



Universidad Nacional de Costa Rica
Sede Regional Brunca
Campus PZ

Guia StandBy-Oracle
Administración de Bases de Datos

Estudiante:

Kevin Josue Arauz Jiemenz

Kevin Fallas Chavarria

James Rivera Nuñes

Profesor:

Prof. Máster Carlos Carranza Blanco

2025

Índice general

1 Implementación de Oracle Data Guard Standby - Proyecto Vitalis	1
1.1 Introducción	1
1.1.1 Objetivos del Proyecto	1
1.1.2 Conceptos Fundamentales	1
1.2 Arquitectura del Sistema	2
1.3 Configuración de Docker	2
1.3.1 Archivo docker-compose.yml	2
1.3.2 Dockerfile.vitalis-primary	4
1.3.3 Dockerfile.vitalis-standby	4
1.4 Prerrequisitos del Sistema	4
1.4.1 Requerimientos Hardware	4
1.4.2 Requerimientos Software	4
1.4.3 Configuración de Puertos	5
1.4.4 Configuración de Oracle Container Registry	5
1.5 Estructura del Manual	5
1.6 Resumen de Configuración Inicial	5
1.6.1 Comandos Principales de Despliegue	5
1.6.2 Puertos de Acceso	6
1.6.3 Credenciales por Defecto	6
1.7 Próximos Pasos	6
1.8 Referencias Adicionales	7
2 Configuración Avanzada y Scripts de Administración	8
2.1 Proceso de Inicialización Completo	8
2.2 Script initialize_vitalis.sh (Standby) - Análisis Detallado	8
2.2.1 Configuración del Daemon SSH	8

2.2.2	Creación de Estructura de Directories	9
2.2.3	Configuración Inicial de Base de Datos Oracle	9
2.2.4	Creación del Archivo de Contraseñas Oracle	10
2.2.5	Configuración de Red Oracle (Standby)	10
2.2.6	Reinicio y Finalización del Listener	12
2.3	Configuración SSH entre Contenedores.	12
2.4	Script initialize_vitalis.sh (Primary) - Análisis Detallado	12
2.4.1	Creación de Directorios y Configuración Inicial	12
2.4.2	Configuración de Parámetros Data Guard (Primary)	13
2.4.3	Parámetros Adicionales de Data Guard	13
2.4.4	Parámetros Adicionales de Data Guard	13
2.4.5	Configuración de Data Guard Broker y Redo Logs	14
2.4.6	Creación del Archivo de Contraseñas	15
2.4.7	Configuración de Red Oracle (Primary)	15
2.4.8	Creación y Transferencia del PFILE	16
2.4.9	Verificación de Conectividad SSH	16
2.4.10	Duplicación RMAN	17
2.4.11	Configuración Final de Data Guard	17
2.5	Scripts de Administración y Mantenimiento	18
2.5.1	Script backup_vitalis.sh	18
2.5.2	Script daily_backup.sh	19
2.5.3	Script purge_applied_logs.sh	19
2.5.4	Script purge_complete_logs_in_standby.sh	19
2.6	Verificación del Funcionamiento	19
2.6.1	Verificación en Primary Database	19
2.6.2	Verificación en Standby Database	20
2.7	Resolución de Problemas Comunes.	21
2.7.1	Error de Conectividad SSH	21
2.7.2	Gap en Archive Logs	21
2.7.3	Standby No Aplica Logs	21
3	Procedimientos de Switchover y Conversión a Primary	22
3.1	Introducción al Switchover	22
3.2	Proceso de Switchover - Guía Completa	22
3.2.1	Prerrequisitos para Switchover	22
3.2.2	Verificación del Estado Inicial	23
3.2.3	Proceso Manual de Switchover	23
3.3	Scripts Automatizados de Switchover	25
3.3.1	Script switchover_to_primary.sh	25
3.4	Configuración Post-Switchover	26
3.4.1	Script fix_listener.sh	26

3.5	Procedimiento de Switchover Rápido	30
3.5.1	Guía Simplificada	30
3.5.2	Comandos de Verificación Post-Switchover	30
3.6	Consideraciones Importantes y Limitaciones	31
3.6.1	Impacto del Switchover	31
3.6.2	Recomendaciones de Uso	32
3.6.3	Procedimiento de Rollback	32

1. Implementación de Oracle Data Guard Standby - Proyecto Vitalis

1.1. Introducción

Este manual proporciona una guía completa para la implementación de una base de datos Oracle Data Guard standby para el sistema Vitalis. La configuración utiliza contenedores Docker para facilitar el despliegue y la administración de tanto la base de datos principal como la standby, garantizando alta disponibilidad y protección de datos.

1.1.1. Objetivos del Proyecto

- Implementar una solución de alta disponibilidad para la base de datos Vitalis
- Configurar replicación automática entre base de datos primaria y standby
- Proporcionar capacidad de switchover para continuidad del negocio
- Establecer procedimientos de backup y recuperación automatizados

1.1.2. Conceptos Fundamentales

Oracle Data Guard es una funcionalidad que proporciona alta disponibilidad, protección de datos y recuperación ante desastres para bases de datos Oracle [1]. Mantiene una o más réplicas (standby databases) sincronizadas con la base de datos principal (primary database) [2].

Componentes Principales:

- **Primary Database:** Base de datos principal que recibe todas las transacciones
- **Standby Database:** Réplica de la base de datos principal que se mantiene sincronizada
- **Archive Logs:** Archivos que contienen los cambios realizados en la base de datos
- **Redo Logs:** Logs de transacciones en línea

1.2. Arquitectura del Sistema

La arquitectura implementada utiliza Docker [3] para crear dos contenedores independientes que se comunican a través de una red Docker personalizada:

Componente	Base de Datos Primary	Base de Datos Standby
Contenedor	vitalis-primary	vitalis-standby
Puerto Externo	1521	1522
Hostname	vitalis-primary	vitalis-standby
Rol	PRIMARY (Lectura/Escritura)	PHYSICAL STANDBY (Solo Lectura)

Flujo de Datos entre Contenedores:



Características del Flujo:

- **Archive Logs:** Transferidos automáticamente desde primary a standby
- **Status/Control:** Información de estado y control enviada desde standby a primary
- **Red Docker:** Comunicación interna a través de la red vitalis-network
- **Sincronización:** Proceso continuo de replicación de datos

1.3. Configuración de Docker

1.3.1. Archivo docker-compose.yml

El archivo `docker-compose.yml` es el núcleo de la configuración del sistema. Define dos servicios principales que trabajan en conjunto:

Servicio vitalis-primary

- **Imagen Base:** Construida desde `Dockerfile.vitalis-primary`
- **Hostname:** `vitalis-primary` (para comunicación entre contenedores)
- **Variables de Entorno:**
 - `ORACLE_SID=VITALIS`: Identificador de la base de datos principal
 - `ORACLE_STANDBY_SID=VITALISSB`: Identificador de la base de datos standby
 - `ORACLE_PDB=VITALISPDB1`: Nombre de la base de datos pluggable principal

-
- ORACLE_PWD=VITALIS: Contraseña del usuario sys
 - ENABLE_ARCHIVELOG=true: Habilita el modo archive log (esencial para Data Guard)

- **Puertos Expuestos:**

- 1521:1521: Puerto principal de Oracle para conexiones SQL
- 5500:5500: Oracle Enterprise Manager Express

- **Volúmenes:**

- vitalis-data-primary: Almacena los archivos de datos de Oracle
- vitalis-backup-primary: Directorio para backups
- ./scripts/main: Scripts de administración montados desde el host

- **Recursos:** Limitado a 4GB de RAM con 1GB de memoria compartida

Servicio vitalis-standby

- **Imagen Base:** Construida desde Dockerfile.vitalis-standby

- **Hostname:** vitalis-standby

- **Variables de Entorno:**

- ORACLE_SID=VITALISSB: SID de la base de datos standby
- ORACLE_MAIN_SID=VITALIS: Referencia al SID principal
- ORACLE_PDB=VITALISBPDB1: PDB de la standby
- ORACLE_MAIN_PDB=VITALISPDB1: Referencia a la PDB principal

- **Puertos Expuestos:**

- 1522:1521: Puerto Oracle mapeado a 1522 para evitar conflictos
- 5501:5500: Enterprise Manager en puerto alternativo

- **Dependencias:** depends_on: vitalis-primary asegura el orden de inicio

Configuración de Red y Volúmenes

- **Red personalizada:** vitalis-network con driver bridge permite comunicación entre contenedores

- **Volúmenes persistentes:** Garantizan que los datos persistan entre reinicios de contenedores

- **Política de reinicio:** unless-stopped asegura alta disponibilidad

1.3.2. Dockerfile.vitalis-primary

Este Dockerfile configura el contenedor para la base de datos primaria:

- **Imagen Base:** container-registry.oracle.com/database/enterprise:19.3.0.0
- **Instalación de Herramientas:**
 - vim: Editor de texto para administración
 - expect: Herramienta para automatizar interacciones de línea de comandos
 - openssh-server: Servidor SSH para comunicación con standby
- **Configuración SSH:**
 - Genera claves SSH del host (RSA, ECDSA, ED25519)
 - Configura permisos apropiados para el usuario oracle
 - Inicia el daemon SSH para permitir conexiones entrantes
- **Puertos Exuestos:** 1521 (Oracle) y 22 (SSH)

1.3.3. Dockerfile.vitalis-standby

Similar al Dockerfile primario pero optimizado para la base de datos standby:

- **Diferencias Clave:**
 - No incluye expect ya que la standby principalmente recibe conexiones
 - Configuración SSH idéntica para permitir comunicación bidireccional
 - Mismos puertos expuestos para mantener consistencia
- **Propósito:** Recibir archive logs desde primary y mantener sincronización de datos

1.4. Prerrequisitos del Sistema

1.4.1. Requerimientos Hardware

- Mínimo 8GB de RAM disponible (4GB por contenedor)
- 50GB de espacio en disco libre
- Procesador compatible con virtualización

1.4.2. Requerimientos Software

- Docker Engine (versión 20.10 o superior)
 - Docker Compose (versión 2.0 o superior)
 - Sistema operativo compatible: Linux, Windows 10/11, macOS
-

1.4.3. Configuración de Puertos

- Puerto 1521: Base de datos primary
- Puerto 1522: Base de datos standby
- Puerto 5500: Oracle Enterprise Manager (primary)
- Puerto 5501: Oracle Enterprise Manager (standby)
- Puertos SSH internos para comunicación entre contenedores

1.4.4. Configuración de Oracle Container Registry

CRÍTICO: Antes de proceder, es necesario autenticarse en Oracle Container Registry [4]:

1. Crear cuenta en Oracle: <https://profile.oracle.com>
2. Aceptar los términos de Oracle Standard Terms and Restrictions
3. Ejecutar login en Docker:

```
docker login container-registry.oracle.com
```

4. Ingresar credenciales de Oracle cuando se solicite

Nota: Sin este paso, la construcción de las imágenes Docker fallará.

1.5. Estructura del Manual

Este manual está organizado en los siguientes capítulos:

- **Capítulo 1:** Introducción, arquitectura y configuración Docker (documento actual)
- **Capítulo 2:** Configuración avanzada, scripts de administración y mantenimiento
- **Capítulo 3:** Procedimientos de switchover y conversión a primary

1.6. Resumen de Configuración Inicial

1.6.1. Comandos Principales de Despliegue

1. Construcción y despliegue:

```
docker-compose up -d
```

2. Configuración SSH (en primary):

```
docker exec -it vitalis-primary /bin/bash
ssh-keygen -t rsa -N "" -f ~/.ssh/id_rsa
ssh-copy-id oracle@vitalis-standby
```

3. Inicialización standby:

```
docker exec -it vitalis-standby /bin/bash
cd /home/oracle/scripts
./initialize_vitalis.sh
```

4. Inicialización primary:

```
docker exec -it vitalis-primary /bin/bash
cd /home/oracle/scripts
./initialize_vitalis.sh
```

1.6.2. Puertos de Acceso

- **Primary Database:** localhost:1521
- **Standby Database:** localhost:1522
- **Primary EM Express:** localhost:5500
- **Standby EM Express:** localhost:5501

1.6.3. Credenciales por Defecto

- **Usuario Oracle:** sys
- **Contraseña Primary:** VITALIS
- **Contraseña Standby:** VITALISSB
- **Usuario SSH:** oracle / oracle

1.7. Próximos Pasos

Una vez completada la configuración básica descrita en este capítulo, proceder al **Capítulo 2** para:

- Configuración avanzada de parámetros Data Guard
- Implementación de scripts de backup automatizado
- Configuración de limpieza de archive logs
- Procedimientos de verificación y monitoreo

El **Capítulo 3** cubre los procedimientos de switchover para convertir la base de datos standby en primary cuando sea necesario.

1.8. Referencias Adicionales

Para información complementaria sobre implementación práctica de bases de datos standby, consulte el artículo especializado de Patra [5]. El portal oficial de Oracle Container Registry [6] proporciona acceso directo a las imágenes oficiales de Oracle Database. El código completo de este proyecto está disponible en el repositorio GitHub [7].

2. Configuración Avanzada y Scripts de Administración

2.1. Proceso de Inicialización Completo

Siguiendo la guía de implementación, la configuración debe realizarse en el siguiente orden específico:

1. **Paso 1:** Configurar el contenedor standby (receptor)
2. **Paso 2:** Configurar SSH entre contenedores
3. **Paso 3:** Configurar el contenedor primary (transmisor)

IMPORTANTE: Este orden es crítico porque el standby debe estar preparado para recibir conexiones SSH antes de que el primary intente conectarse.

2.2. Script initialize_vitalis.sh (Standby) - Análisis Detallado

El script del contenedor standby prepara la infraestructura necesaria para recibir la configuración desde el primary.

2.2.1. Configuración del Daemon SSH

```
echo "Iniciando SSH daemon..."  
/usr/sbin/sshd  
  
# Verificar que SSH esté activo  
if ! pgrep -x "sshd" > /dev/null; then  
    echo "ERROR: SSH daemon no se pudo iniciar"  
    exit 1  
fi
```

Función crítica: El daemon SSH es el primer componente que se configura porque:

- Permite que el primary se conecte remotamente al standby
- Facilita la transferencia de archivos de configuración (PFILE)
- Habilita la ejecución de comandos remotos durante la configuración
- Es prerequisito para todo el proceso de duplicación RMAN

2.2.2. Creación de Estructura de Directorios

```
echo "Creando los directorios necesarios."
mkdir -p /opt/oracle/oradata/$ORACLE_SID/recovery_files
mkdir -p /home/oracle/scp/
mkdir -p /home/oracle/scp/recovery_files/
```

Propósito de cada directorio:

- /opt/oracle/oradata/\$ORACLE_SID/recovery_files: Almacena archive logs y backups RMAN del standby
- /home/oracle/scp/: Directorio temporal para transferencias SSH desde primary
- /home/oracle/scp/recovery_files/: Subdirectorio específico para archivos de recuperación

2.2.3. Configuración Inicial de Base de Datos Oracle

```
echo "Configurando parámetros iniciales de Oracle..."
sqlplus sys/$ORACLE_PWD as sysdba <<EOF
    STARTUP NOMOUNT;
    ALTER SYSTEM SET LOCAL_LISTENER =
        '(ADDRESS = (PROTOCOL=TCP)(HOST=vitalis-standby)(PORT=1521))';
    ALTER SYSTEM SET REMOTE_LOGIN_PASSWORDFILE = 'EXCLUSIVE' scope = spfile;
    ALTER SYSTEM SET REMOTE_OS_AUTHENT = FALSE scope = spfile;

    -- Parámetros para Data Guard Broker
    ALTER SYSTEM SET DG_BROKER_START=TRUE SCOPE=SPFILE;
    ALTER SYSTEM SET DG_BROKER_CONFIG_FILE1=
        '/opt/oracle/oradata/$ORACLE_SID/dr1$ORACLE_SID.dat' SCOPE=SPFILE;
    ALTER SYSTEM SET DG_BROKER_CONFIG_FILE2=
        '/opt/oracle/oradata/$ORACLE_SID/dr2$ORACLE_SID.dat' SCOPE=SPFILE;
    EXIT;
EOF
```

Parámetros esenciales configurados:

- STARTUP NOMOUNT: Inicia la instancia en modo nomount para configuración inicial
-

-
- LOCAL_LISTENER: Define la dirección de escucha específica del standby
 - REMOTE_LOGIN_PASSWORDFILE: Habilita autenticación remota vía archivo de contraseñas
 - REMOTE_OS_AUTHENT: Desactiva autenticación por SO para mayor seguridad
 - DG_BROKER_START: Prepara para Data Guard Broker (administración avanzada)
 - DG_BROKER_CONFIG_FILE1/2: Define ubicación de archivos de configuración del broker

2.2.4. Creación del Archivo de Contraseñas Oracle

```
echo "Creando archivo de contraseñas Oracle..."  
orapwd file=$ORACLE_HOME/dbs/orapw$ORACLE_SID  
      password=$ORACLE_MAIN_SID-$ORACLE_SID-1  
      entries=10 force=y
```

Importancia del archivo de contraseñas:

- Permite que el primary se autentique remotamente como SYSDBA en el standby
- La contraseña debe coincidir entre primary y standby para Data Guard
- entries=10: Permite hasta 10 usuarios con privilegios SYSDBA/SYSOPER
- force=y: Sobrescribe archivo existente si lo hay

2.2.5. Configuración de Red Oracle (Standby)

Archivo tnsnames.ora

```
cat <<EOF > $ORACLE_HOME/network/admin/tnsnames.ora  
$ORACLE_SID=localhost:1521/$ORACLE_PWD  
$ORACLE_PDB=  
(DESCRIPTION =  
  (ADDRESS = (PROTOCOL = TCP)(HOST = 0.0.0.0)(PORT = 1521))  
  (CONNECT_DATA =  
    (SERVER = DEDICATED)  
    (SERVICE_NAME = $ORACLE_PDB)  
  )  
)  
  
$ORACLE_MAIN_SID=  
(DESCRIPTION =  
  (ADDRESS = (PROTOCOL = TCP)(HOST = vitalis-primary)(PORT = 1521))  
  (CONNECT_DATA =  
    (SERVER = DEDICATED)  
    (SERVICE_NAME = $ORACLE_MAIN_SID))
```

```
)  
)  
EOF
```

Configuraciones de conectividad:

- Conexión local al standby: \$ORACLE_SID=localhost:1521/\$ORACLE_PWD
- Conexión a PDB local: Configurada para acceso interno
- Conexión al primary remoto: HOST = vitalis-primary (hostname del contenedor)
- Protocolo TCP dedicado para todas las conexiones

Archivo listener.ora

```
cat <<EOF > $ORACLE_HOME/network/admin/listener.ora
LISTENER =
(DESCRIPTION_LIST =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC1))
    (ADDRESS = (PROTOCOL = TCP)(HOST = 0.0.0.0)(PORT = 1521))
  )
)

SID_LIST_LISTENER =
  (SID_LIST =
    (SID_DESC = (GLOBAL_DBNAME = $ORACLE_SID)
      (ORACLE_HOME = $ORACLE_HOME)
      (SID_NAME = $ORACLE_SID)
    )
  )

DEDICATED_THROUGH_BROKER_LISTENER=ON
DIAG_ADR_ENABLED = off
EOF
```

Elementos clave del listener:

- HOST = 0.0.0.0: Escucha en todas las interfaces de red
- PORT = 1521: Puerto estándar de Oracle
- PROTOCOL = IPC: Comunicación inter-procesos local
- SID_LIST_LISTENER: Registro estático del SID standby
- DEDICATED_THROUGH_BROKER_LISTENER=ON: Optimización para Data Guard

2.2.6. Reinicio y Finalización del Listener

```
echo "Reiniciando listener..."  
lsnrctl stop  
sleep 2  
lsnrctl start  
  
echo "Deteniendo base de datos para finalizar configuración..."  
/home/oracle/shutDown.sh immediate
```

Proceso de finalización:

- Reinicia el listener con la nueva configuración
- Detiene la base de datos dejándola lista para la configuración desde primary
- El SSH daemon permanece activo para recibir conexiones

2.3. Configuración SSH entre Contenedores

Antes de proceder con el primary, es crítico configurar SSH sin contraseña:

```
# En el contenedor primary:  
docker exec -it vitalis-primary /bin/bash  
ssh-keygen -t rsa -b 2048 -f ~/.ssh/id_rsa -N ""  
ssh-copy-id oracle@vitalis-standby  
# Contraseña: oracle
```

2.4. Script initialize_vitalis.sh (Primary) - Análisis Detallado

Una vez configurado el standby y establecida la conectividad SSH, se procede a configurar el primary que actuará como transmisor de datos.

2.4.1. Creación de Directorios y Configuración Inicial

```
echo "Creando los directorios necesarios."  
mkdir -p /opt/oracle/oradata/$ORACLE_SID/recovery_files  
mkdir /home/oracle/scp/
```

Directorios del primary:

- recovery_files: Almacena archive logs locales y backups RMAN
- scp: Directorio temporal para archivos que se enviarán al standby

2.4.2. Configuración de Parámetros Data Guard (Primary)

El script configura una serie completa de parámetros esenciales para Data Guard:

```
sqlplus sys/$ORACLE_SID-$ORACLE_STANDBY_SID-1 as sysdba <<EOF
    ALTER SYSTEM SET DB_UNIQUE_NAME=$ORACLE_SID SCOPE=SPFILE;
    ALTER SYSTEM SET DB_RECOVERY_FILE_DEST_SIZE=40G SCOPE=BOTH;
    ALTER SYSTEM SET DB_RECOVERY_FILE_DEST=
        '/opt/oracle/oradata/$ORACLE_SID/recovery_files' SCOPE=BOTH;
    ALTER SYSTEM SET LOG_ARCHIVE_CONFIG=
        'DG_CONFIG=($ORACLE_SID,$ORACLE_STANDBY_SID)';
    ALTER SYSTEM SET LOG_ARCHIVE_DEST_1=
        'LOCATION=USE_DB_RECOVERY_FILE_DEST VALID_FOR=
            (ALL_LOGFILES,ALL_ROLES) DB_UNIQUE_NAME=$ORACLE_SID
            MANDATORY REOPEN=60' SCOPE=BOTH;
    ALTER SYSTEM SET LOG_ARCHIVE_DEST_2=
        'SERVICE=$ORACLE_STANDBY_SID ASYNC VALID_FOR=
            (ONLINE_LOGFILES, PRIMARY_ROLE) DB_UNIQUE_NAME=
            $ORACLE_STANDBY_SID DELAY=10' SCOPE=BOTH;
EOF
```

Parámetros fundamentales de Data Guard:

- **DB_UNIQUE_NAME**: Identifica únicamente esta base de datos en la configuración Data Guard
- **DB_RECOVERY_FILE_DEST**: Ubicación para archive logs y backups (40GB de espacio)
- **LOG_ARCHIVE_CONFIG**: Define la configuración Data Guard con ambas bases de datos
- **LOG_ARCHIVE_DEST_1**: Destino local para archive logs (obligatorio)
- **LOG_ARCHIVE_DEST_2**: Destino remoto (standby) con delay de 10 segundos para protección

2.4.3. Parámetros Adicionales de Data Guard

2.4.4. Parámetros Adicionales de Data Guard

Habilitación de Destinos de Archive Log

```
ALTER SYSTEM SET log_archive_dest_state_1=ENABLE SCOPE=BOTH;
ALTER SYSTEM SET log_archive_dest_state_2=ENABLE SCOPE=BOTH;
ALTER SYSTEM SET standby_file_management=AUTO SCOPE=BOTH;
```

Configuración de Temporización y Formato

```
ALTER SYSTEM SET ARCHIVE_LAG_TARGET=300 SCOPE=BOTH;
ALTER SYSTEM SET REMOTE_LOGIN_PASSWORDFILE=EXCLUSIVE SCOPE=SPFILE;
ALTER SYSTEM SET LOG_ARCHIVE_FORMAT='%t_%s_%r.arc' SCOPE=SPFILE;
```

Configuración de Recuperación Automática (FAL)

```
ALTER SYSTEM SET FAL_SERVER=$ORACLE_STANDBY_SID;
ALTER SYSTEM SET FAL_CLIENT=$ORACLE_SID;
```

Conversión de Nombres de Archivos

```
ALTER SYSTEM SET DB_FILE_NAME_CONVERT=
  '/$ORACLE_STANDBY_SID/' , '$ORACLE_SID/' SCOPE=SPFILE;
ALTER SYSTEM SET LOG_FILE_NAME_CONVERT=
  '/$ORACLE_STANDBY_SID/' , '$ORACLE_SID/' SCOPE=SPFILE;
```

Parámetros críticos para el proyecto:

- ARCHIVE_LAG_TARGET=300: Cumple requerimiento - actualización cada 5 minutos (300 segundos)
- log_archive_dest_state_1/2: Habilita ambos destinos de archive logs
- standby_file_management: Automatiza la gestión de archivos en standby
- FAL_SERVER/CLIENT: Para recuperación automática de gaps (Fetch Archive Log)
- db_file_name_convert: Convierte nombres de archivos entre primary y standby
- LOG_ARCHIVE_FORMAT: Define formato de nombres de archive logs

2.4.5. Configuración de Data Guard Broker y Redo Logs

```
-- Parámetros para Data Guard Broker y Fast-Start Failover
ALTER SYSTEM SET DG_BROKER_START=TRUE SCOPE=BOTH;
ALTER SYSTEM SET DG_BROKER_CONFIG_FILE1=
  '/opt/oracle/oradata/$ORACLE_SID/dr1$ORACLE_SID.dat' SCOPE=SPFILE;
ALTER SYSTEM SET DG_BROKER_CONFIG_FILE2=
  '/opt/oracle/oradata/$ORACLE_SID/dr2$ORACLE_SID.dat' SCOPE=SPFILE;

-- Agregar redo logs adicionales de 50MB cada uno
ALTER DATABASE ADD LOGFILE GROUP 4
  ('$ORACLE_BASE/oradata/$ORACLE_SID/recovery_files/redo04.log') SIZE 50M;
ALTER DATABASE ADD LOGFILE GROUP 5
  ('$ORACLE_BASE/oradata/$ORACLE_SID/recovery_files/redo05.log') SIZE 50M;
ALTER DATABASE ADD LOGFILE GROUP 6
  ('$ORACLE_BASE/oradata/$ORACLE_SID/recovery_files/redo06.log') SIZE 50M;

ALTER SYSTEM SWITCH LOGFILE;
ALTER SYSTEM CHECKPOINT;
```

Configuración avanzada:

- **Data Guard Broker:** Habilita administración avanzada y automatización
-

-
- **Redo logs de 50MB: Cumple requerimiento - switch automático al llenarse**
 - **Múltiples grupos:** Mejora el rendimiento de escritura y replicación
 - **Switch y Checkpoint:** Fuerza escritura inmediata para sincronización

2.4.6. Creación del Archivo de Contraseñas

```
echo "Eliminados. Creando orapwd"
orapwd file=$ORACLE_HOME/dbs/orapw$ORACLE_SID
        password=$ORACLE_SID-$ORACLE_STANDBY_SID-1
        entries=10 force=y
```

Propósito: Crea un archivo de contraseñas compatible que permite la autenticación remota entre primary y standby usando la misma contraseña.

2.4.7. Configuración de Red Oracle (Primary)

tnsnames.ora

```
$ORACLE_SID=localhost:1521/$ORACLE_PWD
$ORACLE_PDB=
(DESCRIPTION =
  (ADDRESS = (PROTOCOL = TCP)(HOST = 0.0.0.0)(PORT = 1521))
  (CONNECT_DATA =
    (SERVER = DEDICATED)
    (SERVICE_NAME = $ORACLE_PDB)
  )
)

$ORACLE_STANDBY_SID=
(DESCRIPTION =
  (ADDRESS = (PROTOCOL = TCP)(HOST = vitalis-standby)(PORT = 1521))
  (CONNECT_DATA =
    (SERVER = DEDICATED)
    (SERVICE_NAME = $ORACLE_STANDBY_SID)
  )
)
```

Elementos clave:

- Alias local para conexión directa al primary
 - Alias remoto para conexión al standby usando hostname del contenedor
 - Configuración de PDBs para ambos sistemas
-

2.4.8. Creación y Transferencia del PFILE

```
sqlplus sys/$ORACLE_SID-$ORACLE_STANDBY_SID-1 as sysdba <<EOF
CREATE PFILE='/home/oracle/scp/init$ORACLE_STANDBY_SID.ora' FROM SPFILE;
EXIT;
EOF
```

El script modifica automáticamente el PFILE para el standby mediante tres transformaciones específicas:

Cambio de Identificadores SID

```
sed -i "s|$ORACLE_SID|$ORACLE_STANDBY_SID|g;" 
"/home/oracle/scp/init$ORACLE_STANDBY_SID.ora"
```

Inversión de Conversión de Archivos de Datos

```
sed -i "s|'/$ORACLE_SID|','/$ORACLE_STANDBY_SID||'|
'|'$ORACLE_STANDBY_SID|','|$ORACLE_SID'||g;" 
"/home/oracle/scp/init$ORACLE_STANDBY_SID.ora"
```

Ajuste de Conversión de Log Files

```
sed -i "s|*.log_file_name_convert='/$ORACLE_STANDBY_SID|','/$ORACLE_SID||'|
|*.log_file_name_convert='|$ORACLE_SID|','|$ORACLE_STANDBY_SID||'lg;" 
"/home/oracle/scp/init$ORACLE_STANDBY_SID.ora"
```

Modificaciones realizadas:

- Cambia todos los identificadores de SID
- Invierte la conversión de nombres de archivos
- Adapta rutas específicas para el standby

2.4.9. Verificación de Conectividad SSH

```
SSH_READY=false
RETRY_COUNT=0
MAX_RETRIES=30

while [ "$SSH_READY" = false ] && [ $RETRY_COUNT -lt $MAX_RETRIES ]; do
    if ssh -o ConnectTimeout=5 -o StrictHostKeyChecking=no
        oracle@vitalis-standby "echo 'SSH Ready'" >/dev/null 2>&1; then
            SSH_READY=true
            echo " Conexión SSH establecida con standby"
        else
            ((RETRY_COUNT++))
    fi
done
```

```
        echo "Esperando conexión SSH... intento $((RETRY_COUNT + 1))/$MAX_RETRIES"
        sleep 5
        RETRY_COUNT=$((RETRY_COUNT + 1))
    fi
done
```

Función: Implementa un mecanismo de retry robusto que espera hasta que SSH esté disponible en el contenedor standby antes de continuar.

2.4.10. Duplicación RMAN

El corazón del proceso de configuración es la duplicación de la base de datos usando RMAN (Recovery Manager) [8]:

```
rman TARGET sys/$ORACLE_SID-$ORACLE_STANDBY_SID-1@$ORACLE_SID
      AUXILIARY sys/$ORACLE_SID-$ORACLE_STANDBY_SID-1@$ORACLE_STANDBY_SID <<EOF
DUPLICATE TARGET DATABASE FOR STANDBY FROM ACTIVE DATABASE
DORECOVER
NOFILENAMECHECK;
EOF
```

Parámetros del comando DUPLICATE:

- FOR STANDBY: Especifica que se crea una base de datos standby
- FROM ACTIVE DATABASE: Realiza la duplicación en línea (sin backups previos)
- DORECOVER: Aplica automáticamente todos los archive logs disponibles
- NOFILENAMECHECK: Omite verificaciones de nombres de archivos (necesario para contenedores)

2.4.11. Configuración Final de Data Guard

Inicio de Managed Recovery

```
rman target sys/$ORACLE_SID-$ORACLE_STANDBY_SID-1@$ORACLE_STANDBY_SID <<EOF
CROSSCHECK ARCHIVELOG ALL;
ALTER DATABASE RECOVER MANAGED STANDBY DATABASE DISCONNECT FROM SESSION;
EOF
```

Funciones:

- CROSSCHECK: Verifica la integridad de todos los archive logs
 - RECOVER MANAGED: Inicia el proceso automático de aplicación de logs
 - DISCONNECT FROM SESSION: Permite que el proceso continúe en background
-

Configuración de Políticas RMAN

```
rman TARGET sys/$ORACLE_SID-$ORACLE_STANDBY_SID-1@$ORACLE_SID <<EOF
CONFIGURE CONTROLFILE AUTOBACKUP ON;
CONFIGURE ARCHIVELOG DELETION POLICY TO SHIPPED TO ALL STANDBY;
EOF
```

Políticas configuradas:

- CONTROLFILE AUTOBACKUP: Backup automático del controlfile tras cambios estructurales
- ARCHIVELOG DELETION POLICY: Los archive logs solo se eliminan después de ser enviados al standby

2.5. Scripts de Administración y Mantenimiento

2.5.1. Script backup_vitalis.sh

Este script realiza backups completos de la base de datos y los transfiere al standby.

Configuración de Variables

```
export ORACLE_HOME=/opt/oracle/product/19c/dbhome_1
export ORACLE_SID=VITALIS
export ORACLE_STANDBY_SID=VITALISSB
SOURCE_FOLDER="/opt/oracle/oradata/VITALIS/recovery_files/VITALIS"
DEST_FOLDER="/opt/oracle/oradata/VITALISSB/recovery_files"
```

Proceso de Backup

```
$ORACLE_HOME/bin/rman target sys/VITALIS-VITALISSB-1@VITALIS <<EOF
BACKUP DATABASE PLUS ARCHIVELOG;
EOF
```

Comando explicado:

- BACKUP DATABASE: Realiza backup completo de todos los datafiles
- PLUS ARCHIVELOG: Incluye todos los archive logs en el backup
- Se ejecuta antes y después del backup de datos para asegurar consistencia

Transferencia al Standby

```
scp -r "${SOURCE_FOLDER}" oracle@vitalis-standby:"${DEST_FOLDER}"
```

Función: Copia recursivamente todos los archivos de backup al standby usando SCP para garantizar disponibilidad de backups en ambos sitios.

2.5.2. Script daily_backup.sh

Idéntico a backup_vitalis.sh, diseñado para ejecución automática diaria mediante cron o schedulers.

Uso recomendado:

```
# Agregar al crontab del contenedor primary:  
0 2 * * * /home/oracle/scripts/daily_backup.sh >> /var/log/backup.log 2>&1
```

2.5.3. Script purge_applied_logs.sh

Gestiona la limpieza de archive logs en el primary después de confirmar que fueron aplicados en el standby.

Proceso de Limpieza

```
/opt/oracle/product/19c/dbhome_1/bin/rman target  
sys/$ORACLE_SID-$ORACLE_STANDBY_SID-1@$ORACLE_SID <<EOF  
CROSSCHECK ARCHIVELOG ALL;  
DELETE NOPROMPT ARCHIVELOG ALL;  
EOF
```

Comandos explicados:

- CROSSCHECK ARCHIVELOG ALL: Verifica el estado de todos los archive logs
- DELETE NOPROMPT ARCHIVELOG ALL: Elimina archive logs que ya fueron enviados al standby
- La política configurada previamente protege contra eliminación prematura

2.5.4. Script purge_complete_logs_in_standby.sh

Script que se ejecuta en el primary pero limpia logs en el standby via SSH.

```
ssh oracle@vitalis-standby "/home/oracle/scripts/delete_obsolete_vitalis.sh"
```

Función: Ejecuta remotamente el script de limpieza en el standby, manteniendo centralizadas las operaciones de mantenimiento.

2.6. Verificación del Funcionamiento

2.6.1. Verificación en Primary Database

Estado de Data Guard

```
SELECT database_role, protection_mode, protection_level  
FROM v$database;
```

Resultado esperado:

- DATABASE_ROLE: PRIMARY
- PROTECTION_MODE: MAXIMUM PERFORMANCE
- PROTECTION_LEVEL: MAXIMUM PERFORMANCE

Estado de Archive Logs

```
SELECT dest_id, status, destination, error
FROM v$archive_dest
WHERE dest_id IN (1,2);
```

Verificaciones:

- DEST_ID 1: STATUS = VALID (destino local)
- DEST_ID 2: STATUS = VALID (destino standby)
- ERROR debe estar vacío o NULL

2.6.2. Verificación en Standby Database

Estado de la Base de Datos

```
SELECT database_role, open_mode FROM v$database;
```

Resultado esperado:

- DATABASE_ROLE: PHYSICAL STANDBY
- OPEN_MODE: MOUNTED (en modo recovery) o READ ONLY WITH APPLY

Verificación de Aplicación de Logs

```
SELECT sequence#, applied, completion_time
FROM v$archived_log
WHERE applied = 'YES'
ORDER BY sequence# DESC;
```

Indicadores de salud:

- Secuencias continuas sin gaps
- APPLIED = YES para logs recientes
- COMPLETION_TIME actualizado regularmente

2.7. Resolución de Problemas Comunes

2.7.1. Error de Conectividad SSH

Síntoma: "Permission denied" o "Connection refused"

Solución:

```
# En ambos contenedores:  
/usr/sbin/sshd  
ssh-keygen -t rsa -N "" -f ~/.ssh/id_rsa  
ssh-copy-id oracle@[hostname-destino]
```

2.7.2. Gap en Archive Logs

Síntoma: v\$archive_gap muestra filas

Solución:

```
# En standby:  
ALTER DATABASE RECOVER MANAGED STANDBY DATABASE CANCEL;  
ALTER DATABASE RECOVER MANAGED STANDBY DATABASE  
    FROM 'ruta_del_archive_log_faltante';  
ALTER DATABASE RECOVER MANAGED STANDBY DATABASE DISCONNECT;
```

2.7.3. Standby No Aplica Logs

Verificación:

```
SELECT process, status, sequence# FROM v$managed_standby;
```

Solución:

```
ALTER DATABASE RECOVER MANAGED STANDBY DATABASE CANCEL;  
ALTER DATABASE RECOVER MANAGED STANDBY DATABASE DISCONNECT;
```

3. Procedimientos de Switchover y Conversión a Primary

3.1. Introducción al Switchover

El switchover es un proceso planificado que convierte una base de datos standby en la nueva base de datos primaria. Este procedimiento es esencial para:

- Mantenimiento planificado del servidor primary
- Migración a nuevo hardware
- Testing de la configuración de alta disponibilidad
- Distribución geográfica de cargas de trabajo

IMPORTANTE: El switchover es un proceso irreversible que cambia permanentemente los roles de las bases de datos.

3.2. Proceso de Switchover - Guía Completa

3.2.1. Prerrequisitos para Switchover

Antes de iniciar el switchover, verificar:

1. **Sincronización completa:** No debe haber gaps en archive logs
2. **Ambas bases de datos operativas:** Primary y standby funcionando correctamente
3. **Conectividad de red:** Comunicación estable entre contenedores
4. **Espacio disponible:** Suficiente espacio en disco en ambos sistemas
5. **Backup reciente:** Backup completo previo al switchover

3.2.2. Verificación del Estado Inicial

En la Base de Datos Standby

```
docker exec -it vitalis-standby /bin/bash
sqlplus sys/VITALISSB as sysdba

-- Verificar estado actual
SELECT database_role, open_mode FROM v$database;
-- Resultado esperado: PHYSICAL STANDBY, MOUNTED

-- Verificar ausencia de gaps
SELECT * FROM v$archive_gap;
-- Resultado esperado: No rows selected

-- Verificar último log aplicado
SELECT sequence#, applied FROM v$archived_log
WHERE applied = 'YES'
ORDER BY sequence# DESC;
```

En la Base de Datos Primary

```
docker exec -it vitalis-primary /bin/bash
sqlplus sys/VITALIS as sysdba

-- Verificar estado de destinos de archive
SELECT dest_id, status, error FROM v$archive_dest
WHERE dest_id IN (1,2);
-- Ambos destinos deben tener STATUS = VALID

-- Generar switch de log para asegurar sincronización
ALTER SYSTEM SWITCH LOGFILE;
```

3.2.3. Proceso Manual de Switchover

Paso 1: Detener Aplicación Automática de Logs

En la base de datos standby:

```
sqlplus sys/VITALISSB as sysdba

-- Cancelar aplicación automática
ALTER DATABASE RECOVER MANAGED STANDBY DATABASE CANCEL;
```

Efecto: Detiene el proceso MRP (Managed Recovery Process) que aplica automáticamente los archive logs recibidos.

Paso 2: Aplicar Todos los Logs Pendientes

```
-- Forzar aplicación de todos los logs disponibles  
ALTER DATABASE RECOVER MANAGED STANDBY DATABASE FINISH;
```

Función crítica: Este comando asegura que la standby tenga exactamente los mismos datos que la primary al momento del switchover. El proceso puede tardar varios minutos dependiendo de la cantidad de logs pendientes.

Paso 3: Activar la Standby como Nueva Primary

```
-- COMANDO IRREVERSIBLE: Activar como primary  
ALTER DATABASE ACTIVATE STANDBY DATABASE;
```

ADVERTENCIA: Este comando es irreversible. Una vez ejecutado:

- La base de datos cambia su rol a PRIMARY permanentemente
- Se reinicia la numeración de SCN (System Change Numbers)
- La antigua primary no puede volver a ser primary sin reconstrucción completa
- Se activan todos los procesos necesarios para aceptar transacciones

Paso 4: Abrir la Nueva Base de Datos Primary

```
-- Abrir en modo lectura/escritura  
ALTER DATABASE OPEN;
```

```
-- Abrir la PDB  
ALTER PLUGGABLE DATABASE VITALISPDB1 OPEN;
```

Paso 5: Verificación Final

```
-- Verificar nuevo estado  
SELECT database_role, open_mode FROM v$database;  
-- Resultado esperado: PRIMARY, READ WRITE  
  
-- Verificar estado de PDB  
SELECT name, open_mode FROM v$pdb$ WHERE name = 'VITALISPDB1';  
-- Resultado esperado: VITALISPDB1, READ WRITE  
  
-- Probar transacción de escritura  
CREATE TABLE test_switchover (id NUMBER, fecha DATE);  
INSERT INTO test_switchover VALUES (1, SYSDATE);  
COMMIT;
```

3.3. Scripts Automatizados de Switchover

3.3.1. Script switchover_to_primary.sh

Este script automatiza todo el proceso de switchover para reducir errores humanos y tiempo de inactividad.

Verificación Inicial Automatizada

```
echo "1. Verificando estado actual de la base de datos standby..."  
sqlplus -s sys/$ORACLE_PWD as sysdba <<EOF  
SET PAGESIZE 0  
SELECT 'Estado actual: ' || DATABASE_ROLE FROM V$DATABASE;  
SELECT 'Modo de acceso: ' || OPEN_MODE FROM V$DATABASE;  
EXIT;  
EOF
```

Función: Muestra el estado actual antes del switchover para confirmar que se está ejecutando en la base de datos correcta.

Secuencia de Comandos Automatizada

Detener aplicación de logs:

```
echo "2. Deteniendo la aplicación de logs en standby..."  
sqlplus -s sys/$ORACLE_PWD as sysdba <<EOF  
ALTER DATABASE RECOVER MANAGED STANDBY DATABASE CANCEL;  
EXIT;  
EOF
```

Aplicar logs pendientes:

```
echo "3. Finalizando la aplicación de todos los logs pendientes..."  
sqlplus -s sys/$ORACLE_PWD as sysdba <<EOF  
ALTER DATABASE RECOVER MANAGED STANDBY DATABASE FINISH;  
EXIT;  
EOF
```

Activar como primary:

```
echo "4. Activando la base de datos standby como primaria..."  
sqlplus -s sys/$ORACLE_PWD as sysdba <<EOF  
ALTER DATABASE ACTIVATE STANDBY DATABASE;  
EXIT;  
EOF
```

Abrir base de datos:

```
echo "5. Abriendo la base de datos en modo READ WRITE..."  
sqlplus -s sys/$ORACLE_PWD as sysdba <<EOF  
ALTER DATABASE OPEN;  
EXIT;  
EOF  
  
echo "6. Abriendo la PDB..."  
sqlplus -s sys/$ORACLE_PWD as sysdba <<EOF  
ALTER PLUGGABLE DATABASE $ORACLE_MAIN_PDB OPEN;  
EXIT;  
EOF
```

Verificación Final del Script

```
echo "7. Verificando el estado final..."  
sqlplus -s sys/$ORACLE_PWD as sysdba <<EOF  
SET PAGESIZE 0  
SELECT 'Nuevo estado: ' || DATABASE_ROLE FROM V$DATABASE;  
SELECT 'Modo de acceso: ' || OPEN_MODE FROM V$DATABASE;  
SELECT 'PDB Estado: ' || NAME || ' - ' || OPEN_MODE  
FROM V$PDBS WHERE NAME = '$ORACLE_MAIN_PDB';  
EXIT;  
EOF
```

Información de conexión post-switchover:

```
echo "La base de datos standby ahora es la nueva primaria"  
echo "Puedes conectarte usando:"  
echo " - CDB: sqlplus sys/$ORACLE_PWD@localhost:1521/$ORACLE_SID as sysdba"  
echo " - PDB: sqlplus sys/$ORACLE_PWD@localhost:1521/$ORACLE_MAIN_PDB as sysdba"
```

3.4. Configuración Post-Switchover

3.4.1. Script fix_listener.sh

Después del switchover, es necesario reconfigurar el listener para que refleje el nuevo rol de la base de datos.

Validación de Entorno

```
if [ "$ORACLE_SID" != "VITALISSB" ]; then  
    echo "ERROR: Este script debe ejecutarse en el contenedor vitalis-standby"  
    echo "ORACLE_SID actual: $ORACLE_SID"  
    exit 1  
fi
```

Función: Verificación de seguridad para asegurar que el script se ejecuta en el contenedor correcto.

Reconfiguración del Listener

Detener listener actual:

```
echo "Paso 1: Deteniendo el listener actual..."  
lsnrctl stop
```

Nuevo listener.ora:

```
cat <<EOF > $ORACLE_HOME/network/admin/listener.ora  
LISTENER =  
(DESCRIPTION_LIST =  
  (DESCRIPTION =  
    (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC1))  
    (ADDRESS = (PROTOCOL = TCP)(HOST = 0.0.0.0)(PORT = 1521))  
  )  
)  
  
SID_LIST_LISTENER =  
  (SID_LIST =  
    (SID_DESC = (GLOBAL_DBNAME = VITALISSB)  
      (ORACLE_HOME = $ORACLE_HOME)  
      (SID_NAME = VITALISSB)  
    )  
)  
  
# Permitir registro dinámico de servicios  
DEDICATED_THROUGH_BROKER_LISTENER=ON  
DIAG_ADR_ENABLED = off  
EOF
```

Cambios clave:

- Actualiza GLOBAL_DBNAME para reflejar el nuevo rol
- Mantiene DEDICATED_THROUGH_BROKER_LISTENER=ON para Data Guard
- Configura escucha en todas las interfaces

Configuración de tnsnames.ora

```
cat <<EOF > $ORACLE_HOME/network/admin/tnsnames.ora  
# Configuración para CDB (acceso interno)  
VITALISSB=  
(DESCRIPTION =
```

```
(ADDRESS = (PROTOCOL = TCP)(HOST = 0.0.0.0)(PORT = 1521))
(CONNECT_DATA =
  (SERVER = DEDICATED)
  (SERVICE_NAME = VITALISSB)
)
)

# Configuración para PDB (acceso interno)
VITALISPDB1=
(DESCRIPTION =
  (ADDRESS = (PROTOCOL = TCP)(HOST = 0.0.0.0)(PORT = 1521))
  (CONNECT_DATA =
    (SERVER = DEDICATED)
    (SERVICE_NAME = VITALISPDB1)
  )
)

# Configuración para acceso desde host externo al CDB
VITALISSB_EXT=
(DESCRIPTION =
  (ADDRESS = (PROTOCOL = TCP)(HOST = vitalis-standby)(PORT = 1521))
  (CONNECT_DATA =
    (SERVER = DEDICATED)
    (SERVICE_NAME = VITALISSB)
  )
)
EOF
```

Configuraciones incluidas:

- Acceso interno (0.0.0.0) para conexiones dentro del contenedor
- Acceso externo (vitalis-standby) para conexiones desde otros contenedores
- Alias separados para CDB y PDB

Configuración de Servicios Oracle

```
echo "Paso 5: Configurando Oracle para registrar servicios automáticamente . . ."
sqlplus sys/$ORACLE_PWD as sysdba <<EOF
-- Configurar parámetros para registro automático
ALTER SYSTEM SET LOCAL_LISTENER =
  '(ADDRESS = (PROTOCOL=TCP)(HOST=0.0.0.0)(PORT=1521))' SCOPE=BOTH;
ALTER SYSTEM SET SERVICE_NAMES = 'VITALISSB,VITALISPDB1' SCOPE=BOTH;

-- Registrar servicios manualmente
```

```
ALTER SYSTEM REGISTER;

-- Verificar que la PDB esté abierta
SELECT name, open_mode FROM v\$pdbs;

-- Si la PDB no está abierta, abrirla
ALTER PLUGGABLE DATABASE VITALISPDB1 OPEN;

-- Crear servicio específico para la PDB
BEGIN
    DBMS_SERVICE.CREATE_SERVICE(
        SERVICE_NAME => 'VITALISPDB1',
        NETWORK_NAME => 'VITALISPDB1'
    );
END;
/
-- Iniciar el servicio
BEGIN
    DBMS_SERVICE.START_SERVICE('VITALISPDB1');
END;
/
-- Registrar nuevamente todos los servicios
ALTER SYSTEM REGISTER;
EXIT;
EOF
```

Configuraciones realizadas:

- LOCAL_LISTENER: Define la dirección local del listener
- SERVICE_NAMES: Lista los servicios disponibles
- ALTER SYSTEM REGISTER: Fuerza el registro de servicios en el listener
- DBMS_SERVICE: Crea y gestiona servicios específicos para PDB

Verificaciones Finales

```
echo "Paso 6: Verificando el estado del listener..."
sleep 3
lsnrctl status

echo "Paso 7: Verificando servicios registrados..."
lsnrctl services
```

Información de conexión final:

```
echo "Ahora puedes conectarte usando:"
echo ""
echo "Desde DENTRO del contenedor:"
echo "  CDB: sqlplus sys/$ORACLE_PWD@localhost:1521/VITALISSB as sysdba"
echo "  PDB: sqlplus sys/$ORACLE_PWD@localhost:1521/VITALISPDB1 as sysdba"
echo ""
echo "Desde FUERA del contenedor (tu máquina):"
echo "  CDB: sqlplus sys/$ORACLE_PWD@localhost:1522/VITALISSB as sysdba"
echo "  PDB: sqlplus sys/$ORACLE_PWD@localhost:1522/VITALISPDB1 as sysdba"
```

3.5. Procedimiento de Switchover Rápido

3.5.1. Guía Simplificada

Para usuarios experimentados, el switchover se puede realizar con estos comandos simples:

1. **Conectarse al contenedor standby:**

```
docker exec -it vitalis-standby /bin/bash
```

2. **Ejecutar switchover automatizado:**

```
cd /home/oracle/scripts
./switchover_to_primary.sh
./fix_listener.sh
```

3. **Verificar funcionamiento:**

```
sqlplus sys/VITALISSB@localhost:1521/VITALISSB as sysdba
SELECT database_role FROM v$database;
```

3.5.2. Comandos de Verificación Post-Switchover

Verificar Estado de la Base de Datos

```
-- Conectar a la nueva primary
sqlplus sys/VITALISSB@localhost:1522/VITALISSB as sysdba

-- Verificar rol y modo
SELECT database_role, open_mode FROM v$database;
```

```
-- Resultado esperado: PRIMARY, READ WRITE

-- Verificar PDBs
SELECT name, open_mode FROM v$pdbs;
-- VITALISPDB1 debe estar en READ WRITE
```

Verificar Datos

```
-- Conectar a la PDB
sqlplus sys/VITALISSB@localhost:1522/VITALISPDB1 as sysdba

-- Ver tablas del usuario VITALIS (si existen)
SELECT table_name FROM all_tables WHERE owner = 'VITALIS';

-- Probar inserción de datos
CREATE TABLE test_post_switchover (
    id NUMBER PRIMARY KEY,
    mensaje VARCHAR2(100),
    fecha DATE DEFAULT SYSDATE
);

INSERT INTO test_post_switchover VALUES (1, 'Switchover exitoso', SYSDATE);
COMMIT;

-- Verificar inserción
SELECT * FROM test_post_switchover;
```

3.6. Consideraciones Importantes y Limitaciones

3.6.1. Impacto del Switchover

Durante el proceso:

- Interrupción temporal del servicio (2-5 minutos típicamente)
- No hay pérdida de datos si se ejecuta correctamente
- Las conexiones existentes se desconectan
- Las aplicaciones deben reconectarse

Después del switchover:

- La antigua primary queda sin función (requiere reconfiguración manual)
 - Los puertos de acceso cambian (1522 en lugar de 1521 para acceso externo)
 - Se pierde la protección de Data Guard hasta reconfigurar nueva standby
-

3.6.2. Recomendaciones de Uso

Cuándo usar switchover:

- Mantenimiento planificado del servidor primary
- Testing de procedimientos de alta disponibilidad
- Migración de hardware
- Verificación de integridad de backups

Cuándo NO usar switchover:

- En producción sin planificación previa
- Si hay dudas sobre la sincronización de datos
- Sin backups recientes y verificados
- Durante horarios de alta demanda del sistema

3.6.3. Procedimiento de Rollback

IMPORTANTE: El switchover es irreversible, pero se puede crear una nueva configuración:

1. Detener la antigua primary
2. Reconfigurarla como nueva standby
3. Duplicar desde la nueva primary
4. Re-establecer la sincronización Data Guard

Este proceso requiere tiempo significativo y debe planificarse cuidadosamente.

Referencias bibliográficas

- [1] Oracle Corporation, *Oracle Data Guard Concepts and Administration*, Oracle Corporation, 2024, documentación oficial de Oracle Data Guard para Oracle Database 19c.
- [2] ——, “Configure a Standby Database,” <https://docs.oracle.com/en/solutions/configure-standby-db/index.html>, 2024, guía oficial de Oracle para la configuración de bases de datos standby.
- [3] Docker Inc, *Docker Compose Reference*, <https://docs.docker.com/compose/>, Docker Inc, 2024, documentación oficial de Docker Compose.
- [4] Oracle Corporation, “Oracle Container Registry,” <https://container-registry.oracle.com/>, 2024, registro oficial de contenedores Oracle para imágenes de base de datos.
- [5] P. K. Patra, “Creating Physical Standby Database from Prebuilt Database Image,” *LinkedIn*, 2024, artículo sobre implementación práctica de bases de datos standby físicas.
- [6] Oracle Corporation, “Oracle Container Registry Portal,” <https://container-registry.oracle.com/ords/f?p=113:10:::::>, 2024, portal de acceso al Oracle Container Registry.
- [7] kjarj54, “Vitalis Database Project Repository,” https://github.com/kjarj54/vitalis_db, 2025, repositorio GitHub del proyecto Vitalis con implementación de Oracle Data Guard Standby.
- [8] Oracle Corporation, *Oracle Database Administrator’s Guide*, Oracle Corporation, 2024, guía del administrador de Oracle Database 19c Enterprise Edition.