



Implementa Google Cloud NetApp Volumes con Oracle HA

NetApp database solutions

NetApp
June 25, 2026

Tabla de contenidos

Implementa Google Cloud NetApp Volumes con Oracle HA	1
Aprovisiona instancias de Google Compute Engine para Google Cloud NetApp Volumes	1
Paso 1: crear las máquinas virtuales	1
Paso 2: Configura el firewall de la VPC para TCP 1521	2
Paso 3: Configura el nombre de host, DNS y /etc/hosts	2
Paso 4: prepara el sistema operativo únicamente en los servidores de base de datos	3
Paso 5: Captura el nombre IQN del iniciador iSCSI	4
¿Qué sigue?	5
Aprovisiona almacenamiento iSCSI de Google Cloud NetApp Volumes para Oracle Database 26ai	5
Paso 1: Crear grupos iSCSI de GCNV	5
Paso 2: crea grupos de hosts	6
Paso 3: crear volúmenes iSCSI de GCNV	6
Paso 4: Configura iSCSI y multivía	7
Paso 5: particionar dispositivos ASM	9
Paso 6: formatear y montar /u01	10
¿Qué sigue?	11
Instala Oracle Grid Infrastructure y Oracle Database 26ai en Google Cloud NetApp Volumes	11
Paso 1: instala Grid Infrastructure en cada servidor de base de datos	11
Paso 2: instala Oracle Database en cada servidor de base de datos	14
¿Qué sigue?	16
Crea la base de datos principal de Oracle en Google Cloud NetApp Volumes	16
¿Qué sigue?	18
Crea la base de datos en espera de Oracle con la siembra en la capa de almacenamiento de Google Cloud NetApp Volumes	18
Paso 1: Configura los parámetros del listener y de Data Guard	18
Paso 2: Prepara el pfile de reserva y NOMOUNT	19
Paso 3: inicializa el almacenamiento en espera con GCNV	21
Paso 4: Registrar el standby con Oracle Restart	24
¿Qué sigue?	26
Finaliza la base de datos de reserva para Data Guard en Google Cloud NetApp Volumes	26
Paso 1: crea archivos de registro de recuperación en modo de espera	27
Paso 2: activa flashback e inicia la recuperación	27
Paso 3: Activar el envío de registro de recuperación	28
Paso 4: comprobar el estado de Data Guard	29
¿Qué sigue?	30
Configura Data Guard Broker y Fast-Start Failover para Oracle Database 26ai en Google Cloud NetApp Volumes	30
Paso 1: activar Data Guard Broker	31
Paso 2: Confirma el flashback para FSFO	32
Paso 3: configura y activa FSFO	32
Paso 4: instala Instant Client en Observer	33
Paso 5: Ejecuta Observer como servicio de systemd	34

Paso 6: Probar FSFO.....	37
¿Qué sigue?.....	38

Implementa Google Cloud NetApp Volumes con Oracle HA

Aprovisiona instancias de Google Compute Engine para Google Cloud NetApp Volumes

Provisiona máquinas virtuales de Google Compute Engine para alojar Oracle Database 26ai en el almacenamiento iSCSI de Google Cloud NetApp Volumes. Este procedimiento cubre la creación de los hosts de base de datos principal y de reserva y la máquina virtual Fast-Start Failover Observer, la configuración de las reglas del firewall de la VPC para Oracle Net, la configuración de la resolución de nombres de host, la preparación del sistema operativo y la captura de los nombres de los iniciadores iSCSI para el aprovisionamiento de almacenamiento GCNV.

Paso 1: crear las máquinas virtuales

Crea tres máquinas virtuales de Google Compute Engine en diferentes zonas de la misma región para aislar los fallos zonales. Utiliza la Cloud Console, `gcloud`, Terraform o tu flujo de trabajo de aprovisionamiento habitual.

1. Crea las tres máquinas virtuales con las especificaciones que se indican en la tabla siguiente.

Prefiere una región con menores emisiones de carbono para el coste total de propiedad (TCO) y la sostenibilidad, siempre que cumpla con las necesidades de latencia y cumplimiento (por ejemplo `us-west1` vs `us-central1`):

VM	Zona	Tipo de máquina	Disco de arranque	Red	Objetivo
oracdb1	us-west1-a	n4-highmem-8 (ejemplo) o c4-standard-*	OL 10, 50 GB Hyperdisk Balanced (solo para el sistema operativo)	oracle-vpc / oracle-subnet, gVNIC	Base de datos principal
oracdb2	us-west1-b	Igual que el principal	OL 10, 50 GB Hyperdisk Balanced (solo para el sistema operativo)	Lo mismo	Base de datos en espera
oradg-obs	us-west1-c	e2-medium	OL 10, 20 GB Hyperdisk equilibrado	Lo mismo	FSFO Observer (solo para Instant Client)

Utiliza el nivel de red Premium cuando la latencia o el tráfico de salida (>~200 GiB/mes) sean factores importantes; utiliza Standard para un menor TCO en desarrollo y pruebas.

2. Activa las funciones de la máquina virtual protegida y comprueba la configuración del disco de arranque:

Activa **Arranque seguro**, **vTPM** y **Supervisión de integridad** en las tres máquinas virtuales.

El disco de arranque solo contiene el sistema operativo. /u01, los directorios de Grid/DB, el almacenamiento temporal y todos los datos ASM usan volúmenes iSCSI de GCNV (ver [Provisiona volúmenes iSCSI de GCNV](#))

No conectes un disco de datos GCE independiente para /u01.

Paso 2: Configura el firewall de la VPC para TCP 1521

Crea reglas de firewall en la VPC para permitir TCP/1521 entre las tres máquinas virtuales para el transporte de red de Oracle Net y la conectividad de Observer. La falta de reglas interrumpe la replicación de Data Guard.

1. Crea una regla de entrada del firewall de la VPC para permitir TCP/1521 entre las tres IP internas de las máquinas virtuales. Usa reglas del firewall de la VPC o políticas de firewall con la misma allowlist:

Cloud Console: Red VPC → Cortafuegos → Crear regla `allow-oracle-net-dbhosts` en `oracle-vpc` — Ingreso, Permitir, fuentes = tres /32 IPs, TCP 1521. Refleja la salida si es necesario.

2. Comprueba la conectividad desde cada máquina virtual para verificar que las reglas del cortafuegos estén en su lugar:

```
sudo dnf install -y nmap-ncat

for tgt in <oracdb1-ip> <oracdb2-ip> <oradg-obs-ip>; do
  nc -zv -w 5 "$tgt" 22
  nc -zv -w 5 "$tgt" 1521
done
```

Puerto	Previsto	Significado
22	Conectado	La ruta SSH funciona
1521	Conexión rechazada	El cortafuegos está abierto; el listener de Grid se inicia durante Paso 1: instalar Oracle Grid Infrastructure (Oracle Restart) en cada servidor de base de datos
O bien	Tiempo de espera	Corregir el firewall o el enrutamiento

Ejecuta desde las tres máquinas virtuales hacia cada dirección IP de par.

Paso 3: Configura el nombre de host, DNS y /etc/hosts

Configura el nombre de host y la resolución DNS en las tres máquinas virtuales para que la resolución de nombres directa e inversa funcione para Oracle Net, el Data Guard Broker y el Observer.

1. Configura el nombre de host y añade /etc/hosts entradas en los tres hosts. Sustituye las direcciones IP internas de GCE (visibles en la lista **Compute Engine** → **VM instances**, columna *Internal IP*):

```
# Run on each VM, substituting the local short name (oracdb1, oracdb2,
oradg-obs)
sudo hostnamectl set-hostname <this-host>.example.internal

# Run on every VM (same content)
sudo tee -a /etc/hosts >/dev/null <<EOF

# Oracle DG peers + FSFO Observer
<oracdb1-ip>    oracdb1.example.internal    oracdb1
<oracdb2-ip>    oracdb2.example.internal    oracdb2
<oradg-obs-ip>  oradg-obs.example.internal    oradg-obs
EOF
```

2. Comprueba la resolución de nombres desde cada host:

```
ping -c 1 oracdb1 && ping -c 1 oracdb2 && ping -c 1 oradg-obs
```

Paso 4: prepara el sistema operativo únicamente en los servidores de base de datos

Prepara el sistema operativo en `oracdb1` y `oracdb2` para Oracle Database 26ai instalando el paquete de preinstalación, creando usuarios y grupos, instalando los paquetes iSCSI y multivía, y configurando el iniciador iSCSI. La configuración de Observer se explica en [Paso 4: instala Oracle Instant Client en el host Observer](#).



Requisito previo: Conexión HTTPS saliente a `yum.oracle.com` (Cloud NAT o servidor espejo interno en subredes privadas).

1. Instala el paquete de preinstalación de Oracle Database, crea el usuario `grid` y los grupos ASM, y añade el usuario `oracle` a los grupos ASM:

```
# Oracle 26ai preinstall (package name varies by repo)
sudo dnf install -y oracle-ai-database-preinstall-26ai \
  || sudo dnf install -y oracle-database-preinstall-26ai \
  || sudo dnf install -y oracle-database-preinstall-23ai

# grid user + asm groups
sudo groupadd -g 54327 asmadmin; sudo groupadd -g 54328 asmdba; sudo
groupadd -g 54329 asmoper
sudo useradd -u 54322 -g oinstall -G dba,oper,asmadmin,asmdba,asmoper
grid
sudo passwd -l grid; sudo passwd -l oracle
sudo usermod -a -G asmdba,asmadmin oracle
```

2. Instala los paquetes iSCSI, multivía y JDK, luego verifica el THP y la sincronización horaria:

```
sudo dnf install -y iscsi-initiator-utils device-mapper-multipath
sg3_utils \
    java-21-openjdk-headless libxcrypt-compat

# THP and time
cat /sys/kernel/mm/transparent_hugepage/enabled # expect [never]
timedatectl
chronyc tracking
```

3. Configura los ajustes de SELinux, el cortafuegos y el iniciador iSCSI, luego reinicia:



Configuración de seguridad (OL 10): Los siguientes comandos configuran SELinux en modo permisivo y desactivan `firewalld`. Esta es solo una configuración mínima para el laboratorio. Para una configuración reforzada de SELinux y del cortafuegos, consulta la línea base de seguridad de tu organización.

```
sudo setenforce 0
sudo sed -i 's/^SELINUX=.*SELINUX=permissive/' /etc/selinux/config
sudo systemctl disable --now firewalld

sudo cp -n /etc/iscsi/iscsid.conf /etc/iscsi/iscsid.conf.orig
sudo sed -i '/^[#[:space:]]*node\.session\.timeo\.replacement_timeout/d'
/etc/iscsi/iscsid.conf
echo "node.session.timeo.replacement_timeout = 120" | sudo tee -a
/etc/iscsi/iscsid.conf
sudo systemctl enable --now iscsid

sudo reboot
```

Paso 5: Captura el nombre IQN del iniciador iSCSI

Anota el nombre del iniciador iSCSI (IQN) de cada host de base de datos tras el reinicio. Utilizarás estos IQN para crear los grupos de hosts GCNV en [Paso 2: Crear los grupos de hosts](#).

1. Copia el IQN de `oracdb1` y anótalo:

```
sudo cat /etc/iscsi/initiatorname.iscsi
# InitiatorName=iqn.1994-05.com.redhat:abc123def456
```

2. Repite el proceso en `oracdb2` y anota su IQN. Utiliza un grupo de hosts por cada host, de modo que el reinicio de un host concreto o la regeneración de su IQN no puedan afectar a la visibilidad del volumen iSCSI GCNV de otro host:



Máquinas virtuales clonadas: Si ambos hosts comparten el mismo IQN, regénéralo en `oracdb2` (detén `iscsi`, borra `/var/lib/iscsi/nodes/*`, nuevo `InitiatorName` en `/etc/iscsi/initiatorname.iscsi`, reinicia `iscsid`).

¿Qué sigue?

Para proporcionar almacenamiento compartido para los binarios de Oracle y los grupos de discos ASM, ve a [Aprovisiona pools iSCSI, grupos de hosts y volúmenes de Google Cloud NetApp Volumes](#).

Aprovisiona almacenamiento iSCSI de Google Cloud NetApp Volumes para Oracle Database 26ai

Implementa almacenamiento en bloques iSCSI de Google Cloud NetApp Volumes para la alta disponibilidad de Oracle Database 26ai en Google Compute Engine. Este procedimiento cubre la creación de grupos de almacenamiento unificados GCNV Flex, la definición de grupos de hosts, la creación de volúmenes iSCSI para cada host de base de datos, la configuración de iSCSI y multivía en Linux, el particionado de los dispositivos de respaldo ASM y el montaje del `/u01` sistema de archivos.

Paso 1: Crear grupos iSCSI de GCNV

Crea dos grupos de almacenamiento Flex Unified, uno en cada zona de base de datos, para proporcionar volúmenes iSCSI para los hosts principal y de reserva. Cada host de base de datos utiliza volúmenes del grupo de su zona local.

1. Crea dos grupos de almacenamiento mediante la Consola en la nube. Utiliza las especificaciones de la tabla siguiente y repite el proceso de creación para cada zona:

Nombre del pool	Zona	Utilizado por
<code>oracle-pool-a</code>	<code>us-west1-a</code>	<code>oracdb1</code> (primaria)
<code>oracle-pool-b</code>	<code>us-west1-b</code>	<code>oracdb2</code> (en espera)

NetApp Volumes → **Storage pools** → **Create** para cada pool:

- **Nivel de servicio:** Flex (no Premium)
 - **Tipo:** Unificado
 - **Zona:** debe coincidir con la zona de la máquina virtual de la base de datos (`us-west1-a` / `us-west1-b`)
 - **Aviso de interés público:** conectado a `oracle-vpc`
 - **Capacidad:** dimensionada en función de la carga de trabajo; usa un rendimiento personalizado provisionado o IOPS cuando las operaciones de redo, copia de seguridad o restauración superen el margen predeterminado (hasta 5120 MiB/s o 160K IOPS por grupo, según los límites del producto)
2. Espera a que ambos grupos alcancen el estado `READY` antes de continuar. Adapta el tamaño de los grupos al espacio que ocupa tu base de datos (los tamaños que aparecen en [Paso 3: Crear los volúmenes iSCSI de GCNV](#) son ejemplos):



Modo predeterminado (esta guía): Los grupos unificados de Flex utilizan el modo predeterminado (`--mode=default`). Crea grupos y volúmenes iSCSI con Cloud Console o `gcloud netapp`. La replicación de volúmenes, las instantáneas y los clones utilizan las API de Google Cloud ([Paso 3: inicialización del modo de espera de GCNV](#)).

Paso 2: crea grupos de hosts

Crea un grupo de hosts por cada host de base de datos, de modo que cada máquina virtual solo vea sus propios volúmenes. Los hosts principal y de reserva no deben compartir volúmenes iSCSI de GCNV para mantener la independencia del almacenamiento.

1. Crea el grupo de hosts para `oracdb1` usando la Cloud Console:

NetApp Volúmenes → Grupos de hosts → Crear

- **Nombre:** `oracdb1-hg`
 - **Región:** `us-west1`
 - **Tipo:** iniciador iSCSI
 - **Tipo de sistema operativo:** Linux
 - **Servidores:** pega el IQN de `oracdb1` (el valor de `/etc/iscsi/initiatorname.iscsi`)
 - **Descripción:** «Oracle primary host `oracdb1`»
 - **Crear**
2. Repite el proceso para `oracdb2` con el nombre `oracdb2-hg` y el IQN de `oracdb2`. El host Observer no requiere recursos GCNV.

Paso 3: crear volúmenes iSCSI de GCNV

Crea cinco volúmenes iSCSI de GCNV para cada host de base de datos: uno para `/u01` y cuatro para los dispositivos de respaldo de ASM. Los volúmenes de cada host deben crearse en el grupo de almacenamiento de su zona local con su grupo de hosts correspondiente.

1. Crea los cinco volúmenes para `oracdb1` en `oracle-pool-a` con el grupo de hosts `oracdb1-hg`. Utiliza las especificaciones que figuran en la tabla siguiente:

Volumen iSCSI de GCNV	Size	Uso	Alias de multivía
<code>ora_<host>_u01</code>	100 GiB	<code>/u01</code> Volumen iSCSI de GCNV — Grid/Oracle homes, entorno de prueba	<code>/dev/mapper/ora_<host>_u01</code>
<code>ora_<host>_data_01</code>	50 GiB	ASM +DATA	<code>/dev/mapper/ora_<host>_data_01</code>
<code>ora_<host>_data_02</code>	50 GiB	ASM +DATA (a rayas)	<code>/dev/mapper/ora_<host>_data_02</code>
<code>ora_<host>_arch_01</code>	100 GiB	ASM +RECO	<code>/dev/mapper/ora_<host>_arch_01</code>

Volumen iSCSI de GCNV	Size	Uso	Alias de multivía
ora_<host>_fra_01	100 GiB	ASM +FRA	/dev/mapper/ora_<host>_fra_01

Nombres de volúmenes: solo letras, números y guiones bajos (sin guiones).



Configuración mínima (solo para validación): Dos LUN por host (*_data, *_reco) con arch_01p1→+RECO y arch_01p2→+FRA es aceptable para laboratorio; en producción se usan cinco volúmenes por [Paso 3: Crear los volúmenes iSCSI de GCNV](#).

2. Crea los cinco volúmenes para oracdb2 en oracle-pool-b con el grupo de hosts oracdb2-hg usando las mismas especificaciones. Para cada pool, usa **NetApp Volumes** → **Volumes** → **Create** — iSCSI, pool y grupo de hosts correctos, Linux. Anota la siguiente información:
 - Direcciones IP del portal iSCSI → <ISCSI_PORTAL_1>, <ISCSI_PORTAL_2> (portales del grupo principal en oracdb1; portales del grupo de reserva en oracdb2 — pueden variar)
 - Número de serie de volumen desde la Cloud Console: úsalo con el WWID detectado por el host en [Paso 4: Configura iSCSI y multivía en Linux para los volúmenes iSCSI de GCNV](#)

Paso 4: Configura iSCSI y multivía

Configura iSCSI y device-mapper-multivía en cada host de base de datos para acceder a los volúmenes GCNV a través de ambas direcciones IP del portal de almacenamiento. Ejecuta estos pasos en oracdb1 usando las direcciones IP del portal del pool primario y luego repítelos en oracdb2 usando las direcciones IP del portal del pool de reserva. Si la salida del host está restringida, permite TCP/3260 desde cada máquina virtual de base de datos hacia sus direcciones IP de portal iSCSI de GCNV (además de TCP/1521 entre máquinas virtuales desde [Paso 2: Cortafuegos de VPC — añadir el puerto TCP/1521 a la lista de permisos en las tres zonas](#)).

1. Descubre los objetivos, inicia sesión y mantén el inicio del nodo:

```
sudo iscsiadm --mode discovery --op update --type sendtargets --portal
<ISCSI_PORTAL_1>
sudo iscsiadm --mode discovery --op update --type sendtargets --portal
<ISCSI_PORTAL_2>
sudo iscsiadm --mode node --op update --name node.startup --value
automatic
sudo iscsiadm --mode node -l all
sudo systemctl enable --now iscsid iscsi multipathd
sudo iscsiadm --mode session          # expect 10 sessions (5 GCNV iSCSI
volumes × 2 portals)
sudo lsblk -o NAME,SIZE,WWN,VENDOR,MODEL
```

Después de reiniciar, vuelve a comprobar antes de iniciar Oracle:

```
sudo iscsiadm --mode session
sudo multipath -ll
```

2. Configura device-mapper-multipath con los valores predeterminados y las reglas de la lista negra:

```
sudo tee /etc/multipath.conf >/dev/null <<'EOF'
defaults {
    find_multipaths    yes
    user_friendly_names yes
}
blacklist {
    devnode "^(ram|raw|loop|fd|md|dm-|sr|scd|st)[0-9]*"
    devnode "^hd[a-z]"
    devnode "^cciss.*"
}
EOF

sudo systemctl enable --now multipathd
sudo multipath -ll
```

3. Agrega los alias de WWID detectados por el host a /etc/multipath.conf (no adivines — multipath.conf **no** expande variables de shell). Detecta los WWID:

```
sudo multipath -ll
for dev in /dev/sd*; do
    [ -b "$dev" ] || continue
    printf '%s: ' "$dev"
    sudo /usr/lib/udev/scsi_id --whitelisted --device="$dev" 2>/dev/null
    || true
    echo
done
```

Añade los alias concretos de ese host a /etc/multipath.conf, luego a `sudo systemctl restart multipathd`.

En `oracdb1`, añade:

```

multipaths {
    multipath { wwid <host-discovered-wwid-for-u01>      alias
ora_oracdb1_u01      }
    multipath { wwid <host-discovered-wwid-for-data-01>  alias
ora_oracdb1_data_01 }
    multipath { wwid <host-discovered-wwid-for-data-02>  alias
ora_oracdb1_data_02 }
    multipath { wwid <host-discovered-wwid-for-arch-01>  alias
ora_oracdb1_arch_01 }
    multipath { wwid <host-discovered-wwid-for-fra-01>   alias
ora_oracdb1_fra_01  }
}

```

En oracdb2, utiliza el mismo patrón con `ora_oracdb2_* alias`, luego:

```

sudo systemctl restart multipathd
ls -l /dev/mapper/ora_$(hostname -s)_*

```

Paso 5: particionar dispositivos ASM

Divide los cuatro dispositivos de soporte de ASM (excluyendo u01) con una partición GPT cada uno para el consumo de ASM, luego configura las reglas de udev para la propiedad de grid. Ejecuta estos pasos en cada host de base de datos.

1. Particiona los cuatro dispositivos de almacenamiento ASM con GPT y comprueba las particiones:

```

HOST=$(hostname -s)      # oracdb1 on the primary, oracdb2 on the
standby
for dev in /dev/mapper/ora_${HOST}_data_01 \
           /dev/mapper/ora_${HOST}_data_02 \
           /dev/mapper/ora_${HOST}_arch_01 \
           /dev/mapper/ora_${HOST}_fra_01; do
    sudo parted -s "$dev" mklabel gpt
    sudo parted -s "$dev" mkpart primary 0% 100%
done
sudo partprobe
sudo systemctl reload multipathd
ls /dev/mapper/ora_${HOST}_*p1      # expect 4 partitions

```

2. Configura las reglas de udev para asignar la propiedad de grid y aplicar los cambios:

```

HOST=$(hostname -s)
sudo tee /etc/udev/rules.d/99-oracle-asm.rules >/dev/null <<'EOF'
KERNEL=="dm-*", ENV{DM_UUID}=="part?-mpath-*",
ENV{DM_NAME}=="ora_oracdb*_*p?", \
    OWNER="grid", GROUP="asmadmin", MODE="0660"
EOF

sudo udevadm control --reload-rules
for part in /dev/mapper/ora_${HOST}_*p1; do
    dm=$(readlink -f "$part" | xargs basename)
    sudo udevadm trigger --action=change --name-match="/dev/${dm}"
done
sudo udevadm settle
ls -lL /dev/mapper/ora_${HOST}_*p1    # grid:asmadmin 0660

```

Paso 6: formatear y montar /u01

Formatea el volumen GCNV con ora_<host>_u01 XFS y móntalo de forma persistente usando el UUID en /etc/fstab. El sistema de archivos /u01 contiene Grid home, Oracle home y archivos de staging.

1. Formatea el dispositivo multivía con XFS y anota su UUID:

```

HOST=$(hostname -s)
U01_DEV=/dev/mapper/ora_${HOST}_u01
ls -l "$U01_DEV"

sudo mkfs.xfs -f "$U01_DEV"
U01_UUID=$(sudo blkid -s UUID -o value "$U01_DEV")

```

2. Añade la entrada de montaje basada en UUID a /etc/fstab y monta el sistema de archivos:

```

sudo mkdir -p /u01
echo "UUID=${U01_UUID} /u01 xfs defaults,_netdev,nofail,x-
systemd.requires=iscsi.service,x-systemd.requires=multipathd.service,x-
systemd.after=iscsi.service,x-systemd.after=multipathd.service 0 0" |
sudo tee -a /etc/fstab
sudo mount -a

```

3. Crea la estructura de directorios con la propiedad adecuada para el software de Grid y Oracle:

```
sudo mkdir -p /u01/app/oraInventory /u01/app/26ai/grid /u01/app/grid \  
/u01/app/oracle/product/26ai/db_1 /u01/stage  
sudo chown -R grid:oinstall /u01/app/oraInventory /u01/app/26ai  
/u01/app/grid  
sudo chown -R oracle:oinstall /u01/app/oracle /u01/stage  
sudo chmod -R 775 /u01/app /u01/stage
```

Reinicia una vez y confirma que se monten /u01 antes de [instalar el software de Oracle](#).

¿Qué sigue?

Para instalar los binarios de Oracle Grid Infrastructure y Database en los hosts preparados, ve a [Instala el software Oracle Grid Infrastructure y Oracle Database](#) en ambos hosts.

Instala Oracle Grid Infrastructure y Oracle Database 26ai en Google Cloud NetApp Volumes

Instala Oracle Grid Infrastructure con Oracle Restart y ASM en Google Cloud NetApp Volumes almacenamiento iSCSI para cada host de base de datos, luego instala el software Oracle Database 26ai. Este procedimiento incluye preparar Oracle GoldImages, ejecutar instalaciones silenciosas con archivos de respuesta, crear grupos de discos ASM en volúmenes GCNV y preparar tanto el host primario como el de reserva con el mismo software de Oracle antes de crear la base de datos.

Paso 1: instala Grid Infrastructure en cada servidor de base de datos

Instala Oracle Grid Infrastructure GoldImage en cada host de base de datos para habilitar Oracle Restart y ASM. Ambos hosts requieren su propio Grid home, instancia de ASM y grupos de discos; Data Guard replica los datos a través de Oracle Net, no a través de almacenamiento compartido. Completa todos los pasos en oracdb1 antes de repetirlos en oracdb2.

1. Almacena los binarios de Oracle GoldImages, Release Update y OPatch en /u01/stage:

```
sudo chown oracle:oinstall /u01/stage && sudo chmod 775 /u01/stage  
# Upload GoldImages, RU, OPatch to /u01/stage.
```

2. Descomprime el archivo Grid GoldImage en el Grid home de destino. El 26ai GoldImage se instala descomprimiéndolo directamente en el directorio de destino:

```

sudo -u grid bash -c '
cd /u01/app/26ai/grid
unzip -q /u01/stage/LINUX.X64_<RELEASE>_grid_home.zip
'
sudo chown -R grid:oinstall /u01/app/26ai/grid

```

Si la versión de Grid GoldImage es anterior a la RU de destino, aplica el parche a la carpeta de instalación de Grid durante la configuración usando el flujo `gridSetup.sh -applyRU`, o usa una GoldImage que incluya la RU. Mantén las carpetas de instalación de Grid y de la base de datos en el mismo nivel de parche previsto.

3. Crea el `gridSetup` archivo de respuesta `/tmp/grid.rsp` en cada host. Sustituye el nombre del host y utiliza contraseñas seguras:

```

HOST=$(hostname -s)

sudo -u grid bash -c "cat > /tmp/grid.rsp <<RSP
oracle.install.responseFileVersion=/oracle/install/rspfmt_crsinstall_res
ponse_schema_v23.0.0
INVENTORY_LOCATION=/u01/app/oraInventory
installOption=HA_CONFIG
ORACLE_BASE=/u01/app/grid
clusterUsage=GENERAL_PURPOSE
OSDBA=asmdba
OSOPER=asmoper
OSASM=asmadmin
storageOption=FLEX_ASM_STORAGE
sysasmPassword=WelcomeOracle1!
asmsnmpPassword=WelcomeOracle1!
diskGroupName=DATA
redundancy=EXTERNAL
auSize=4
diskString=/dev/mapper/ora_${HOST}_*p*
diskList=/dev/mapper/ora_${HOST}_data_01p1,/dev/mapper/ora_${HOST}_data_
02p1
managementOption=NONE
RSP"
sudo -u grid chmod 600 /tmp/grid.rsp

```

4. Ejecuta `gridSetup.sh` en modo silencioso para copiar los binarios y preparar la configuración. Espera `Successfully Setup Software with warning(s)`. y el código de salida 6 (advertencias) o 0:

```
sudo -u grid bash -c '  
export ORACLE_HOME=/u01/app/26ai/grid  
export ORACLE_BASE=/u01/app/grid  
cd /u01/app/26ai/grid  
./gridSetup.sh -silent -responseFile /tmp/grid.rsp -ignorePrereqFailure  
'
```

5. Ejecuta `oraInstRoot.sh` y `root.sh` como `root`. El script `root.sh` crea los envoltorios `crsctl`, `srvctl` y `asmcmd` y pone en marcha OHAS:

```
sudo /u01/app/oraInventory/oraInstRoot.sh  
sudo /u01/app/26ai/grid/root.sh
```

6. Ejecuta `gridSetup.sh -executeConfigTools` para ejecutar los asistentes de configuración (NETCA, ASMCA, CVU) contra [el archivo de respuesta](#). Esto crea la instancia ASM y el `+DATA` grupo de discos. Espera `Successfully Configured Software`. después de NETCA / ASMCA / CVU:

```
sudo -u grid bash -c '  
export ORACLE_HOME=/u01/app/26ai/grid  
export ORACLE_BASE=/u01/app/grid  
cd /u01/app/26ai/grid  
./gridSetup.sh -silent -executeConfigTools -responseFile /tmp/grid.rsp  
'
```

7. Crea los grupos de discos `+RECO` y `+FRA` usando `asmca`. La instalación de un solo paso solo crea `+DATA`:

```

HOST=$(hostname -s)

sudo -u grid bash -c "
export ORACLE_HOME=/u01/app/26ai/grid
export ORACLE_SID=+ASM

\${ORACLE_HOME}/bin/asmca -silent -createDiskGroup \
  -diskGroupName RECO \
  -disk /dev/mapper/ora_${HOST}_arch_01p1 \
  -redundancy EXTERNAL -au_size 4

\${ORACLE_HOME}/bin/asmca -silent -createDiskGroup \
  -diskGroupName FRA \
  -disk /dev/mapper/ora_${HOST}_fra_01p1 \
  -redundancy EXTERNAL -au_size 4
"

```

8. Comprueba el estado de los grupos de discos ASM y del recurso Oracle Restart:

```

sudo -u grid ORACLE_HOME=/u01/app/26ai/grid ORACLE_SID=+ASM \
  /u01/app/26ai/grid/bin/sqlplus -s / as sysasm <<'SQL'
SELECT name, total_mb, free_mb, state FROM v$asm_diskgroup ORDER BY
name;
SQL

sudo /u01/app/26ai/grid/bin/crsctl stat res -t
# Expected ONLINE: ora.DATA.dg, ora.RECO.dg, ora.FRA.dg,
ora.LISTENER.lsnr, ora.asm, ora.cssd, ora.evmd.

```

9. Repite los pasos anteriores en `oracdb2`. El patrón `HOST=$(hostname -s)` en [pasos 3 y 4](#) y [paso 7](#) selecciona automáticamente los dispositivos iSCSI GCNV de ese host.

Utiliza los mismos nombres de grupos de discos ASM — Data Guard replica a través de Oracle Net, no almacenamiento.

Paso 2: instala Oracle Database en cada servidor de base de datos

Instala el software home de Oracle Database 26ai en cada host de base de datos usando una instalación silenciosa, solo de software, con la última Release Update aplicada. Completa todos los pasos en `oracdb1` antes de repetirlos en `oracdb2`.

1. Descomprime el directorio Database home, el último OPatch y el parche RU en sus respectivos directorios. Consulta la documentación de Oracle para conocer la estructura de directorios de RU y la ruta `-applyRU`:

```

sudo su - oracle
cd /u01/app/oracle/product/26ai/db_1
unzip -q /u01/stage/LINUX.X64_<RELEASE>_db_home.zip
rm -rf OPatch
unzip -q /u01/stage/p6880880_<base>_Linux-x86-64.zip
# latest OPatch
unzip -q /u01/stage/p<RU_PATCH>_<base>_Linux-x86-64.zip -d /u01/stage
# latest 26ai RU

```

2. Crea el archivo de respuesta de instalación y ejecuta la instalación silenciosa (solo software) con la RU aplicada. En OL 8/9, omite `-applyOneOffs` de la línea `runInstaller`:

```

sudo -u oracle tee /u01/stage/dbinstall.rsp >/dev/null <<'EOF'
oracle.install.option=INSTALL_DB_SWONLY
UNIX_GROUP_NAME=oinstall
INVENTORY_LOCATION=/u01/app/oraInventory
ORACLE_HOME=/u01/app/oracle/product/26ai/db_1
ORACLE_BASE=/u01/app/oracle
oracle.install.db.InstallEdition=EE
oracle.install.db.OSDBA_GROUP=dba
oracle.install.db.OSOPER_GROUP=oper
oracle.install.db.OSBACKUPDBA_GROUP=backupdba
oracle.install.db.OSDGDBA_GROUP=dgdba
oracle.install.db.OSKMDBA_GROUP=kmdba
oracle.install.db.OSRACDBA_GROUP=racdba
oracle.install.db.rootconfig.executeRootScript=false
EOF

sudo -u oracle bash -c '
export CV_ASSUME_DISTID=OEL10      # OEL9 / OEL8.10 if cluify requires it
cd /u01/app/oracle/product/26ai/db_1
./runInstaller -applyRU /u01/stage/<RU_PATCH> \
  -applyOneOffs /u01/stage/39292021 \
  -silent -ignorePrereqFailure -responseFile /u01/stage/dbinstall.rsp
'

```

3. Ejecuta el script de root posterior a la instalación:

```

sudo /u01/app/oracle/product/26ai/db_1/root.sh

```

4. Configura el entorno de Oracle en cada host de base de datos. Usa `ORACLE_SID=orcl` en `oracdb1` y `ORACLE_SID=orcls` en `oracdb2`:

```
sudo -u oracle tee -a /home/oracle/.bash_profile >/dev/null <<'EOF'  
export ORACLE_HOME=/u01/app/oracle/product/26ai/db_1  
export ORACLE_SID=orcl # use 'orcls' on oracdb2  
export GRID_HOME=/u01/app/26ai/grid  
export PATH=$ORACLE_HOME/bin:$GRID_HOME/bin:$PATH  
export TNS_ADMIN=$ORACLE_HOME/network/admin  
EOF
```

La base de datos de espera se crea en [Crear la base de datos de reserva](#).

¿Qué sigue?

Para crear la instancia primaria de producción para tu implementación de alta disponibilidad, ve a [Crear la base de datos principal de Oracle](#) en oracdb1.

Crea la base de datos principal de Oracle en Google Cloud NetApp Volumes

Crea la base de datos principal de Oracle en el almacenamiento iSCSI de Google Cloud NetApp Volumes utilizando el Asistente de configuración de bases de datos de Oracle en modo silencioso. Este procedimiento cubre ejecutar `dbca` para crear la base de datos de contenedor y la base de datos conectable en grupos de discos ASM respaldados por GCNV, configurar los destinos de los registros de archivo y añadir un servicio de aplicación basado en roles para la conmutación por error transparente después de habilitar Data Guard.

Pasos

Crea la base de datos contenedora de Oracle y la base de datos conectable en oracdb1 usando `dbca` en modo silencioso, configura los destinos de los registros de archivo, verifica el registro de Oracle Restart y añade un servicio de aplicación basado en roles para la conmutación por error transparente de los clientes.

1. Ejecuta `dbca` en modo silencioso para crear el CDB y el PDB en los grupos de discos ASM:

```

sudo -u oracle bash -c '
export ORACLE_HOME=/u01/app/oracle/product/26ai/db_1
export PATH=$ORACLE_HOME/bin:$PATH

dbca -silent -createDatabase \
  -templateName General_Purpose.dbc \
  -gdbname orcl -sid orcl \
  -characterSet AL32UTF8 -nationalCharacterSet AL16UTF16 \
  -sysPassword "ChangeMe!1" -systemPassword "ChangeMe!1" \
  -emConfiguration NONE \
  -datafileDestination +DATA -storageType ASM \
  -recoveryAreaDestination +FRA -recoveryAreaSize 25000 \
  -enableArchive true -archiveLogMode AUTO \
  -memoryMgmtType AUTO_SGA -totalMemory 4096 \
  -databaseType MULTIPURPOSE \
  -createAsContainerDatabase true -numberOfPDBs 1 \
  -pdbName orclpdb -pdbAdminPassword "ChangeMe!1" \
  -ignorePreReqs
'

```

2. Dirige los archivelogs a +RECO y abre y guarda el estado de la pluggable database. El standby usa configuraciones de archivelog equivalentes en [Paso 2: Standby init.ora, pfile y NOMOUNT](#):

```

sudo -u oracle bash -c '
export ORACLE_HOME=/u01/app/oracle/product/26ai/db_1
export ORACLE_SID=orcl
$ORACLE_HOME/bin/sqlplus -s / as sysdba <<SQL
ALTER SYSTEM SET log_archive_dest_1='\"'LOCATION=+RECO
VALID_FOR=(ALL_LOGFILES,ALL_ROLES) DB_UNIQUE_NAME=orcl'\"' SCOPE=BOTH;
ALTER PLUGGABLE DATABASE ALL OPEN;
ALTER PLUGGABLE DATABASE ALL SAVE STATE;
EXIT
SQL
'

```

3. Comprueba que la base de datos se esté ejecutando en Oracle Restart:

```

sudo /u01/app/26ai/grid/bin/srvctl status database -d orcl
# Expected: Database is running

sudo -u oracle sqlplus -s / as sysdba <<<"SELECT name, open_mode,
log_mode FROM v\\$database;"
# Expected: ORCL, READ WRITE, ARCHIVELOG

```

4. Crea un servicio de aplicaciones basado en roles para que las aplicaciones se conecten a través de orclapp y la conmutación por error sea transparente cuando Data Guard esté habilitado:

```
sudo -u oracle bash -c '  
export GRID_HOME=/u01/app/26ai/grid  
export ORACLE_HOME=/u01/app/oracle/product/26ai/db_1  
export PATH=$ORACLE_HOME/bin:$GRID_HOME/bin:$PATH  
  
srvctl add service \  
-db orcl \  
-service orclapp \  
-pdb orclpdb \  
-role PRIMARY \  
-policy AUTOMATIC  
  
srvctl start service -db orcl -service orclapp  
srvctl status service -db orcl -service orclapp  
'
```

Después de habilitar Data Guard Broker, orclapp se ejecuta solo en el PRIMARY. Multiplexa los archivos de control entre los grupos de discos ASM y ajusta la memoria a la carga de trabajo.

¿Qué sigue?

Para establecer la protección en modo de espera y la preparación para la conmutación por error, ve a [Crear la base de datos de reserva de Oracle](#) en oracdb2.

Crea la base de datos en espera de Oracle con la siembra en la capa de almacenamiento de Google Cloud NetApp Volumes

Crea la base de datos física en espera de Oracle utilizando la replicación en la capa de almacenamiento de Google Cloud NetApp Volumes, instantáneas o clones para acelerar la inicialización de la base de datos en espera en comparación con los métodos RMAN tradicionales. Este procedimiento abarca la configuración del listener, la creación del pfile de la base de datos en espera, la inicialización de los volúmenes de espera con la replicación GCNV, la finalización de la instancia de Oracle y el registro de la base de datos en espera con Oracle Restart. Todos los niveles de HA completan estos pasos. Para **Prod HA (Data Guard + FSFO)**, continúa con [Finalización de Data Guard](#) antes de configurar [Data Guard Broker, Fast-Start Failover y el Observer](#).

Paso 1: Configura los parámetros del listener y de Data Guard

Configura el listener en ambos servidores de base de datos para que admita conexiones de Data Guard, incluido el servicio requerido para el broker. `_DGMGRL` Configura el archivo de contraseñas y los parámetros de

registro de archivos en la base de datos principal.

1. Configura el listener principal y comprueba el entorno en `oracdb1`:

```
sudo su - oracle
. ~/.bash_profile          # ORACLE_SID=orcl, ORACLE_HOME set
```

2. Configura el listener en espera en `oracdb2` para que incluya los servicios `orcls` y `orcls_DGMGRL`:

```
GRID_HOME=/u01/app/26ai/grid
sudo -u grid tee "$GRID_HOME/network/admin/listener.ora" >/dev/null <<
'EOF'
LISTENER =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP) (HOST = oracdb2.example.internal) (PORT =
1521)))

SID_LIST_LISTENER =
  (SID_LIST =
    (SID_DESC = (GLOBAL_DBNAME = orcls)          (ORACLE_HOME =
/u01/app/oracle/product/26ai/db_1) (SID_NAME = orcls))
    (SID_DESC = (GLOBAL_DBNAME = orcls_DGMGRL) (ORACLE_HOME =
/u01/app/oracle/product/26ai/db_1) (SID_NAME = orcls)))
EOF
```

3. Reinicia el listener mediante Oracle Restart en ambos hosts y verifica que el servicio `_DGMGRL` esté registrado:

```
sudo -u grid bash -c '
export GRID_HOME=/u01/app/26ai/grid
export ORACLE_HOME=$GRID_HOME
$GRID_HOME/bin/srvctl stop listener
$GRID_HOME/bin/srvctl start listener
$GRID_HOME/bin/lsnrctl status
'
```

`lsnrctl status` debe incluir `<SID>` y `<SID>_DGMGRL`.

Paso 2: Prepara el pfile de reserva y NOMOUNT

Prepara la instancia de base de datos de reserva copiando el archivo de contraseñas desde la principal, creando un archivo `init.ora` mínimo con los parámetros de Data Guard e iniciando la instancia en modo NOMOUNT.

1. Copia el archivo de contraseñas principal al host de reserva utilizando IAP y `gcloud compute scp`:

```
PRIMARY_ZONE=us-west1-a      # zone of oracdb1
STANDBY_ZONE=us-west1-b      # zone of oracdb2

gcloud compute scp \
  oracdb1:/u01/app/oracle/product/26ai/db_1/dbs/orapworcl ./orapworcl \
  --zone=$PRIMARY_ZONE --tunnel-through-iap

gcloud compute scp \
  ./orapworcl oracdb2:/u01/app/oracle/product/26ai/db_1/dbs/orapworcls \
  --zone=$STANDBY_ZONE --tunnel-through-iap
```

2. Consulta el valor del parámetro `compatible` de la base de datos principal:

```
# On oracdb1
sudo -u oracle sqlplus -s / as sysdba \
  <<<"SELECT value FROM v\${parameter} WHERE name='compatible';"
```

3. Crea el archivo `pfile` de reserva en `oracdb2`, configura la propiedad del archivo de contraseñas e inicia la instancia en modo `NOMOUNT`. Sustituye el valor de `compatible` del paso anterior por

`<COPY_FROM_PRIMARY>`:

```
sudo -u oracle mkdir -p /u01/app/oracle/admin/orcls/adump
sudo chown oracle:oinstall
/u01/app/oracle/product/26ai/db_1/dbs/orapworcls
sudo chmod 0600 /u01/app/oracle/product/26ai/db_1/dbs/orapworcls

sudo -u oracle tee /u01/app/oracle/product/26ai/db_1/dbs/initorcls.ora
>/dev/null <<'EOF'
*.db_name='orcl'
*.db_unique_name='orcls'
*.audit_file_dest='/u01/app/oracle/admin/orcls/adump'
*.diagnostic_dest='/u01/app/oracle'
*.compatible='<COPY_FROM_PRIMARY>'
*.sga_target=3072m
*.pga_aggregate_target=1024m
*.processes=320
*.remote_login_passwordfile='EXCLUSIVE'
*.standby_file_management='AUTO'
*.fal_server='orcl'
*.log_archive_config='DG_CONFIG=(orcl,orcls)'
*.log_archive_dest_1='LOCATION=+RECO VALID_FOR=(ALL_LOGFILES,ALL_ROLES)
DB_UNIQUE_NAME=orcls'
```

```

*.log_archive_dest_2='SERVICE=orcl AFFIRM SYNC
VALID_FOR=(ONLINE_LOGFILES,PRIMARY_ROLE) DB_UNIQUE_NAME=orcl'
*.log_archive_dest_state_2='DEFER'
*.log_archive_format='%t_%s_%r.arc'
*.dg_broker_start=TRUE
*.undo_tablespace='UNDOTBS1'
*.open_cursors=300
*.db_create_file_dest='+DATA'
*.db_create_online_log_dest_1='+DATA'
*.db_recovery_file_dest='+FRA'
*.db_recovery_file_dest_size=25000m
EOF

echo "orcls:/u01/app/oracle/product/26ai/db_1:N" | sudo tee -a
/etc/oratab

sudo -u oracle bash -c '
export ORACLE_HOME=/u01/app/oracle/product/26ai/db_1
export ORACLE_SID=orcls
sqlplus / as sysdba <<SQL
STARTUP NOMOUNT
PFIL=/u01/app/oracle/product/26ai/db_1/dbs/initorcls.ora;
EXIT
SQL
'

```

La instancia en espera se encuentra ahora en modo NOMOUNT sin archivos de datos hasta [Paso 3: inicializa el almacenamiento en espera con GCNV](#).

Paso 3: inicializa el almacenamiento en espera con GCNV

Inicia los volúmenes de reserva en `oracle-pool-b`, conéctalos a `oracdb2`, monta los grupos de discos ASM y finaliza la instancia de reserva en estado MOUNT.

Utiliza la replicación GCNV para la inicialización de entornos de producción y la inicialización mediante instantáneas para flujos de trabajo puntuales en laboratorio.

Elige la ruta de siembra

Elige el método de siembra en espera según tu entorno y tus requisitos de recuperación.

- **Recomendado para producción:** Usa la ruta de replicación en [Ruta de replicación: crear y sincronizar replicas](#) y [Ruta de replicación: conmutación y conexión de volúmenes en espera](#).
- **Alternativa para laboratorios:** Usa [Ruta alternativa: semilla a partir de una instantánea](#).

Todos los enlaces se reúnen en [Montar grupos de discos ASM en modo de espera](#) y [Finaliza la instancia en espera](#).

Comprueba los requisitos previos

Confirma los siguientes requisitos previos antes de inicializar los volúmenes de reserva.

- `gcloud netapp` con compatibilidad con la replicación de volúmenes.
- Dos grupos **Default-mode** en diferentes ubicaciones (`oracle-pool-a`, `oracle-pool-b`).
- Los volúmenes de origen del grupo primario se encuentran en `oracdb1-hg`; los volúmenes de destino se crean mediante replicación.
- Ejecuta la replicación desde Cloud Shell o desde una estación de trabajo, no desde DB VMs.
- En `oracdb2`, completa la configuración completa de iSCSI y ASM del host desde [Paso 4](#), [Paso 5](#) y [Paso 6](#).

```
export PROJECT=<your-gcp-project>
export LOC_A=us-west1-a
export LOC_B=us-west1-b
export DEST_POOL="projects/${PROJECT}/locations/${LOC_B
}/storagePools/oracle-pool-b"
```

- Crea un grupo de reserva si es necesario:

```
gcloud netapp storage-pools create oracle-pool-b \
  --project="${PROJECT}" --location="${LOC_B}" \
  --service-level=flex --type=unified --mode=default \
  --capacity=1024 --network=name=<your-vpc>
```

Crear y sincronizar replications

Crema relaciones de replicación desde los volúmenes primarios a los volúmenes de reserva y luego espera a que termine la sincronización inicial.

```
gcloud netapp volumes replications create repl-oracdb2-data \
  --project="${PROJECT}" --location="${LOC_A}" --volume=oracdb1_data \
  --replication-schedule=EVERY_10_MINUTES \
  --destination-volume-parameters="storage_pool=${DEST_POOL
},volume_id=oracdb2_data,share_name=oracdb2_data"

gcloud netapp volumes replications create repl-oracdb2-reco \
  --project="${PROJECT}" --location="${LOC_A}" --volume=oracdb1_reco \
  --replication-schedule=EVERY_10_MINUTES \
  --destination-volume-parameters="storage_pool=${DEST_POOL
},volume_id=oracdb2_reco,share_name=oracdb2_reco"
```

+

Espera hasta que `mirrorState` esté `MIRRORED` y se haya completado la sincronización inicial para cada replicación.

Realiza el corte y conecta los volúmenes de reserva

Pon en reposo el servidor principal, detén la replicación después de la última sincronización y conecta los volúmenes de destino al grupo de hosts en espera.

En el primario, suspende las escrituras y captura los metadatos de recuperación:

```
ALTER DATABASE BEGIN BACKUP;
SELECT CURRENT_SCN FROM V$DATABASE;
ALTER DATABASE CREATE STANDBY CONTROLFILE AS '/tmp/orcls_stby.ctl';
```

Deja que se complete un último ciclo de replicación y, a continuación, detén las replicaciones:

```
gcloud netapp volumes replications stop repl-oracdb2-data \
  --project="${PROJECT}" --location="${LOC_A}" --volume=oracdb1_data
--force

gcloud netapp volumes replications stop repl-oracdb2-reco \
  --project="${PROJECT}" --location="${LOC_A}" --volume=oracdb1_reco
--force
```

Conectar los volúmenes de destino a `oracdb2-hg` (los LUN replicados pueden conservar los nombres de origen):

```
HG=$(gcloud netapp host-groups describe oracdb2-hg --project="${PROJECT}" \
  --location=us-west1 --format='value(name)')

gcloud netapp volumes update oracdb2_data --project="${PROJECT}" \
  --location="${LOC_B}" \
  --block-devices="name=oracdb1_data_lun,host-groups=${HG},os-type=LINUX"
```

Copia el archivo de control de reserva en `oracdb2`, luego finaliza el modo de copia de seguridad en el servidor principal:

```
ALTER DATABASE END BACKUP;
```

Semilla a partir de una instantánea

Utiliza esta ruta para la inicialización puntual de laboratorio cuando no sea necesaria la replicación continua.

Para una instalación inicial de laboratorio, crea una instantánea de origen y crea volúmenes en espera a partir

de esa instantánea en `oracle-pool-b` (Cloud Console o API). Asocia los volúmenes creados a `oracdb2-hg` y luego continúa con [Montar grupos de discos ASM en modo de espera](#).

Montar grupos de discos ASM en modo de espera

En el servidor de reserva, identifica las rutas de almacenamiento conectadas y monta los grupos de discos ASM antes de la recuperación de bases de datos.

En `oracdb2`, inicia sesión en los portales iSCSI de `standby-pool` y vuelve a escanear los dispositivos multivía. Si los encabezados de los discos ASM coinciden con la nomenclatura principal en un flujo de trabajo de laboratorio, utiliza alias al estilo principal (por ejemplo, `ora_oracdb1_data_01`, `ora_oracdb1_arch_01`), configura `asm_diskstring='/dev/mapper/ora_oracdb1_*p*'` y confirma que la propiedad de la partición sea `grid:asmadmin`, luego monta los grupos de discos:

```
ALTER DISKGROUP DATA MOUNT FORCE;
ALTER DISKGROUP RECO MOUNT FORCE;
ALTER DISKGROUP FRA MOUNT FORCE;
```

Finaliza la instancia en espera

Restaura el archivo de control de espera, recupera hasta el SCN capturado, convierte a espera física e inicia la recuperación gestionada.

```
STARTUP NOMOUNT;
RESTORE STANDBY CONTROLFILE FROM '/tmp/orcls_stby.ctl';
ALTER DATABASE MOUNT;
RECOVER DATABASE UNTIL SCN <quiesce_scn>;
ALTER DATABASE CONVERT TO PHYSICAL STANDBY;
SHUTDOWN IMMEDIATE;
STARTUP MOUNT;
ALTER DATABASE RECOVER MANAGED STANDBY DATABASE DISCONNECT FROM SESSION;
```

En este momento, el servidor de reserva debería ser `PHYSICAL STANDBY` y `MOUNTED` con la recuperación gestionada ya iniciada.

Próximos pasos específicos para cada nivel:

- **Prod HA (sin Data Guard):** Ve directamente a [Paso 4: Registrar el standby con Oracle Restart](#).
- **Prod HA (Data Guard + FSFO):** Sigue en [Paso 4: Registrar el standby con Oracle Restart](#), luego ve a [Pasos finales de Data Guard](#).

Paso 4: Registrar el standby con Oracle Restart

Registra la base de datos de reserva en Oracle Restart para que, al reiniciar, se recuperen automáticamente los grupos de discos ASM, se monte la base de datos de reserva y se reinicie la recuperación gestionada. Añade también el servicio de aplicación a ambos recursos de base de datos.

1. Captura la ubicación del `spfile` de la base de datos en espera y regístrala con Oracle Restart en `oracdb2`.

Sustituye <STANDBY_SPFILE_PATH> de la consulta (a menudo bajo +DATA):

```
sudo -u oracle bash -c '  
export ORACLE_SID=orcls  
sqlplus -s / as sysdba <<< "SHOW PARAMETER spfile;"  
'  
  
sudo -u oracle bash -c '  
export GRID_HOME=/u01/app/26ai/grid  
export ORACLE_HOME=/u01/app/oracle/product/26ai/db_1  
export PATH=$ORACLE_HOME/bin:$GRID_HOME/bin:$PATH  
  
srvctl add database \  
-db orcls \  
-dbname orcl \  
-oraclehome /u01/app/oracle/product/26ai/db_1 \  
-spfile <STANDBY_SPFILE_PATH> \  
-pwfile /u01/app/oracle/product/26ai/db_1/dbs/orapworcls \  
-role PHYSICAL_STANDBY \  
-startoption MOUNT \  
-stopoption IMMEDIATE \  
-diskgroup DATA,RECO,FRA  
  
srvctl config database -db orcls  
srvctl status database -db orcls  
'
```

2. Comprueba y actualiza el recurso de base de datos principal en oracdb1 para que incluya todas las dependencias de los grupos de discos ASM:

```
sudo -u oracle bash -c '  
export GRID_HOME=/u01/app/26ai/grid  
export ORACLE_HOME=/u01/app/oracle/product/26ai/db_1  
export PATH=$ORACLE_HOME/bin:$GRID_HOME/bin:$PATH  
srvctl config database -db orcl  
srvctl modify database -db orcl -diskgroup DATA,RECO,FRA  
srvctl config database -db orcl  
'
```

3. Agrega el servicio de la aplicación al recurso de base de datos en espera (orcls en oracdb2). Utiliza role PRIMARY en ambos lados para que orclapp esté disponible tras la conmutación de sitios:

```

sudo -u oracle bash -c '
export GRID_HOME=/u01/app/26ai/grid
export ORACLE_HOME=/u01/app/oracle/product/26ai/db_1
export PATH=$ORACLE_HOME/bin:$GRID_HOME/bin:$PATH

srvctl add service \
  -db orcls \
  -service orclapp \
  -pdb orclpdb \
  -role PRIMARY \
  -policy AUTOMATIC

srvctl config service -db orcls -service orclapp
'
```

4. Comprueba el recurso de la base de datos de reserva en oracdb2:

```

sudo -u oracle bash -c '
export GRID_HOME=/u01/app/26ai/grid
export ORACLE_HOME=/u01/app/oracle/product/26ai/db_1
export PATH=$ORACLE_HOME/bin:$GRID_HOME/bin:$PATH
srvctl status database -db orcls
'
```

¿Qué sigue?

Específico de cada nivel:

- **Prod HA (sin Data Guard):** Para mantener un destino de recuperación basado en la replicación de almacenamiento, la inicialización del servidor en espera ha finalizado y la base de datos en espera se ha registrado en Oracle Restart como instancia de respaldo.
- **Prod HA (Data Guard + FSFO):** Para habilitar la conmutación de sitios gestionada por el broker y la conmutación por error de inicio rápido, continúa con [Finaliza la base de datos de reserva para Data Guard](#).

Finaliza la base de datos de reserva para Data Guard en Google Cloud NetApp Volumes

Finaliza la base de datos en espera para Oracle Data Guard en Google Cloud NetApp Volumes creando archivos de registro de recuperación, habilitando flashback database, activando el envío de registros de recuperación y verificando el estado de Data Guard.

Específico del nivel: Este procedimiento solo es necesario para el nivel **Prod HA (Data Guard + FSFO)**.

Paso 1: crea archivos de registro de recuperación en modo de espera

Crea archivos de registro de recuperación en modo de espera en ambos hosts de la base de datos para permitir Fast-Start Failover. El tamaño debe ser mayor o igual que el del registro de recuperación en línea primario más grande, y la cantidad debe ser igual a (grupos en línea por hilo) + 1. Después de la siembra de GCNV, elimina y vuelve a crear los registros de recuperación en modo de espera en el standby para corregir las rutas replicadas.

1. Crea archivos de registro de recuperación en modo de espera en la base de datos principal (orcl):

```
ALTER SYSTEM SET db_create_file_dest='+DATA' SCOPE=BOTH;
ALTER DATABASE ADD STANDBY LOGFILE THREAD 1 ('+DATA') SIZE 1024M;
-- repeat (online log groups + 1) times
```

2. Elimina y vuelve a crear los archivos de registro de recuperación en espera (orcls después de la inicialización de GCNV. Las rutas replicadas bajo +DATA/ORCL/... provocan ORA-19527 / ORA-16086 hasta que se reconstruyan:

```
ALTER DATABASE RECOVER MANAGED STANDBY DATABASE CANCEL;
ALTER SYSTEM SET standby_file_management=MANUAL SCOPE=BOTH;
-- DROP STANDBY LOGFILE GROUP for each group# in v$standby_log;
ALTER SYSTEM SET db_create_file_dest='+DATA' SCOPE=BOTH;
ALTER SYSTEM SET standby_file_management=AUTO SCOPE=BOTH;
ALTER DATABASE ADD STANDBY LOGFILE THREAD 1 ('+DATA') SIZE 1024M;
-- repeat (online groups + 1) times; one member per group
ALTER DATABASE RECOVER MANAGED STANDBY DATABASE USING CURRENT LOGFILE
DISCONNECT FROM SESSION;
```

Paso 2: activa flashback e inicia la recuperación

Habilita flashback database en el servidor en espera para permitir la restauración automática tras una conmutación por error y luego inicia la recuperación gestionada con aplicación en tiempo real. Flashback debe habilitarse antes de iniciar la recuperación gestionada porque no se puede habilitar mientras MRP está activo.

1. Apaga la base de datos en espera, reiniciala en modo MOUNT y activa la función flashback database en oracdb2:

```
# On oracdb2
sudo -u oracle bash -c '
. ~/.bash_profile
export ORACLE_SID=orcls
sqlplus / as sysdba <<SQL
SHUTDOWN IMMEDIATE;
STARTUP MOUNT;
ALTER SYSTEM SET db_flashback_retention_target=1440 SCOPE=BOTH;
ALTER DATABASE FLASHBACK ON;
EXIT
SQL'
```

2. Inicia la recuperación gestionada con aplicación en tiempo real:

```
sudo -u oracle bash -c '
. ~/.bash_profile
export ORACLE_SID=orcls
sqlplus / as sysdba <<SQL
ALTER DATABASE RECOVER MANAGED STANDBY DATABASE USING CURRENT LOGFILE
DISCONNECT FROM SESSION;
EXIT
SQL'
```

USING CURRENT LOGFILE permite la aplicación en tiempo real (el redo se aplica en cuanto llega a los SRL).

Paso 3: Activar el envío de registro de recuperación

Habilita el transporte de redo desde el primario al standby activando LOG_ARCHIVE_DEST_STATE_2, que se configuró deliberadamente como DEFER en [Paso 2](#) del procedimiento de inicialización del standby para suprimir errores de ORA-12154 durante la creación del standby.

1. Cambia LOG_ARCHIVE_DEST_STATE_2 por ENABLE y fuerza un cambio de registro para iniciar el trasvase de registros:

```
sudo -u oracle bash -c '
. ~/.bash_profile
sqlplus / as sysdba <<SQL
ALTER SYSTEM SET LOG_ARCHIVE_DEST_STATE_2=ENABLE SCOPE=BOTH;
ALTER SYSTEM SWITCH LOGFILE;
ALTER SYSTEM ARCHIVE LOG CURRENT;
EXIT
SQL'
```

2. Comprueba que el envío de redo funciona correctamente:

```
sudo -u oracle bash -c '  
. ~/.bash_profile  
sqlplus / as sysdba <<SQL  
SELECT dest_id, status, error FROM v\${archive_dest_status} WHERE dest_id  
IN (1,2);  
EXIT  
SQL'  
# Expected: dest_id=2, STATUS=VALID, ERROR null.
```

Si dest_2 muestra ORA-12154, reinicia el primario. Después de [Paso 1: Habilitar el broker en ambas bases de datos](#), gestiona el transporte vía DGMGRL.

Paso 4: comprobar el estado de Data Guard

Comprueba que la base de datos principal esté en modo READ WRITE y que la base de datos en espera esté montada con recuperación gestionada aplicando los redo logs.

1. Comprueba el rol de la base de datos principal y el modo de apertura en oracdb1:

```
sudo -u oracle sqlplus -s / as sysdba \  
<<<"SELECT database_role || ' | ' || open_mode FROM v\${database};"  
# Expected: PRIMARY | READ WRITE
```

2. Comprueba la función de la base de datos en espera, el modo de apertura y el estado de la recuperación gestionada en oracdb2:

```

gcloud compute ssh oracdb2 --tunnel-through-iap --zone=us-west1-b

sudo -u oracle bash <<'BASH'
. ~/.bash_profile
export ORACLE_SID=orcls

sqlplus -s / as sysdba <<'SQL'
SELECT database_role || ' | ' || open_mode
FROM v$database;

SELECT process, status, sequence#
FROM v$managed_standby
WHERE process IN ('MRP0','RFS');

EXIT
SQL
BASH

```

Se espera que estén disponibles en el standby: PHYSICAL STANDBY | MOUNTED; MRP0 con APPLYING_LOG.

3. Si el servidor en espera muestra MOUNTED pero apply no se está ejecutando, reinicia la recuperación gestionada en oracdb2:

```

sudo -u oracle bash -c '
. ~/.bash_profile
export ORACLE_SID=orcls
sqlplus / as sysdba <<SQL
ALTER DATABASE RECOVER MANAGED STANDBY DATABASE CANCEL;
ALTER DATABASE RECOVER MANAGED STANDBY DATABASE USING CURRENT LOGFILE
DISCONNECT FROM SESSION;
EXIT
SQL'

```

¿Qué sigue?

Para activar la gestión automática de roles y la protección contra fallos, continúa con [Configura Oracle Data Guard Broker, Fast-Start Failover y el Observer](#).

Configura Data Guard Broker y Fast-Start Failover para Oracle Database 26ai en Google Cloud NetApp Volumes

Configura Oracle Data Guard Broker y Fast-Start Failover con un Observer dedicado para habilitar las transiciones automáticas de roles para Oracle Database 26ai en Google

Cloud NetApp Volumes.

Específico del nivel: Este procedimiento solo se aplica al nivel **Prod HA (Data Guard + FSFO)**.

Este procedimiento describe cómo habilitar el broker en ambas bases de datos, crear la configuración de Data Guard, habilitar FSFO con el modo de protección MaxAvailability, instalar Oracle Instant Client en el host del Observer, iniciar el Observer como servicio de systemd con credenciales basadas en wallet y probar la conmutación de sitios y la conmutación por error. Tras `ENABLE CONFIGURATION`, gestiona el transporte y los roles a través de **DGMGRL** (no ad-hoc `LOG_ARCHIVE_DEST_* SQL`).

Paso 1: activar Data Guard Broker

Activa Data Guard Broker en ambos hosts de base de datos y crea la configuración del broker que vincula las bases de datos primaria y de reserva bajo una gestión unificada.

1. Configura `dg_broker_start=TRUE` en los hosts de la base de datos principal y de reserva:

```
sudo -u oracle bash -c '  
. ~/.bash_profile  
sqlplus / as sysdba <<SQL  
ALTER SYSTEM SET dg_broker_start=TRUE SCOPE=BOTH;  
EXIT  
SQL'
```

2. En el servidor primario, conéctate a DGMGRL con la autenticación del sistema operativo y crea la configuración del broker:



Solo en el host Observer, usa `dgmgrl /@orcl` después de que exista el wallet de inicio de sesión automático. No pongas contraseñas en la línea de comandos `dgmgrl`.

```
sudo -u oracle bash -c '  
export ORACLE_HOME=/u01/app/oracle/product/26ai/db_1  
export ORACLE_SID=orcl  
export PATH=$ORACLE_HOME/bin:$PATH  
dgmgrl /  
'
```

```
DGMGRL> CREATE CONFIGURATION 'orcl_dg' AS  
PRIMARY DATABASE IS 'orcl' CONNECT IDENTIFIER IS orcl;  
DGMGRL> ADD DATABASE 'orcls' AS CONNECT IDENTIFIER IS orcls;  
DGMGRL> ENABLE CONFIGURATION;  
DGMGRL> SHOW CONFIGURATION;  
-- Expect: Configuration Status: SUCCESS, both members SUCCESS.
```

3. Valida la configuración: corrige cualquier `WARNING` o valor que no sea `NULL ERROR` antes de [Paso 3](#):

Configura las propiedades de FSFO y actívalo:

```
DGMGRL> VALIDATE DATABASE 'orcls';  
DGMGRL> SHOW CONFIGURATION VERBOSE;
```

Paso 2: Confirma el flashback para FSFO

Confirma que la base de datos flashback esté habilitada en ambos hosts. Flashback es necesario para la auto-reinstalación de FSFO, lo que permite que el antiguo primario se reincorpore automáticamente a la configuración como secundario después de una conmutación por fallo.

1. Confirma que `flashback_on` es YES en ambos hosts de bases de datos:

```
sudo -u oracle bash -c '  
. ~/.bash_profile  
sqlplus -s / as sysdba <<<"SELECT flashback_on FROM v\${database};"  
'  
# Expected on both hosts: YES
```

2. Solo en el primario, si aún no se ha configurado la retención de flashback:

```
sudo -u oracle bash -c '  
. ~/.bash_profile  
export ORACLE_SID=orcl  
sqlplus / as sysdba <<SQL  
ALTER SYSTEM SET db_flashback_retention_target=1440 SCOPE=BOTH;  
EXIT  
SQL'
```

Paso 3: configura y activa FSFO

Configura el transporte SYNC redo, configura el modo de protección MaxAvailability, define los destinos FSFO en cada base de datos y habilita Fast-Start Failover.

1. Configura el modo de transporte redo en SYNC ambas bases de datos y eleva el modo de protección a MaxAvailability:

```
DGMGRL> EDIT DATABASE 'orcl' SET PROPERTY LogXptMode='SYNC';  
DGMGRL> EDIT DATABASE 'orcls' SET PROPERTY LogXptMode='SYNC';  
DGMGRL> EDIT CONFIGURATION SET PROTECTION MODE AS MaxAvailability;
```

2. Establece los destinos FSFO de modo que cada base de datos designe a la otra como su destino de conmutación por error, luego configura el umbral y el comportamiento de restablecimiento automático:

```

-- Each side names the other
DGMGRL> EDIT DATABASE 'orcl' SET PROPERTY FastStartFailoverTarget =
'orcls';
DGMGRL> EDIT DATABASE 'orcls' SET PROPERTY FastStartFailoverTarget =
'orcl';

-- 30 s is the default; lower for faster RTO but more sensitive to
network blips
DGMGRL> EDIT CONFIGURATION SET PROPERTY FastStartFailoverThreshold = 30;
DGMGRL> EDIT CONFIGURATION SET PROPERTY FastStartFailoverAutoReinststate =
TRUE;

```

3. Activa la conmutación por error de inicio rápido y confirma la configuración:

```

DGMGRL> ENABLE FAST_START FAILOVER;
DGMGRL> SHOW FAST_START FAILOVER;
-- Expected: Threshold 30 seconds, Target orcls, Observer not yet
registered.

```

Paso 4: instala Instant Client en Observer

Instala Oracle Instant Client en la máquina virtual dedicada de Observer (`oradg-obs`), crea un usuario dedicado del sistema operativo `oracle` y configura el entorno Oracle Net para que Observer pueda conectarse a ambos miembros de la base de datos en TCP/1521.

1. Instala los paquetes de Oracle Instant Client en el host Observer (`oradg-obs`):

```

# Use -el8 / -el9 if the Observer is on an older OL/RHEL release
sudo dnf install -y oracle-instantclient-release-el10
sudo dnf install -y oracle-instantclient-basic \
                 oracle-instantclient-sqlplus \
                 oracle-instantclient-tools

```

2. Crea un usuario dedicado de `oracle` OS que será el propietario del monedero y de la unidad `systemd`:

```

sudo useradd -u 54321 -m oracle
sudo passwd -l oracle

```

3. Configura el entorno de Oracle Net y crea `tnsnames.ora` con las entradas correspondientes a ambos servidores de base de datos:

```

sudo mkdir -p /etc/oracle/network/admin
sudo chown -R oracle:oracle /etc/oracle

sudo -u oracle tee /home/oracle/.bash_profile >/dev/null <<'EOF'
export ORACLE_HOME=/usr/lib/oracle/26/client64
export LD_LIBRARY_PATH=$ORACLE_HOME/lib
export PATH=$ORACLE_HOME/bin:$PATH
export TNS_ADMIN=/etc/oracle/network/admin
EOF

# tnsnames.ora – must reach both DB hosts on TCP/1521
sudo tee /etc/oracle/network/admin/tnsnames.ora >/dev/null <<'EOF'
orcl =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP) (HOST = oracdb1) (PORT = 1521))
      (CONNECT_DATA = (SERVER = DEDICATED) (SERVICE_NAME =
orcl)))
orcls =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP) (HOST = oracdb2) (PORT = 1521))
      (CONNECT_DATA = (SERVER = DEDICATED) (SERVICE_NAME =
orcls)))
EOF
sudo chown oracle:oracle /etc/oracle/network/admin/tnsnames.ora

```

Paso 5: Ejecuta Observer como servicio de systemd

Crea un monedero con inicio de sesión automático que contenga las credenciales de ambos miembros de la base de datos, luego configura e inicia el Observer como un servicio de systemd para que sobreviva a los reinicios y se vuelva a conectar automáticamente a la configuración.

Guarda las credenciales de una cuenta administrativa dedicada de Data Guard (por ejemplo, SYS_{SDG}) en el monedero en lugar de SYS. Las credenciales nunca deben aparecer en una línea de comandos de `dgmgrl`, donde son visibles para `ps` y `journalctl`; conéctate siempre usando `/@<tns_alias>` en el Observer.

1. Crea el monedero cifrado e introduce las credenciales de ambos miembros de la base de datos:

```

sudo -iu oracle bash <<'BASH'
mkdir -p $TNS_ADMIN/wallet
mkstore -wrl $TNS_ADMIN/wallet -create          # prompts for a wallet
password - store in your secrets manager
mkstore -wrl $TNS_ADMIN/wallet -createCredential orcl sys ChangeMe!1
mkstore -wrl $TNS_ADMIN/wallet -createCredential orcls sys ChangeMe!1
BASH

sudo tee /etc/oracle/network/admin/sqlnet.ora >/dev/null <<'EOF'
WALLET_LOCATION = (SOURCE = (METHOD = FILE) (METHOD_DATA = (DIRECTORY =
/etc/oracle/network/admin/wallet)))
SQLNET.WALLET_OVERRIDE = TRUE
EOF
sudo chown oracle:oracle /etc/oracle/network/admin/sqlnet.ora
sudo chmod -R 0700 /etc/oracle/network/admin/wallet

sudo -iu oracle ls -l /etc/oracle/network/admin/wallet
# Expected: cwallet.sso and ewallet.p12

sudo -iu oracle bash <<'BASH'
sqlplus -L "/@orcl as sysdba" <<'SQL'
SELECT database_role FROM v$database;
EXIT
SQL
BASH

sudo -iu oracle bash <<'BASH'
sqlplus -L "/@orcls as sysdba" <<'SQL'
SELECT database_role FROM v$database;
EXIT
SQL
BASH

sudo -iu oracle dgmg1 /@orcl 'SHOW CONFIGURATION;'
sudo -iu oracle dgmg1 /@orcls 'SHOW CONFIGURATION;'

```

2. Genera la cartera de inicio de sesión automático `cwallet.sso`) para que el servicio `systemd Observer` pueda iniciarse sin que se solicite una contraseña. Si `cwallet.sso` falta después de ejecutar `mkstore`, usa `orapki` del paquete de herramientas de Instant Client o de un home de base de datos para crearlo y luego vuelve a añadir las credenciales almacenadas:

```

sudo -iu oracle orapki wallet create \
  -wallet /etc/oracle/network/admin/wallet \
  -auto_login
sudo -iu oracle ls -l /etc/oracle/network/admin/wallet
# Expected: cwallet.sso and ewallet.p12

```

3. Crea la unidad de systemd, activa el servicio y comprueba que el Observer está conectado:

```

sudo tee /etc/systemd/system/dgmgml-observer.service >/dev/null <<'EOF'
[Unit]
Description=Oracle Data Guard Fast-Start Failover Observer
After=network-online.target
Wants=network-online.target

[Service]
Type=simple
User=oracle
Group=oracle
Environment=ORACLE_HOME=/usr/lib/oracle/26/client64
Environment=LD_LIBRARY_PATH=/usr/lib/oracle/26/client64/lib
Environment=TNS_ADMIN=/etc/oracle/network/admin
Environment=PATH=/usr/lib/oracle/26/client64/bin:/usr/bin:/bin
ExecStart=/usr/lib/oracle/26/client64/bin/dgmgml -silent /@orcl "START
OBSERVER FILE IS '/var/lib/oracle/dgmgml-observer.dat'"
Restart=always
RestartSec=5

[Install]
WantedBy=multi-user.target
EOF

```

```

sudo install -d -o oracle -g oracle -m 0755 /var/lib/oracle
sudo install -o oracle -g oracle -m 0640 /dev/null /var/log/dgmgml-
observer.log

sudo tee /etc/logrotate.d/dgmgml-observer >/dev/null <<'EOF'
/var/log/dgmgml-observer.log {
    weekly
    rotate 8
    compress delaycompress missingok notifempty
    create 0640 oracle oracle
    copytruncate
}
EOF

sudo systemctl daemon-reload && sudo systemctl enable --now dgmgml-
observer.service
sudo systemctl status dgmgml-observer.service

```

El observador debe leer **CONNECTED** del primario (un **DISCONNECTED** observador suspende el FSFO de forma silenciosa):

```

DGMGRL> SHOW FAST_START FAILOVER;
DGMGRL> SHOW CONFIGURATION;          -- Configuration Status: SUCCESS,
FSFO: ENABLED

```

Paso 6: Probar FSFO

Valida la configuración de Data Guard con `VALIDATE DATABASE`, luego realiza una conmutación de sitios planificada y, en una ventana de pruebas, una conmutación por error no planificada tras un reinicio de la máquina virtual para confirmar que FSFO funciona de principio a fin.

1. Prueba una conmutación de sitios planificada y restaura la topología original:

```

DGMGRL> VALIDATE DATABASE 'orcls';
DGMGRL> SWITCHOVER TO 'orcls';
DGMGRL> SHOW CONFIGURATION;
DGMGRL> SWITCHOVER TO 'orcl';          -- restore topology

```

2. Prueba una conmutación por error no planificada mediante el reinicio de una máquina virtual en un intervalo de tiempo de prueba controlado:

Utiliza la opción **Reset** de la máquina virtual (prueba de tipo crash); una **Stop** normal puede que no active FSFO. Tail `/var/log/dgmgml-observer.log` en `oradg-obs` para monitorear el progreso de la conmutación por error; restaura la topología cuando termines.

¿Qué sigue?

Ya está lista la configuración de Oracle Data Guard Broker, Fast-Start Failover y Observer para esta implementación.

Información de copyright

Copyright © 2026 NetApp, Inc. Todos los derechos reservados. Imprimido en EE. UU. No se puede reproducir este documento protegido por copyright ni parte del mismo de ninguna forma ni por ningún medio (gráfico, electrónico o mecánico, incluidas fotocopias, grabaciones o almacenamiento en un sistema de recuperación electrónico) sin la autorización previa y por escrito del propietario del copyright.

El software derivado del material de NetApp con copyright está sujeto a la siguiente licencia y exención de responsabilidad:

ESTE SOFTWARE LO PROPORCIONA NETAPP «TAL CUAL» Y SIN NINGUNA GARANTÍA EXPRESA O IMPLÍCITA, INCLUYENDO, SIN LIMITAR, LAS GARANTÍAS IMPLÍCITAS DE COMERCIALIZACIÓN O IDONEIDAD PARA UN FIN CONCRETO, CUYA RESPONSABILIDAD QUEDA EXIMIDA POR EL PRESENTE DOCUMENTO. EN NINGÚN CASO NETAPP SERÁ RESPONSABLE DE NINGÚN DAÑO DIRECTO, INDIRECTO, ESPECIAL, EJEMPLAR O RESULTANTE (INCLUYENDO, ENTRE OTROS, LA OBTENCIÓN DE BIENES O SERVICIOS SUSTITUTIVOS, PÉRDIDA DE USO, DE DATOS O DE BENEFICIOS, O INTERRUPTIÓN DE LA ACTIVIDAD EMPRESARIAL) CUALQUIERA SEA EL MODO EN EL QUE SE PRODUJERON Y LA TEORÍA DE RESPONSABILIDAD QUE SE APLIQUE, YA SEA EN CONTRATO, RESPONSABILIDAD OBJETIVA O AGRAVIO (INCLUIDA LA NEGLIGENCIA U OTRO TIPO), QUE SURJAN DE ALGÚN MODO DEL USO DE ESTE SOFTWARE, INCLUSO SI HUBIEREN SIDO ADVERTIDOS DE LA POSIBILIDAD DE TALES DAÑOS.

NetApp se reserva el derecho de modificar cualquiera de los productos aquí descritos en cualquier momento y sin aviso previo. NetApp no asume ningún tipo de responsabilidad que surja del uso de los productos aquí descritos, excepto aquello expresamente acordado por escrito por parte de NetApp. El uso o adquisición de este producto no lleva implícita ninguna licencia con derechos de patente, de marcas comerciales o cualquier otro derecho de propiedad intelectual de NetApp.

Es posible que el producto que se describe en este manual esté protegido por una o más patentes de EE. UU., patentes extranjeras o solicitudes pendientes.

LEYENDA DE DERECHOS LIMITADOS: el uso, la copia o la divulgación por parte del gobierno están sujetos a las restricciones establecidas en el subpárrafo (b)(3) de los derechos de datos técnicos y productos no comerciales de DFARS 252.227-7013 (FEB de 2014) y FAR 52.227-19 (DIC de 2007).

Los datos aquí contenidos pertenecen a un producto comercial o servicio comercial (como se define en FAR 2.101) y son propiedad de NetApp, Inc. Todos los datos técnicos y el software informático de NetApp que se proporcionan en este Acuerdo tienen una naturaleza comercial y se han desarrollado exclusivamente con fondos privados. El Gobierno de EE. UU. tiene una licencia limitada, irrevocable, no exclusiva, no transferible, no sublicenciable y de alcance mundial para utilizar los Datos en relación con el contrato del Gobierno de los Estados Unidos bajo el cual se proporcionaron los Datos. Excepto que aquí se disponga lo contrario, los Datos no se pueden utilizar, desvelar, reproducir, modificar, interpretar o mostrar sin la previa aprobación por escrito de NetApp, Inc. Los derechos de licencia del Gobierno de los Estados Unidos de América y su Departamento de Defensa se limitan a los derechos identificados en la cláusula 252.227-7015(b) de la sección DFARS (FEB de 2014).

Información de la marca comercial

NETAPP, el logotipo de NETAPP y las marcas que constan en <http://www.netapp.com/TM> son marcas comerciales de NetApp, Inc. El resto de nombres de empresa y de producto pueden ser marcas comerciales de sus respectivos propietarios.