

Práctica *scripts* bash

Ismael Macareno Chouikh

2025-02-13

Índice

1. <i>Scripts</i> de inicio y parada de Oracle	2
2. Arrancar automáticamente Oracle cuando se inicie el servidor	2
3. Crea usuarios de base de datos	3
4. Almacena información periódicamente en la base de datos	5
5. Envía un correo periódicamente	7
6. <i>units</i> de systemd	8
7. Fichero <code>/etc/crontab</code>	8

1. *Scripts* de inicio y parada de Oracle

```
1  #!/bin/bash
2
3  # Variables de entorno
4  export ORACLE_SID=asir
5  export ORACLE_HOME=/opt/oracle-install
6  export PATH=$ORACLE_HOME/bin:$PATH
7
8  # Iniciar el listener
9  lsnrctl start
10
11 # Arrancar oracle
12 sqlplus / as sysdba <<EOF
13 startup;
14 exit;
15 EOF
```

```
1  #!/bin/bash
2  # variables de entorno
3  export ORACLE_SID=asir
4  export ORACLE_HOME=/opt/oracle-install
5  export PATH=$ORACLE_HOME/bin:$PATH
6
7  # Parar oracle
8  sqlplus / as sysdba <<EOF
9  shutdown inmediato;
10 exit;
11 EOF
12
13 # Parar el listener
14 lsnrctl stop
```

2. Arrancar automáticamente Oracle cuando se inicie el servidor

```
1  #!/bin/bash
2
3  # por si acaso, deshabilitamos SELinux para que no moleste
4  # modificar /etc/selinux/config para hacerlo permanente
5  setenforce 0
6
7  # ORACLE_HOME del sistema
8  valor_oracle_home=$(env | grep ORACLE_HOME | cut -d "=" -f 2)
9
10 # miro en oratab
11 valor_oracle_home_en_oratab=$(cat /etc/oratab | grep $valor_oracle_home | cut -d ":" -f 3)
12
13
```

```

14 # Habra que crear el fichero de logs si no existe
15 directorio_de_logs="/home/alumno/logs"
16 fichero_de_logs="/home/alumno/logs/oracle.log"
17 mkdir -p "$directorio_de_logs"
18
19
20 # tenia la fecha formateada con el echo normal, pero me apetece meter funciones
21 # date "+%Y-%m-%d-%H:%M:%S" formato de fecha para "$fichero_de_logs"
22 fechaformateada (){
23     date "+%Y-%m-%d-%H:%M:%S"
24 }
25
26
27 if [ -n "$valor_oracle_home_en_oratab" ] && [ $valor_oracle_home_en_oratab = "Y" ]; then
28     echo "$(fechaformteada) Solicitud de arrancar Oracle" | tee -a "$fichero_de_logs"
29     echo "$(fechaformteada) Oracle arrancado porque /etc/oratab indica Y" | tee -a "$fichero_de_logs"
30     sqlplus / as sysdba <<EOF
31 startup;
32 EOF
33     echo "$(fechaformteada) Oracle arrancado" | tee -a "$fichero_de_logs"
34 else
35     echo "$(fechaformteada) Solicitud de arrancar Oracle" | tee -a "$fichero_de_logs"
36     echo "$(fechaformteada) Oracle arrancado porque /etc/oratab indica N" | tee -a "$fichero_de_logs"
37     echo "$(fechaformteada) Solicitud de parar Oracle" | tee -a "$fichero_de_logs"
38     sqlplus / as sysdba <<EOF
39 shutdown immediate;
40 EOF
41     echo "$(fechaformteada) Oracle parado" | tee -a "$fichero_de_logs"
42 fi

```

3. Crea usuarios de base de datos

```

1 #!/bin/bash
2
3 # select username, account_status from dba_users where lock_date is not null / para saber si un usuari
4 # ALTER USER account ACCOUNT UNLOCK; / para desbloquear la cuenta
5
6 # Script para crear nuevos usuarios de oracle
7
8 # Variables de entorno
9 # Variables de entorno
10 export ORACLE_SID=zilae
11 export ORACLE_HOME=/opt/oracle-install
12 export PATH=$ORACLE_HOME/bin:$PATH
13
14
15 nombre_usuario=$1
16 contrasena_usuario=$2
17 pdb="pdzilae"

```

```

18
19 # comprobar si mete mas parametros de los que deberia
20 parametros_introducidos=$#
21
22 if [ "$parametros_introducidos" -ne 2 ]; then
23     echo "Crea un usuario nuevo de oracle, con permisos connect y resource."
24     echo "Si el usuario ya existe, lo desbloquea y le cambia la contraseña."
25     echo ""
26     echo "Uso: nuevo-usuario-oracle.sh <usuario> <contraseña>"
27     sleep 10
28     exit
29 fi
30
31
32 comprobar_si_existe_usuario_oracle (){
33     local nombre_usuario="$1"
34
35     resultado_query=$(sqlplus -s / as sysdba<<EOF
36 SET PAGESIZE 0 FEEDBACK OFF VERIFY OFF HEADING OFF ECHO OFF
37 SELECT COUNT(*) FROM dba_users WHERE USERNAME = UPPER('${nombre_usuario}');
38 EXIT;
39 EOF
40         )
41
42     # recortar espacios en blanco
43     echo "$resultado_query" | xargs
44 }
45
46 comprobar_si_usuario_bloqueado (){
47     resultado_query_block=$(sqlplus -s / as sysdba<<EOF
48 SET PAGESIZE 0 FEEDBACK OFF VERIFY OFF HEADING OFF ECHO OFF
49 SELECT COUNT(*) FROM dba_users WHERE lock_date IS NOT NULL AND USERNAME = UPPER('${nombre_usuario}')
50 EXIT;
51 EOF
52         )
53
54     echo "$resultado_query_block" | xargs
55 }
56
57 verificar_pdb(){
58     estado_pdb=$(sqlplus -s / as sysdba<<EOF
59 SET PAGESIZE 0 FEEDBACK OFF VERIFY OFF HEADING OFF ECHO OFF
60 SELECT OPEN_MODE FROM v\$pdbs WHERE NAME = UPPER('${pdb}');
61 EXIT;
62 EOF
63         )
64     echo "$estado_pdb" | xargs
65 }
66
67 crear_usuario_dentro_de_oracle (){

```

```

68     local estado_pdb=$(verificar_pdb)
69     if [ "$estado_pdb" != "READ WRITE" ]; then
70         sqlplus -s / as sysdba<<EOF
71 ALTER PLUGGABLE DATABASE ${pdb} OPEN;
72 EOF
73     fi
74     sqlplus -s / as sysdba<<EOF
75 ALTER SESSION SET CONTAINER=${pdb};
76 CREATE USER ${nombre_usuario} IDENTIFIED BY ${contrasena_usuario};
77 EXIT;
78 EOF
79 }
80
81 desbloquea_usu(){
82 sqlplus -s / as sysdba<<EOF
83 ALTER USER '${nombre_usuario}' ACCOUNT UNLOCK;
84 EXIT;
85 EOF
86 }
87
88 resultado_funcion_comprobar_usuario=$(comprobar_si_existe_usuario_oracle)
89
90
91 if [ "$resultado_funcion_comprobar_usuario" -gt 0 ]; then
92     echo "El usuario $nombre_usuario ya existe"
93
94     resultado_funcion_usuario_block=$(comprobar_si_usuario_bloqueado)
95     if [ "$resultado_funcion_usuario_block" -eq 1 ]; then
96         echo "El usuario $nombre_usuario esta bloqueado"
97         echo "Desbloqueando....."
98         desbloquea_usu
99         echo "El usuario $nombre_usuario ha sido desbloqueado"
100     fi
101 else
102     echo "El usuario $nombre_usuario no existe"
103     echo "Creando....."
104     crear_usuario_dentro_de_oracle
105     echo "El usuario $nombre_usuario ha sido creado"
106 fi

```

4. Almacena información periódicamente en la base de datos

```

1  #!/bin/bash
2  # Almacena información periódicamente en la base de datos
3
4  # Variables de entorno para que chute sqlplus
5  export ORACLE_SID=zilae
6  export ORACLE_HOME=/opt/oracle-install
7  export PATH=$ORACLE_HOME/bin:$PATH

```

```
8
9  # Creacion de tablas
10 # Quiero comprobar si existe o no la tabla
11 crear_tablas(){
12     # Si se quita el SET HEADING OFF da fallo diciendo que no existe la tabla DF
13     tabla_existe=$(sqlplus -s / as sysdba<<EOF
14 SET HEADING OFF
15 SELECT COUNT(*) FROM all_tables WHERE table_name = 'DF';
16 EXIT;
17 EOF
18     )
19     if [ "$tabla_existe" -eq 0 ]; then
20 sqlplus -s / as sysdba<<EOF
21 create table DF(
22 hora varchar(40),
23 sistema varchar(40),
24 tamaño varchar(40),
25 usado varchar(40),
26 montado varchar(40)
27 );
28 EXIT;
29 EOF
30
31     else
32         echo "Ya existe la tabla DF"
33     fi
34 }
35
36 hora_actual=$(date '+%T')
37
38 crear_tablas
39
40 df -k | awk '{print $1, $2, $3, $6}' | tail -n +2 | while read -r sistema tamaño usado montado; do
41     insertar="INSERT INTO DF(hora, sistema, tamaño, usado, montado) VALUES(
42 '$hora_actual', '$sistema', '$tamaño', '$usado', '$montado');"
43
44     sqlplus -s / as sysdba<<EOF
45 $insertar
46 EOF
47 done
48
49 # df -k | awk '{print $1}' | tail -n +2
50 # df -k | awk '{print $2}' | tail -n +2
51 # df -k | awk '{print $3}' | tail -n +2
52 # df -k | awk '{print $6}' | tail -n +2
```

5. Envía un correo periódicamente

```
1  #!/bin/bash
2  # Envía un correo periódicamente
3  # Enviar a alvaro@alvarogonzalez.no-ip.biz
4
5  # Variables de entorno
6  export ORACLE_SID=zilae
7  export ORACLE_HOME=/opt/oracle-install
8  export PATH=$ORACLE_HOME/bin:$PATH
9
10 sacar_informacion(){
11     info=$(sqlplus -s / as sysdba<<EOF
12 set colsep ,      -- separate columns with a comma
13 set pagesize 0    -- No header rows
14 set trimspace on  -- remove trailing blanks
15 set headsep off   -- this may or may not be useful...depends on your headings.
16 set linesize X    -- X should be the sum of the column widths
17 set numw X        -- X should be the length you want for numbers (avoid scientific notation on IDs)
18
19 spool myfile.csv
20 SELECT
21     sistema, avg(tamano), avg(usado), montado
22 FROM
23     DF
24 GROUP BY
25     sistema, montado;
26 EOF
27 )
28
29 # echo "$info" | xargs | sed 's/\s+/,/' myfile.csv
30 }
31
32 sacar_informacion
33
34 comprobacion_postfix(){
35     comando=$(rpm -qa | grep postfix | wc -l)
36
37     if [ "$comando" -ge 1 ]; then
38         echo "POSTFIX esta instalado"
39     else
40         echo "POSTFIX NO esta instalado"
41         echo "Instalando...."
42         dnf install -y postfix mailx
43         echo "Se ha instalado POSTFIX"
44     fi
45 }
46
47 comprobacion_postfix
48
```

```
49 programar_tarea(){
50     cron_schedule="* * * * * /home/oracle/Documents/practica-scripts/enviar-correo.sh"
51 }
52
53 programar_tarea
54
55 enviar_correo(){
56     fichero_csv="/home/oracle/Documents/practica-scripts/myfile.csv"
57     echo "Informe de la tabla DF" | mail -s "Ismael Macareno Chouikh" -a "$fichero_csv" alvaro@alvarogonza
58 }
59
60 enviar_correo
```

6. *units* de systemd

[Adjunto fichero TXT](#)

7. Fichero /etc/crontab

[Adjunto fichero TXT](#)