Distribuisci Google Cloud NetApp Volumes con Oracle HA

NetApp database solutions

NetApp
June 25, 2026

Sommario

Distribuisci Google Cloud NetApp Volumes con Oracle HA

Effettua il provisioning delle istanze Google Compute Engine per Google Cloud NetApp Volumes

Effettua il provisioning di macchine virtuali Google Compute Engine per ospitare Oracle Database 26ai su storage iSCSI di Google Cloud NetApp Volumes. Questa procedura copre la creazione degli host di database primario e di standby e della VM Fast-Start Failover Observer, la configurazione delle regole del firewall VPC per Oracle Net, l'impostazione della risoluzione dei nomi host, la preparazione del sistema operativo e l'acquisizione dei nomi degli iSCSI initiator per il provisioning dello storage GCNV.

Passaggio 1: Creare le macchine virtuali

Crea tre macchine virtuali Google Compute Engine in zone diverse della stessa regione per l'isolamento dei guasti zonali. Usa la Cloud Console, `gcloud`, Terraform o il tuo flusso di lavoro di provisioning standard.

1. Crea le tre macchine virtuali con le specifiche indicate nella tabella sottostante.

   Preferisci una regione a basse emissioni di carbonio per il TCO e la sostenibilità, dove soddisfa le esigenze di latenza e conformità (ad esempio `us-west1` vs `us-central1`):

VM	Zona	tipo di macchina	Disco di avvio	Rete	Scopo
`oracdb1`	`us-west1-a`	`n4-highmem-8` (esempio) o `c4-standard-*`	OL 10, 50 GB Hyperdisk Bilanciato (solo sistema operativo)	`oracle-vpc` / `oracle-subnet`, gVNIC	Database primario
`oracdb2`	`us-west1-b`	Uguale al primario	OL 10, 50 GB Hyperdisk Bilanciato (solo sistema operativo)	Stesso	DB di riserva
`oradg-obs`	`us-west1-c`	`e2-medium`	OL 10, 20 GB Hyperdisk Bilanciato	Stesso	FSFO Observer (solo Instant Client)

   Usa il livello di rete Premium quando la latenza o il traffico in uscita (>~200 GiB/mese) sono importanti; usa il livello Standard per un TCO inferiore in dev/test.

2. Abilita le funzionalità di Shielded VM e verifica la configurazione del disco di avvio:

   Abilita **Secure Boot**, **vTPM** e **Integrity Monitoring** su tutte e tre le macchine virtuali.

   Il disco di avvio contiene solo il sistema operativo. `/u01` Le directory di Grid/DB, di staging e tutti i dati

ASM usano i volumi iSCSI GCNV (vedi Provisioning dei volumi iSCSI GCNV)

Non collegare un disco dati GCE separato per `/u01`.

Passaggio 2: Configura il firewall VPC per TCP 1521

Crea regole firewall VPC per consentire TCP/1521 tra tutte e tre le VM per il trasporto dei redo log di Oracle Net e la connettività di Observer. La mancanza di regole interrompe la replica di Data Guard.

1. Crea una regola di ingresso del firewall VPC per consentire TCP/1521 tra tutti e tre gli indirizzi IP interni delle VM. Usa le regole del firewall VPC o le Firewall Policies con la stessa allowlist:

   Cloud Console: Rete VPC → Firewall → Crea regola `allow-oracle-net-dbhosts` su `oracle-vpc` — Ingresso, Consenti, origini = tre `/32` IP, TCP 1521. Replica l'uscita se necessario.

2. Verifica la connettività da ogni macchina virtuale per assicurarti che le regole del firewall siano attive:

```
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
```

Porta	Previsto	Significato
`22`	Collegato	Il percorso SSH funziona
`1521`	Connessione rifiutata	Firewall aperto; il listener della griglia si avvia durante Passaggio 1: Installare Oracle Grid Infrastructure (Oracle Restart) su ciascun host del database
O	Timeout	Risolvi il problema del firewall o del routing

Eseguire da tutte e tre le macchine virtuali verso ciascun indirizzo IP del peer.

Passaggio 3: Configura hostname, DNS e `/etc/hosts`

Configura il nome host e la risoluzione DNS su tutte e tre le macchine virtuali in modo che la risoluzione diretta e inversa dei nomi funzioni per Oracle Net, il Data Guard Broker e l'Observer.

1. Imposta il nome host e aggiungi `/etc/hosts` le voci su tutti e tre gli host. Sostituisci gli indirizzi IP interni di GCE (visibili nell'elenco **Compute Engine** → **VM instances**, colonna *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. Convalida la risoluzione dei nomi da ciascun host:

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

Passaggio 4: Prepara il sistema operativo solo sugli host del database

Prepara il sistema operativo su `oracdb1` e `oracdb2` per Oracle Database 26ai installando il pacchetto di preinstallazione, creando utenti e gruppi, installando i pacchetti iSCSI e multipath e configurando l'iSCSI initiator. La configurazione dell'observer è trattata in Passaggio 4: Installare Oracle Instant Client sull'host Observer.

Prerequisito: HTTPS in uscita verso `yum.oracle.com` (Cloud NAT o mirror interno su sottoreti private).

1. Installa il pacchetto di preinstallazione di Oracle Database, crea l' `grid`utente e i gruppi ASM, e aggiungi l' `oracle`utente ai gruppi 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. Installa i pacchetti iSCSI, multipath e JDK, quindi verifica THP e la sincronizzazione dell'ora:

```
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 le impostazioni di SELinux, del firewall e di iSCSI initiator, quindi riavvia:

> **Security posture (OL 10):** I comandi seguenti impostano SELinux su permissivo e disabilitano `firewalld`. Questa è solo una configurazione di laboratorio minima. Per una configurazione SELinux e firewall più sicura, consultare le linee guida di sicurezza della propria organizzazione.

```
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
```

Passaggio 5: Acquisisci il nome IQN dell'iSCSI initiator

Acquisisci il nome dell'iSCSI initiator (IQN) da ciascun host del database dopo il riavvio. Utilizzerai questi IQN per creare i gruppi host GCNV in Passaggio 2: Creare i gruppi host.

1. Acquisisci l'IQN da `oracdb1` e annotalo:

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

2. Ripeti su `oracdb2` e registra il suo IQN. Usa un gruppo host per ogni host così il riavvio o la rigenerazione dell'IQN di un singolo host non possono influire sulla visibilità del volume iSCSI GCNV di un altro host:

VM clonate: Se entrambi gli host condividono lo stesso IQN, rigenerare su `oracdb2` (arresta `iscsi`, cancella `/var/lib/iscsi/nodes/*`, nuovo `InitiatorName` in `/etc/iscsi/initiatorname.iscsi`, riavvia `iscsid`).

E ora?

Per fornire storage condiviso per i file binari Oracle e i gruppi di dischi ASM, vai su Effettua il provisioning di pool iSCSI, gruppi host e volumi Google Cloud NetApp Volumes.

Esegui il provisioning dello storage iSCSI Google Cloud NetApp Volumes per database Oracle 26ai

Esegui il provisioning dello storage a blocchi iSCSI Google Cloud NetApp Volumes per l'alta disponibilità di Oracle Database 26ai su Google Compute Engine. Questa procedura copre la creazione di pool di storage unificato GCNV Flex, la definizione di gruppi di host, la creazione di volumi iSCSI per ciascun database host, la configurazione di iSCSI e multipath su Linux, il partizionamento dei dispositivi di supporto ASM e il montaggio del `/u01` filesystem.

Passaggio 1: Crea pool iSCSI GCNV

Crea due pool di storage Flex Unified, uno in ciascuna zona del database, per fornire volumi iSCSI agli host primario e di standby. Ogni host del database utilizza volumi dal pool della propria zona locale.

1. Crea due pool di archiviazione utilizzando la Cloud Console. Utilizza le specifiche riportate nella tabella seguente e ripeti il processo di creazione per ciascuna zona:

Nome pool	Zona	Utilizzato da
`oracle-pool-a`	`us-west1-a`	`oracdb1` (primario)
`oracle-pool-b`	`us-west1-b`	`oracdb2` (standby)

   NetApp Volumi → **Pool di storage** → **Crea** per ogni pool:

   - **Livello di servizio:** Flex (non Premium)
   - **Tipo:** Unificato
   - **Zona:** corrisponde alla zona VM del database (`us-west1-a`/ `us-west1-b`)
   - **PSA:** collegato a `oracle-vpc`
   - **Capacità:** dimensionata per il carico di lavoro; utilizzare throughput/IOPS personalizzati quando redo, backup o restore superano il margine predefinito (fino a 5120 MiB/s o 160K IOPS per pool, in base ai limiti del prodotto)

2. Attendi che entrambi i pool raggiungano `READY` status prima di procedere. Dimensiona le dimensioni dei pool in base al tuo database (le dimensioni in Passaggio 3: Creare i volumi iSCSI GCNV sono esempi):

Modalità predefinita (questa guida): I pool Flex Unified utilizzano la modalità predefinita (`--mode=default`). Crea pool e volumi iSCSI con Cloud Console o `gcloud netapp`. La replica dei volumi, gli snapshot e i cloni utilizzano le API di Google Cloud (Passaggio 3: Inizializzazione dello standby GCNV).

Passaggio 2: Crea gruppi host

Crea un gruppo host per ogni host del database, così ogni VM vede solo i propri volumi. Gli host primario e di standby non devono condividere i volumi iSCSI GCNV per mantenere uno storage indipendente.

1. Crea il gruppo host per `oracdb1` usando la Cloud Console:

   NetApp Volumi → Gruppi host → Crea

   - **Nome:** `oracdb1-hg`
   - **Regione:** `us-west1`
   - **Tipo:** iSCSI initiator
   - **Tipo di OS:** Linux
   - **Host:** incolla l'IQN da `oracdb1` (il valore di `/etc/iscsi/initiatorname.iscsi`)
   - **Descrizione:** "Host primario Oracle oracdb1"
   - **Crea**

2. Ripeti il processo per `oracdb2` con il nome `oracdb2-hg` e l'IQN di `oracdb2`. L'host Observer non richiede risorse GCNV.

Passaggio 3: Crea volumi iSCSI GCNV

Crea cinque volumi iSCSI GCNV per ogni host del database: uno per `/u01` e quattro per i dispositivi di supporto ASM. I volumi di ciascun host devono essere creati nel pool di storage della zona locale con il relativo gruppo di host.

1. Crea i cinque volumi per `oracdb1` in `oracle-pool-a` con host group `oracdb1-hg`. Usa le specifiche nella tabella qui sotto:

Volume iSCSI GCNV	Dimensione	Usa	Alias multipath
`ora_<host>_u01`	100 GiB	`/u01` Volume iSCSI GCNV — Grid/Oracle homes, staging	`/dev/mapper/ora_<host>_u01`
`ora_<host>_data_01`	50 GiB	ASM `+DATA`	`/dev/mapper/ora_<host>_data_01`
`ora_<host>_data_02`	50 GiB	ASM `+DATA` (a strisce)	`/dev/mapper/ora_<host>_data_02`
`ora_<host>_arch_01`	100 GiB	ASM `+RECO`	`/dev/mapper/ora_<host>_arch_01`
`ora_<host>_fra_01`	100 GiB	ASM `+FRA`	`/dev/mapper/ora_<host>_fra_01`

Nomi dei volumi: solo lettere, numeri e underscore (senza trattini).

> **Layout minimo (solo convalida):** Due LUN per host (`*_data`, `*_reco`) con `arch_01p1` →`+RECO` e `arch_01p2`→`+FRA` è accettabile per il laboratorio; la produzione utilizza cinque volumi per Passaggio 3: Creare i volumi iSCSI GCNV.

2. Crea i cinque volumi per `oracdb2` in `oracle-pool-b` con il gruppo host `oracdb2-hg` usando le stesse specifiche. Per ogni pool, usa **NetApp Volumes** → **Volumes** → **Create** — iSCSI, pool e gruppo host corretti, Linux. Annota le seguenti informazioni:
   - Indirizzi IP del portale iSCSI → `<ISCSI_PORTAL_1>`, `<ISCSI_PORTAL_2>` (portali del pool primario su `oracdb1`; portali del pool di standby su `oracdb2` — potrebbero essere diversi)
   - Volume seriale dalla Cloud Console: utilizzare con WWID rilevato dall'host in Passaggio 4: Configurare Linux iSCSI e multipath per i volumi iSCSI GCNV

Passaggio 4: Configura iSCSI e multipath

Configura iSCSI e device-mapper-multipath su ciascun host del database per accedere ai volumi GCNV tramite entrambi gli IP del portale di storage. Esegui questi passaggi su `oracdb1` utilizzando gli IP del portale del pool primario, poi ripetili su `oracdb2` utilizzando gli IP del portale del pool di standby. Se l'egress dell'host è limitato, consenti TCP/3260 da ogni VM del database verso i suoi IP del portale iSCSI GCNV (oltre a TCP/1521 tra VM Passaggio 2: Firewall VPC — consenti la porta TCP/1521 in tutte e tre le zone).

1. Scopri i target, accedi e mantieni l'avvio 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
```

   Dopo il riavvio, ricontrollare prima di avviare Oracle:

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

2. Configura `device-mapper-multipath` con le impostazioni predefinite e le regole della blacklist:

```
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. Aggiungi gli alias WWID rilevati dall'host `/etc/multipath.conf` (non fare supposizioni — `multipath.conf` **non** espande le variabili della shell). Rileva i 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
```

Aggiungi gli alias concreti per quell'host a `/etc/multipath.conf`, quindi `sudo systemctl restart multipathd`.

Su `oracdb1`, aggiungi:

```
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  }
}
```

Su `oracdb2`, usa lo stesso schema con `ora_oracdb2_*` alias, poi:

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

Passaggio 5: Partiziona i dispositivi ASM

Partiziona i quattro dispositivi di supporto ASM (escluso `u01`) con una partizione GPT ciascuno per l'utilizzo da parte di ASM, poi configura le regole udev per la proprietà di grid. Esegui questi passaggi su ciascun database host.

1. Partiziona i quattro dispositivi di supporto ASM con GPT e verifica le partizioni:

```
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 le regole udev per assegnare la proprietà grid e attivare le modifiche:

```
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
```

Passaggio 6: Formatta e monta `/u01`

Formatta il `ora_<host>_u01`volume GCNV con XFS e montalo in modo permanente utilizzando l'UUID in `/etc/fstab`. Il `/u01`filesystem contiene Grid home, Oracle home e file di staging.

1. Formatta il dispositivo multipath con XFS e acquisisci il suo 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. Aggiungi la voce di montaggio basata su UUID `/etc/fstab` e monta il filesystem:

```
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 struttura di directory con i permessi di proprietà corretti per i software Grid e 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
```

Riavvia una volta e conferma `/u01` i mount prima di installazione del software Oracle.

E ora?

Per installare i binari di Oracle Grid Infrastructure e Database sugli host preparati, vai su Installa il software Oracle Grid Infrastructure e Oracle Database su entrambi gli host.

Installa Oracle Grid Infrastructure e Oracle Database 26ai su Google Cloud NetApp Volumes

Installa Oracle Grid Infrastructure con Oracle Restart e ASM su storage iSCSI Google Cloud NetApp Volumes per ciascun host del database, poi installa il software Oracle Database 26ai. Questa procedura include la preparazione di Oracle GoldImages, l'esecuzione di installazioni silenziose con file di risposta, la creazione di gruppi di dischi ASM su volumi GCNV e la preparazione sia degli host primari che di standby con lo stesso software Oracle prima della creazione del database.

Passaggio 1: Installa Grid Infrastructure su ogni host DB

Installa Oracle Grid Infrastructure GoldImage su ogni database host per abilitare Oracle Restart e ASM. Entrambi gli host richiedono la propria Grid home, istanza ASM e gruppi di dischi; Data Guard replica i dati tramite Oracle Net, non tramite storage condiviso. Completa tutti i passaggi su `oracdb1` prima di ripeterli su `oracdb2`.

1. Prepara i binari di Oracle GoldImages, Release Update e OPatch in `/u01/stage`:

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

2. Decomprimi il Grid GoldImage nella posizione della Grid home di destinazione. Il GoldImage 26ai si installa decomprimendo direttamente nella directory di destinazione:

```
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
```

Se la Grid GoldImage è più vecchia dell'RU di destinazione, applica la patch alla Grid home durante la configurazione usando il `gridSetup.sh -applyRU` flow, oppure usa un GoldImage con l'RU incluso. Mantieni Grid e Database home allo stesso livello di patch previsto.

3. Crea il `gridSetup` file di risposta `/tmp/grid.rsp` su ciascun host. Sostituisci il nome host e utilizza password complesse:

```
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. Esegui `gridSetup.sh`in modalità silenziosa per copiare i file binari e predisporre la configurazione. Aspettati `Successfully Setup Software with warning(s).`i codici di uscita 6 (avvisi) 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. Esegui `orainstRoot.sh` e `root.sh` come root. Lo `root.sh` script crea i wrapper `crsctl`, `srvctl` e `asmcmd` e avvia OHAS:

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

6. Esegui `gridSetup.sh -executeConfigTools` per avviare gli assistenti di configurazione (NETCA, ASMCA, CVU) su il file di risposta. Questo crea l'istanza ASM e il `+DATA` gruppo di dischi. Attendi `Successfully Configured Software.` dopo 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 i gruppi di dischi `+RECO` e `+FRA` utilizzando `asmca`. L'installazione singola crea solo `+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. Verifica i gruppi di dischi ASM e lo stato delle risorse 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. Ripeti i passaggi precedenti su `oracdb2`. Il `HOST=$(hostname -s)` pattern in passaggi 3 e 4 e passo 7 seleziona automaticamente i dispositivi iSCSI GCNV di quell'host.

   Utilizza gli stessi nomi dei gruppi di dischi ASM: Data Guard replica tramite Oracle Net, non tramite storage.

Passaggio 2: Installa il database Oracle su ogni host DB

Installa la home del software Oracle Database 26ai su ogni host del database usando un'installazione silenziosa, solo software, con l'ultimo Release Update applicato. Completa tutti i passaggi su `oracdb1` prima di ripetere su `oracdb2`.

1. Decomprimi la home del database, l'ultima versione di OPatch e la patch RU nelle rispettive directory. Consulta la documentazione Oracle per la struttura delle directory RU e il percorso `-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. Scrivi il file di risposta dell'installazione ed esegui l'installazione silenziosa del solo software con l'RU applicato. Su OL 8/9, ometti `-applyOneOffs` dalla riga `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 cluvfy 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. Esegui lo script di root post-installazione:

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

4. Imposta l'ambiente Oracle su ogni host DB. Usa `ORACLE_SID=orcl` su `oracdb1` e `ORACLE_SID=orcls` su `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
```

Il database standby viene creato in Creare il database standby.

E ora?

Per creare l'istanza primaria di produzione per la tua distribuzione HA, vai a Crea il database Oracle primario su `oracdb1`.

Crea il database Oracle primario su Google Cloud NetApp Volumes

Crea il database Oracle primario su Google Cloud NetApp Volumes storage iSCSI usando Oracle Database Configuration Assistant in modalità silenziosa. Questa procedura copre l'esecuzione di `dbca` per creare il container database e il pluggable database su gruppi di dischi ASM supportati da GCNV, configurare le destinazioni dei log di archivio e aggiungere un servizio dell'applicazione in base al ruolo per il failover trasparente dopo che Data Guard è stato abilitato.

Passaggi

Crea il container database Oracle e il pluggable database su `oracdb1` usando `dbca` in modalità silenziosa, configura le destinazioni dei log di archivio, verifica la registrazione di Oracle Restart e aggiungi un servizio dell'applicazione in base al ruolo per il failover trasparente del client.

1. Esegui `dbca` in modalità silenziosa per creare il CDB e il PDB sui gruppi di dischi 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. Punta i file di archivelog a `+RECO` e apri e salva lo stato del pluggable database. Lo standby utilizza impostazioni di archivelog corrispondenti in Passaggio 2: Standby init.ora, pfile e 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. Verifica che il database sia in esecuzione con 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 servizio dell'applicazione in base al ruolo così le applicazioni si connettono tramite `orclapp` e il failover è trasparente quando Data Guard è abilitato:

```
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
'
```

Dopo l'attivazione di Data Guard Broker, `orclapp` viene eseguito solo sul PRIMARY. Multiplexa i file di controllo tra i gruppi di dischi ASM e dimensiona la memoria in base al carico di lavoro.

E ora?

Per configurare la protezione in standby e la predisposizione al failover, vai su Crea il database Oracle standby su `oracdb2`.

Crea il database Oracle standby con il seeding a livello di storage di Google Cloud NetApp Volumes

Crea il database Oracle physical standby utilizzando la replica a livello di storage, snapshot o cloni di Google Cloud NetApp Volumes per accelerare l'inizializzazione dello standby rispetto ai metodi RMAN tradizionali. Questa procedura include la configurazione del listener, la creazione del pfile di standby, il popolamento dei volumi di standby con la replica GCNV, la finalizzazione dell'istanza Oracle e la registrazione dello standby con Oracle Restart. Tutti i livelli HA completano questi passaggi. Per il livello **Prod HA (Data Guard + FSFO)**, continua con Finalizzazione di Data Guard prima di configurare Data Guard Broker, Fast-Start Failover e Observer.

Passaggio 1: Configura i parametri del listener e di Data Guard

Configura il listener su entrambi gli host del database per supportare le connessioni Data Guard, incluso il `_DGMGRL` servizio necessario per il broker. Configura il file delle password e i parametri del log di archivio sul database primario.

1. Configura il listener principale e verifica l'ambiente su `oracdb1`:

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

2. Configura il listener di standby su `oracdb2` per includere i servizi `orcls` e `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. Riavvia il listener tramite Oracle Restart su entrambi gli host e verifica che il `_DGMGRL` servizio sia registrato:

```
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`deve elencare `<SID>`e `<SID>_DGMGRL`.

Passaggio 2: Prepara il pfile di standby e NOMOUNT

Prepara l'istanza del database di standby copiando il file delle password dalla primaria, creando un file pfile init.ora minimale con i parametri di Data Guard e avviando l'istanza in modalità NOMOUNT.

1. Copia il file della password primaria sull'host di standby utilizzando IAP e `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. Interroga il valore del parametro `compatible` dal database primario:

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

3. Crea il pfile di standby su `oracdb2`, imposta la proprietà del file password e avvia l'istanza in modalità NOMOUNT. Sostituisci il valore `compatible` del passaggio precedente con `<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
PFILE=/u01/app/oracle/product/26ai/db_1/dbs/initorcls.ora;
EXIT
SQL
'
```

L'istanza di standby è ora in modalità NOMOUNT senza file di dati fino a Passaggio 3: Inizializza lo storage di standby con GCNV.

Passaggio 3: Inizializza lo storage di standby con GCNV

Inserisci i volumi di standby in `oracle-pool-b`, collegali a `oracdb2`, monta i gruppi di dischi ASM e finalizza l'istanza di standby allo stato MOUNT.

Utilizza la replica GCNV per il seeding in produzione e il seeding snapshot per i flussi di lavoro di laboratorio una tantum.

Scegli il percorso di seeding

Scegli il metodo di seeding di standby in base al tuo ambiente e ai requisiti di recovery.

- **Consigliato per la produzione:** Usa il percorso di replica in Percorso di replica: crea e sincronizza le repliche e Percorso di replica: passaggio e collegamento dei volumi standby.
- **Alternativa per i laboratori:** Usa Percorso alternativo: seed da snapshot.

Tutti i percorsi si ricongiungono in Monta i gruppi di dischi ASM di standby e Finalizza l'istanza di standby.

Verifica i prerequisiti

Conferma i seguenti prerequisiti prima di eseguire il seeding dei volumi di standby.

- `gcloud netapp` con supporto per la replica dei volumi.

- Due pool **in modalità predefinita** in posizioni diverse (`oracle-pool-a`, `oracle-pool-b`).
- Volumi di origine sul pool primario collegati a `oracdb1-hg`; volumi di destinazione creati dalla replica.
- Esegui la replica da Cloud Shell o da una workstation, non dalle VM del database.
- Su `oracdb2`, completa la configurazione host iSCSI e ASM da Passo 4, Passo 5 e Passo 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 pool di standby se necessario:

```
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>
```

Crea e sincronizza le repliche

Crea relazioni di replica dai volumi primari ai volumi di standby, poi aspetta che la sincronizzazione iniziale sia completata.

```
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"
```

+

Attendi finché `mirrorState` è `MIRRORED` e la sincronizzazione iniziale è completata per ogni replica.

Passa e collega i volumi di standby

Metti in pausa il server primario, interrompi la replica dopo la sincronizzazione finale e collega i volumi di destinazione al gruppo host di standby.

Sul server primario, sospendi le scritture e acquisisci i metadati di recovery:

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

Consenti un ultimo ciclo di replica, poi interrompi le repliche:

```
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
```

Collega i volumi di destinazione a `oracdb2-hg` (le LUN replicate possono mantenere i nomi di origine):

```
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 il file di controllo di standby in `oracdb2`, quindi termina la modalità di backup sul primary:

```
ALTER DATABASE END BACKUP;
```

Seed dallo snapshot

Usa questo percorso per l'avvio una tantum in laboratorio quando la replica continua non è richiesta.

Per un'istanza di test una tantum, crea uno snapshot di origine e crea volumi di standby da tale snapshot in `oracle-pool-b` (Cloud Console o API). Collega i volumi creati a `oracdb2-hg`, quindi continua con Monta i gruppi di dischi ASM di standby.

Monta i gruppi di dischi ASM di standby

Sull'host di standby, scopri i percorsi di storage collegati e monta i gruppi di dischi ASM prima del recovery del database.

Su `oracdb2`, accedi ai portali iSCSI del pool di standby ed esegui una nuova scansione dei dispositivi multipath. Se le intestazioni del disco ASM corrispondono alla denominazione primaria in un flusso di lavoro di

laboratorio, usa alias in stile primario (per esempio `ora_oracdb1_data_01`, `ora_oracdb1_arch_01`), imposta `asm_diskstring='/dev/mapper/ora_oracdb1_*p*'` e conferma che la proprietà della partizione sia `grid:asmadmin`, poi monta i gruppi di dischi:

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

Finalizza l'istanza di standby

Ripristina il file di controllo di standby, esegui il ripristino sull'SCN acquisito, converti in standby fisico e avvia il ripristino gestito.

```
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;
```

A questo punto, lo standby dovrebbe essere `PHYSICAL STANDBY` e `MOUNTED` con il ripristino gestito avviato.

Prossimi passi specifici per ogni livello:

- **Prod HA (senza Data Guard):** Continua direttamente a Passaggio 4: registra lo standby con Oracle Restart.
- **Prod HA (Data Guard + FSFO):** Continua con Passaggio 4: registra lo standby con Oracle Restart, poi procedi con Passaggi finali di Data Guard.

Passaggio 4: registra lo standby con Oracle Restart

Registra il database standby con Oracle Restart così i riavvii recuperano automaticamente i gruppi di dischi ASM, montano il database standby e riavviano il ripristino gestito. Aggiungi anche il servizio dell'applicazione a entrambe le risorse del database.

1. Acquisisci la posizione dello spfile dal database standby e registralo con Oracle Restart su `oracdb2`. Sostituisci `<STANDBY_SPFILE_PATH>` dalla query (spesso sotto `+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. Verifica e aggiorna la risorsa del database primario su `oracdb1` per includere tutte le dipendenze del gruppo di dischi 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. Aggiungi il servizio dell'applicazione alla risorsa del database di standby (`orcls` su `oracdb2`). Usa `role PRIMARY` su entrambi i lati così `orclapp` è disponibile dopo il passaggio di consegne:

```
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. Verifica la risorsa del database di standby su `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
'
```

E ora?

Specifico per livello:

- **Prod HA (senza Data Guard):** Per mantenere una destinazione di ripristino basata sulla replica dello storage, l'inizializzazione dello standby è completata e il database standby è registrato con Oracle Restart come istanza di backup.
- **Prod HA (Data Guard + FSFO):** Per abilitare il passaggio gestito dal broker e il failover rapido, continua con Finalizza il database standby per Data Guard.

Finalizza il database standby per Data Guard su Google Cloud NetApp Volumes

Finalizza il database di standby per Oracle Data Guard su Google Cloud NetApp Volumes creando i file di standby redo log, abilitando il flashback database, attivando il redo shipping e verificando lo stato di Data Guard.

Specifico per livello: Questa procedura è richiesta solo per il livello **Prod HA (Data Guard + FSFO)**.

Passaggio 1: Crea i file di log di ripristino di standby

Crea file di standby redo log su entrambi gli host del database per supportare il Fast-Start Failover. La dimensione deve essere maggiore o uguale a quella del più grande primary online redo log e il numero deve essere pari a (gruppi online per thread) + 1. Dopo il seeding di GCNV, elimina e ricrea i standby redo log sullo standby per correggere i percorsi replicati.

1. Crea file di log di ripristino di standby sul database primario (`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 e ricrea i file di standby redo log sul database di standby (`orcls` dopo il seeding di GCNV. Percorsi replicati sotto `+DATA/ORCL/…` causano `ORA-19527` / `ORA-16086` fino alla ricostruzione:

```
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;
```

Passaggio 2: Abilita flashback e avvia il ripristino

Abilita flashback database sullo standby per supportare il ripristino automatico dopo il failover, poi avvia il ripristino gestito con real-time apply. Flashback deve essere abilitato prima di avviare il ripristino gestito perché non può essere abilitato mentre MRP è attivo.

1. Arresta il database standby, riavvialo in modalità MOUNT e abilita il flashback database su `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. Avvia il ripristino gestito con real-time apply:

```
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` abilita l'applicazione in real-time (il redo viene applicato non appena arriva negli SRL).

Passaggio 3: Abilita la spedizione dei redo

Abilita il trasporto dei redo dal primario allo standby attivando `LOG_ARCHIVE_DEST_STATE_2`, che è stato deliberatamente impostato su `DEFER` in Passo 2 della procedura di inizializzazione dello standby per sopprimere `ORA-12154` errori durante la creazione dello standby.

1. Passa `LOG_ARCHIVE_DEST_STATE_2` a `ENABLE` e forza un cambio di log per avviare il redo shipping:

```
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. Verifica che il reinvio funzioni correttamente:

```
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.
```

Se `dest_2` mostra `ORA-12154`, riavvia il primario. Dopo Passaggio 1: Abilitare il broker su entrambi i database, gestisci il trasporto tramite DGMGRL.

Passaggio 4: Verifica lo stato di Data Guard

Verifica che il database primario sia in modalità READ WRITE e che il database standby sia montato con ripristino gestito applicando i redo log.

1. Verifica il ruolo del database primario e la modalità di apertura su `oracdb1`:

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

2. Verifica il ruolo del database di standby, la modalità di apertura e lo stato del ripristino gestito su `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
```

Previsto sullo standby: `PHYSICAL STANDBY | MOUNTED`; `MRP0` con `APPLYING_LOG`.

3. Se lo standby segnala `MOUNTED` ma l'apply non è in esecuzione, riavvia il managed recovery su `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'
```

E ora?

Per attivare la gestione automatica dei ruoli e la protezione dal failover, continua con Configurare Oracle Data Guard Broker, Fast-Start Failover e Observer.

Configura Data Guard Broker e Fast-Start Failover per Oracle Database 26ai su Google Cloud NetApp Volumes

Configura Oracle Data Guard Broker e Fast-Start Failover con un Observer dedicato per abilitare le transizioni di ruolo automatiche per Oracle Database 26ai su Google Cloud NetApp Volumes.

Specifico per livello: Questa procedura si applica solo al livello **Prod HA (Data Guard + FSFO)**.

Questa procedura comprende l'abilitazione del broker su entrambi i database, la creazione della configurazione di Data Guard, l'abilitazione di FSFO con la modalità di protezione MaxAvailability, l'installazione di Oracle Instant Client sull'host Observer, l'avvio di Observer come servizio systemd con credenziali basate su wallet e il test di switchover e failover. Dopo `ENABLE CONFIGURATION`, gestisci il trasporto e i ruoli tramite **DGMGRL** (non `LOG_ARCHIVE_DEST_*` SQL ad hoc).

Passaggio 1: Abilita Data Guard Broker

Abilita il Data Guard Broker su entrambi gli host del database e crea la configurazione del broker che collega i database primario e di standby sotto gestione unificata.

1. Imposta `dg_broker_start=TRUE` sui database host del database primario e di standby:

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

2. Sul server primario, connettiti a DGMGRL con autenticazione del sistema operativo e crea la configurazione del broker:

> Solo sull'host Observer, utilizzare `dgmgrl /@orcl` dopo che il portafoglio di accesso automatico esiste. Non inserire le password sulla `dgmgrl` riga di comando.

```
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. Convalida la configurazione: correggi eventuali `WARNING` o valori non NULL `ERROR` prima di Passaggio 3: Configura le proprietà FSFO e abilita:

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

Passaggio 2: Conferma il flashback per FSFO

Conferma che flashback database sia abilitato su entrambi gli host. Flashback è necessario per FSFO auto-reinstate, che permette all'ex primario di rientrare automaticamente nella configurazione come standby dopo un failover.

1. Conferma `flashback_on` che `YES` è presente su entrambi gli host del database:

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

2. Solo sul primario, se la conservazione dei flashback non è già impostata:

```
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'
```

Passaggio 3: Configura e abilita FSFO

Imposta il trasporto dei redo SYNC, configura la modalità di protezione MaxAvailability, definisci le destinazioni FSFO su ciascun database e abilita il Fast-Start Failover.

1. Imposta la modalità di trasporto dei redo `SYNC` su entrambi i database e aumenta la modalità di protezione 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. Imposta le destinazioni FSFO in modo che ogni database indichi l'altro come destinazione di failover, quindi configura la soglia e il comportamento di ripristino automatico:

```
-- 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 FastStartFailoverAutoReinstate =
TRUE;
```

3. Abilita la funzione Fast-Start Failover e conferma la configurazione:

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

Passaggio 4: Installa Instant Client su Observer

Installa Oracle Instant Client sulla VM Observer dedicata (`oradg-obs`, crea un utente OS dedicato `oracle` e configura l'ambiente Oracle Net così che l'Observer possa connettersi a entrambi i membri del database su TCP/1521.

1. Installa i pacchetti Oracle Instant Client sull'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 `oracle` utente del sistema operativo dedicato che sarà il proprietario del wallet e dell'unità systemd:

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

3. Configura l'ambiente Oracle Net e crea `tnsnames.ora` con le voci per entrambi gli host del database:

```
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
```

Passaggio 5: Esegui Observer come servizio systemd

Crea un portafoglio di accesso automatico con le credenziali per entrambi i membri del database, poi configura e avvia l'Observer come servizio systemd così da sopravvivere ai riavvii e riconnettersi automaticamente alla configurazione.

Archivia le credenziali per un account amministrativo Data Guard dedicato (ad esempio, `SYSDG`) nel wallet anziché `SYS`. Le credenziali non devono mai apparire su una `dgmgrl` riga di comando, dove sono visibili a `ps` e `journalctl`; connettiti sempre usando `/@<tns_alias>` sull'Observer.

1. Crea il portafoglio crittografato e inserisci le credenziali per entrambi i membri del database:

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

2. Genera il portafoglio di accesso automatico `cwallet.sso` così il servizio systemd Observer può avviarsi senza richiesta della password. Se `cwallet.sso` manca dopo aver eseguito `mkstore`, usa `orapki` dal pacchetto di strumenti Instant Client o da una home del database per crearlo, poi aggiungi di nuovo le credenziali memorizzate:

```
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 l'unità systemd, abilita il servizio e verifica che l'Observer sia connesso:

```
sudo tee /etc/systemd/system/dgmgrl-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/dgmgrl -silent /@orcl "START
OBSERVER FILE IS '/var/lib/oracle/dgmgrl-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/dgmgrl-
observer.log

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

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

L'Observer deve leggere `CONNECTED` dal primario (un `DISCONNECTED` Observer sospende silenziosamente FSFO):

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

Passaggio 6: Testa FSFO

Convalida la configurazione di Data Guard con `VALIDATE DATABASE`, poi esegui uno switchover pianificato e, in una finestra di test, un failover non pianificato con reset della VM per confermare che FSFO funzioni end-to-end.

1. Esegui un test di commutazione pianificata e ripristina la topologia originale:

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

2. Esegui un failover non pianificato tramite il reset di una VM in una finestra di test controllata:

   Usa un **Reset** della VM (test in stile crash); un normale **Stop** potrebbe non attivare FSFO. Tail `/var/log/dgmgrl-observer.log` su `oradg-obs` per monitorare l'avanzamento del failover; ripristina la topologia al termine.

E ora?

La configurazione di Oracle Data Guard Broker, Fast-Start Failover e Observer è ora attiva per questa implementazione.

Informazioni sul copyright

Informazioni sul marchio commerciale