplus / as sysdba

SQL*Plus: Release 18.0.0.0.0 - Production on Tue Mar 2 16:57:25 2021
Version 18.4.0.0.0

Copyright (c) 1982, 2018, Oracle. All rights reserved.

Connected to an idle instance.

SQL> startup
ORACLE instance started.

Total System Global Area 1610612016 bytes
Fixed Size 8658224 bytes
Variable Size 520093696 bytes
Database Buffers 1073741824 bytes
Redo Buffers 8118272 bytes
Base de datos montada.
Base de datos abierta.
SQL> _
```

2. Instalación de PostgreSQL

Vamos a instalar PostgreSQL para luego poder hacer la practica de database link. Esta instalación es muy sencilla.

Hacemos **'sudo apt update'** y luego **'sudo apt install postgresql postgresql-contrib'**

Una vez hayamos hecho esto podemos cambiar a la cuenta de postgres haciendo **'sudo -i -u postgres'** si ponemos **psql** entramos para administrar la base de datos.

Con **'sudo -u postgres psql'** accedemos a la administración de la base de datos sin cambiar de cuenta.

Vamos a crear un nuevo usuario. Vamos a hacerlo con la cuenta de postgres como hemos dicho anteriormente. Cuando estemos dentro ponemos **'createuser -interactive'** nos pedirá el nombre del usuario y que si es un superusuario.

En mi caso el nombre seria abraham y respondemos yes en superusuario. Esto nos quita un paso ya que si hubiéramos puesto otro nombre al usuario deberíamos haber creado un usuario en ubuntu con dicho nombre pero como ya tenemos al usuario abraham no hace falta crear nada. Ya podremos entrar con **psql** directamente:

```
abraham@oracle:~$ psql
psql (12.5 (Ubuntu 12.5-0ubuntu0.20.04.1))
Type "help" for help.

abraham=# _
```

Y ya estaría listo!!

3. Creación de esquema y tablas en Oracle y PostgreSQL

Oracle

Para la creación del esquema y de la tabla utilizare un .sql que tengo del año pasado.

```

1 set echo on
2 spool eslog.err
3 -----
4 /*drop user LINK cascade;*/
5 create user LINK identified by zelda
6     default tablespace USERS
7     quota unlimited on USERS;
8 grant create session, create table, create procedure to LINK;
9 grant create public synonym, create sequence, create view to LINK;
10 connect LINK/zelda@192.168.1.212
11 -----
12 create table MODULOS(
13 idMod          number,
14 NomMod         varchar2(100) not null,
15 CodMod         varchar2(10),
16 Abrev          varchar2(5),
17               constraint pk_modulos primary key (idMod)
18 );
19

```

Y para poblar la tabla:

```

connect LINK/zelda@192.168.1.212
spool er.txt
alter session set nls_date_format = 'dd/mm/yyyy';

-- Población de la tabla MODULOS --

insert into MODULOS (idMod, NomMod, CodMod, Abrev) values (1, 'Administracionde Sistemas
Informaticos en Red', 'IFC01S', 'ASIR');
insert into MODULOS (idMod, NomMod, CodMod, Abrev) values (2, 'Desarrollo de Aplicaciones
Multiplataformas', 'IFC02S', 'DAM');
insert into MODULOS (idMod, NomMod, CodMod, Abrev) values (3, 'Desarrollo de Aplicaciones Web',
'IFC03S', 'DAW');
commit;

```

Para usarlos tenemos que tener los dos ficheros .sql en la carpeta personal y ponemos `sqlplus / as sysdba` luego solo tenemos que poner `@nombre_fichero`

```
SQL> select * from modulos;
```

IDMOD	NOMMOD	CODMOD	ABREV
1	Administracionde Sistemas Informaticos en Red	IFC01S	ASIR
2	Desarrollo de Aplicaciones Multiplataformas	IFC02S	DAM
3	Desarrollo de Aplicaciones Web	IFC03S	DAW

```
SQL> _
```

PostgreSQL

Para crear un esquema debemos poner **'sudo -u abraham createdb abraham'** cuando entremos en psql con abraham ponemos \conninfo para ver en que base nos encontramos.

```
abraham=# \conninfo
you are connected to database "abraham" as user "abraham" via socket in "/var/run/postgresql" at port "5432".
abraham=# _
```

Una vez dentro para crear una tabla esta vez no vamos a usar un fichero y lo haremos por CLI:

```
abraham=# create table MODULOS(
idMod serial PRIMARY KEY,
NomMod varchar (100) NOT NULL,
CodMod varchar (10),
Abrev varchar (5)
);_
```

Tendremos la misma tabla que en oracle. Y la poblaremos así:

```
abraham=# insert into MODULOS (idMod, NomMod, CodMod, Abrev) values (1, 'Administracionde Sistemas Informaticos en Red', 'IFC01S', 'ASIR');
```

```
abraham=# insert into MODULOS (idMod, NomMod, CodMod, Abrev) values (2, 'Desarrollo de Aplicaciones Multiplataformas', 'IFC02S', 'DAM');
```

```
abraham=# insert into MODULOS (idMod, NomMod, CodMod, Abrev) values (3, 'Desarrollo de Aplicaciones Web', 'IFC03S', 'DAW');
```

Podemos ver haciendo un select:

```
abraham=# select * from MODULOS;
 idmod |          nommod          | codmod | abrev
-----+-----+-----+-----
      1 | Administracionde Sistemas Informaticos en Red | IFC01S | ASIR
      2 | Desarrollo de Aplicaciones Multiplataformas  | IFC02S | DAM
      3 | Desarrollo de Aplicaciones Web                | IFC03S | DAW
(3 rows)

abraham=# _
```


4. Creación de un database link

Vamos a crear un database link de Oracle a PostgreSQL. Para poder enlazar los dos servidores siendo de diferentes gestores de base de datos lo que tendremos que hacer antes de nada es instalar los dos siguientes paquetes en la maquina virtual '**sudo apt install odbc-postgresql unixodbc**'

Cuando ya lo hayamos instalado procederemos a realizar la configuración en el servidor de oracle.

Empezaremos modificando el fichero /etc/odbcinst.ini dejando los parámetros de configuración así:

```
GNU nano 4.8 /etc/odbcinst.ini
[PostgreSQL ANSI]
Description=PostgreSQL ODBC driver (ANSI version)
Driver=psqlodbc.a.so
Setup=libodbcpsqlS.so
Debug=0
CommLog=1
UsageCount=1

[PostgreSQL Unicode]
Description=PostgreSQL ODBC driver (Unicode version)
Driver=psqlodbcw.so
Setup=libodbcpsqlS.so
Debug=0
CommLog=1
UsageCount=1
```

Una vez hallamos configurado el fichero procederemos a configurar el segundo fichero que se encuentra en /etc/odbc.ini


```
GNU nano 4.8
[PSQLA]
Debug = 0
CommLog = 0
ReadOnly = 1
Driver = PostgreSQL ANSI
Servername = localhost
Username = abraham
Password = 123456
Port = 5432
Database = abraham
Trace = 0
TraceFile = /tmp/sql.log

[PSQLU]
Debug = 0
CommLog = 0
ReadOnly = 0
Driver = PostgreSQL Unicode
Servername = localhost
Username = abraham
Password = 123456
Port = 5432
Database = abraham
Trace = 0
TraceFile = /tmp/sql.log

[Default]
Driver = /usr/lib/x86_64-linux-gnu/odbc/libodbcS.so
```

Con los parámetros de este fichero estamos indicando la dirección ip de nuestro servidor de postgresql, el nombre del usuario que gestiona la base de datos con su contraseña además de la base de datos a la que vamos a consultar y el puerto por el que va a estar escuchando además del fichero de log del servicio.

Cuando ya hallamos configurado los dos ficheros anteriores podremos comprobar que la configuración se ha realizado correctamente con este comando 'odbcinst -q -d'

```
abraham@oracle:~$ odbcinst -q -d
[PostgreSQL ANSI]
[PostgreSQL Unicode]
abraham@oracle:~$ _
```

Si introducimos este segundo comando `'odbcinst -q -s'` comprobamos si el fichero `odbc.ini` ha sido configurado correctamente.

```
abraham@oracle:~$ odbcinst -q -s
[PSQLA]
[PSQLU]
[Default]
abraham@oracle:~$
```

Vamos a comprobar también la conexión usando `'isql -v PSQLU'` y `'isql -v PSQLA'`

```
abraham@oracle:~$ isql -v PSQLU
+-----+
| Connected!
|
| sql-statement
| help [tablename]
| quit
|
+-----+
SQL> quit
abraham@oracle:~$ isql -v PSQLA
+-----+
| Connected!
|
| sql-statement
| help [tablename]
| quit
|
+-----+
SQL> quit
abraham@oracle:~$
```

Una vez comprobado que todo funciona correctamente procederemos a configurar el `initPSQLU.ora` dentro del directorio `'/opt/oracle/product/18c/dbhomeXE/hs/admin'`

Tendremos que añadir esto al fichero:

```
GNU nano 4.8 initPSQLU
HS_FDS_CONNECT_INFO = PSQLU
HS_FDS_TRACE_LEVEL = 0
HS_FDS_SHAREABLE_NAME = /usr/lib/x86_64-linux-gnu/odbc/psqlodbcw.so
HS_LANGUAGE = AMERICAN_AMERICA.WE8ISO8859P1
set ODBCINI=/etc/odbc.ini
```

Ahora tendremos que editar el fichero del listener.ora añadiendo la nueva conexión

```
GNU nano 4.8 listener.ora
# listener.ora Network Configuration File: /opt/oracle/product/18c/dbhomeXE/network/admin/listener.ora
# Generated by Oracle configuration tools.

LISTENER =
  (DESCRIPTION_LIST =
    (DESCRIPTION =
      (ADDRESS = (PROTOCOL = TCP)(HOST = 0.0.0.0)(PORT = 1521))
      (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC1521))
    )
  )

SID_LIST_LISTENER =
  (SID_LIST =
    (SID_DESC =
      (GLOBAL_DBNAME = ORA18C)
      (ORACLE_HOME = /opt/oracle/product/18c/dbhomeXE)
      (SID_NAME = XE)
    )
  )

SID_LIST_LISTENER=
  (SID_LIST=
    (SID_DESC=
      (SID_NAME=PSQLU)
      (ORACLE_HOME=/opt/oracle/product/18c/dbhomeXE)
      (PROGRAM=dg4odbc)
    )
  )
```

También tendremos que editar el tnsnames.ora

```
GNU nano 4.8 tnsnames.ora
# tnsnames.ora Network Configuration File: /opt/oracle/product/18c/dbhomeXE/network/admin/tnsnames.ora
# Generated by Oracle configuration tools.

ORA18C =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = 192.168.122.42)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = ora18c)
    )
  )

LISTENER =
  (ADDRESS = (PROTOCOL = TCP)(HOST = 192.168.122.42)(PORT = 1521))

PSQLU =
  (DESCRIPTION=
    (ADDRESS=(PROTOCOL=tcp)(HOST=192.168.122.42)(PORT=1521))
    (CONNECT_DATA=(SID=PSQLU))
    (HS=OK)
  )
```

Cuando ya lo tengamos todo ahora tendremos que parar el listener y volver a encenderlo.

```
oracle@oracle:/home/abraham$ lsnrctl stop

LSNRCTL for Linux: Version 18.0.0.0.0 - Production on 02-MAR-2021 21:48:07

Copyright (c) 1991, 2018, Oracle. All rights reserved.

Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=0.0.0.0)(PORT=1521)))
The command completed successfully
```

```
oracle@oracle:/home/abraham$ lsnrctl start

LSNRCTL for Linux: Version 18.0.0.0.0 - Production on 02-MAR-2021 21:48:44

Copyright (c) 1991, 2018, Oracle. All rights reserved.

Starting /opt/oracle/product/18c/dbhomeXE/bin/tnslsnr: please wait...

TNSLSNR for Linux: Version 18.0.0.0.0 - Production
System parameter file is /opt/oracle/product/18c/dbhomeXE/network/admin/listener.ora
Log messages written to /opt/oracle/diag/tnslsnr/oracle/listener/alert/log.xml
Listening on: (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=0.0.0.0)(PORT=1521)))
Listening on: (DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=EXTPROC1521)))

Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=0.0.0.0)(PORT=1521)))
STATUS of the LISTENER
-----
Alias                LISTENER
Version              TNSLSNR for Linux: Version 18.0.0.0.0 - Production
Start Date           02-MAR-2021 21:48:44
Uptime               0 days 0 hr. 0 min. 0 sec
Trace Level          off
Security             ON: Local OS Authentication
SNMP                 OFF
Listener Parameter File /opt/oracle/product/18c/dbhomeXE/network/admin/listener.ora
Listener Log File    /opt/oracle/diag/tnslsnr/oracle/listener/alert/log.xml
Listening Endpoints Summary...
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=0.0.0.0)(PORT=1521)))
  (DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=EXTPROC1521)))
Services Summary...
Service "PSQLU" has 1 instance(s).
  Instance "PSQLU", status UNKNOWN, has 1 handler(s) for this service...
The command completed successfully
```

Ahora una vez hayamos hecho esto tendremos que configurar algunos ficheros de postgresql

Tendremos que editar 'nano /etc/postgresql/12/main/postgresql.conf' y donde pone
listen_addresses = '*'

```
#-----  
# CONNECTIONS AND AUTHENTICATION  
#-----  
  
# - Connection Settings -  
  
listen_addresses = '*'  
#what IP address(es) to listen on;  
# comma-separated list of addresses;  
# defaults to 'localhost'; use '*' for all  
# (change requires restart)
```

y en el fichero 'nano /etc/postgresql/12/main/pg_hba.conf' tendremos que buscar la linea host y poner host all all 0.0.0.0/0 md5 y también host abraham abraham direccion ip md5

```
# Allow replication connections from localhost, by a user with the  
# replication privilege.  
local replication all  
host all all 0.0.0.0/0 md5 peer  
host replication all ::1/128 md5  
host abraham abraham 192.168.122.42/32 md5
```

Reiniciamos el servicio de postgresql haciendo 'systemctl restart postgresql'.

Ya modificado los dos servidores correctamente, vamos a realizar la creación del Link de la base de datos en oracle para que pueda consultar en postgresql.

Pondremos

create public database link PSQL1

connect to "abraham" identified by "123456"

using 'PSQLU';