



Implante o Google Cloud NetApp Volumes com Oracle HA

NetApp database solutions

NetApp
June 25, 2026

Índice

Implante o Google Cloud NetApp Volumes com Oracle HA	1
Provisione instâncias do Google Compute Engine para o Google Cloud NetApp Volumes	1
Etapa 1: criar as VMs	1
Etapa 2: Configurar o firewall da VPC para TCP 1521	2
Etapa 3: Configurar nome do host, DNS e /etc/hosts	2
Etapa 4: prepare o sistema operacional somente nos hosts do banco de dados	3
Etapa 5: Capture o nome IQN do iniciador iSCSI	4
Qual é o próximo passo?	5
Provisionar armazenamento iSCSI do Google Cloud NetApp Volumes para Oracle Database 26ai	5
Etapa 1: criar pools iSCSI GCNV	5
Etapa 2: criar grupos de hosts	6
Etapa 3: criar volumes iSCSI GCNV	6
Etapa 4: Configurar iSCSI e multipath	7
Etapa 5: particionar dispositivos ASM	9
Etapa 6: formatar e montar /u01	10
Qual é o próximo passo?	11
Instale o Oracle Grid Infrastructure e o Oracle Database 26ai no Google Cloud NetApp Volumes	11
Etapa 1: Instale a infraestrutura de Grid em cada host do banco de dados	11
Etapa 2: instale o Oracle Database em cada host do banco de dados	14
Qual é o próximo passo?	16
Crie o banco de dados primário do Oracle no Google Cloud NetApp Volumes	16
Qual é o próximo passo?	18
Crie o banco de dados standby do Oracle com o seeding da camada de storage do Google Cloud NetApp Volumes	18
Etapa 1: Configurar os parâmetros do listener e do Data Guard	18
Etapa 2: prepare o pfile de standby e o NOMOUNT	19
Etapa 3: inicialize o armazenamento em espera com GCNV	21
Etapa 4: registre o standby com o Oracle Restart	24
Qual é o próximo passo?	26
Finalize o banco de dados standby para Data Guard no Google Cloud NetApp Volumes	26
Etapa 1: criar arquivos de redo log em espera	27
Passo 2: ative o flashback e inicie a recuperação	27
Etapa 3: Ative o envio de redo log	28
Etapa 4: Verifique o estado do Data Guard	29
Qual é o próximo passo?	30
Configure o Data Guard Broker e o Fast-Start Failover para o Oracle Database 26ai no Google Cloud NetApp Volumes	30
Etapa 1: habilitar o Data Guard Broker	31
Etapa 2: Confirme o flashback para FSFO	32
Etapa 3: configurar e ativar o FSFO	32
Passo 4: Instalar o Instant Client no Observer	33
Etapa 5: execute o Observer como um serviço systemd	34

Etapa 6: Testar FSFO	37
Qual é o próximo passo?	38

Implante o Google Cloud NetApp Volumes com Oracle HA

Provisione instâncias do Google Compute Engine para o Google Cloud NetApp Volumes

Provisione máquinas virtuais do Google Compute Engine para hospedar o Oracle Database 26ai no armazenamento iSCSI do Google Cloud NetApp Volumes. Este procedimento abrange a criação dos hosts de banco de dados primário e de standby e da VM Fast-Start Failover Observer, a configuração das regras de firewall da VPC para o Oracle Net, a configuração da resolução de nomes de host, a preparação do sistema operacional e a captura dos nomes dos iniciadores iSCSI para o provisionamento de storage GCNV.

Etapa 1: criar as VMs

Crie três VMs do Google Compute Engine em zonas diferentes da mesma região para isolamento de falhas zonais. Use o Cloud Console, `gcloud`, Terraform ou seu fluxo de trabalho de provisionamento padrão.

1. Crie as três máquinas virtuais com as especificações mostradas na tabela abaixo.

Prefira uma região com menor emissão de carbono para TCO e sustentabilidade, desde que atenda às necessidades de latência e conformidade (por exemplo `us-west1` vs `us-central1`):

VM	Zona	Tipo de máquina	Disco de inicialização	Rede	Propósito
oracdb1	us-west1-a	n4-highmem-8(exemplo) ou c4-standard-*	OL 10, 50 GB Hyperdisk Balanceado (somente SO)	oracle-vpc / oracle-subnet, gVNIC	Banco de dados primário
oracdb2	us-west1-b	Igual ao primário	OL 10, 50 GB Hyperdisk Balanceado (somente SO)	Mesmo	Banco de dados em espera
oradg-obs	us-west1-c	e2-medium	OL 10, 20 GB Hyperdisk Balanced	Mesmo	Observador FSFO (somente Instant Client)

Use o plano de rede Premium quando latência ou o volume de dados de saída (acima de ~200 GiB/mês) forem importantes; use o plano Standard para menor TCO em desenvolvimento/teste.

2. Ative os recursos de VM protegida e verifique a configuração do disco de inicialização:

Ative **Inicialização Segura**, **vTPM** e **Monitoramento de Integridade** em todas as três VMs.

O disco de inicialização contém apenas o sistema operacional. /u01 Os diretórios Grid/DB, de preparação e todos os dados ASM usam volumes iSCSI GCNV (consulte [Provisionar volumes iSCSI GCNV](#))

Não anexe um disco de dados GCE separado para /u01.

Etapa 2: Configurar o firewall da VPC para TCP 1521

Crie regras de firewall na VPC para permitir TCP/1521 entre as três VMs para transporte de rede do Oracle Net e conectividade do Observer. A ausência dessas regras interrompe a replicação do Data Guard.

1. Crie uma regra de entrada no firewall da VPC para permitir TCP/1521 entre todos os três IPs internos das VMs. Use regras de firewall da VPC ou políticas de firewall com a mesma lista de permissões.

Console da Nuvem: Rede VPC → Firewall → Criar regra allow-oracle-net-dbhosts em oracle-vpc — Entrada, Permitir, origens = três /32 IPs, TCP 1521. Espelhar saída se necessário.

2. Valide a conectividade de cada máquina virtual para verificar se as regras do firewall estão em vigor:

```
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	Esperado	Significado
22	Conectado	O caminho SSH funciona
1521	Conexão recusada	Firewall aberto; o ouvinte Grid inicia durante Etapa 1: Instale o Oracle Grid Infrastructure (Oracle Restart) em cada host do banco de dados
Qualquer	Tempo esgotado	Corrigir firewall ou roteamento

Execute a partir de todas as três máquinas virtuais em direção a cada endereço IP do par.

Etapa 3: Configurar nome do host, DNS e /etc/hosts

Configure o nome do host e a resolução de DNS em todas as três VMs para que a resolução de nomes direta e reversa funcione para o Oracle Net, o Data Guard Broker e o Observer.

1. Defina o nome do host e adicione /etc/hosts entradas em todos os três hosts. Substitua os endereços IP internos do GCE (visíveis na lista **Compute Engine** → **Instâncias de VM**, coluna *IP interno*):

```
# 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. Valide a resolução de nomes de cada host:

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

Etapa 4: prepare o sistema operacional somente nos hosts do banco de dados

Prepare o sistema operacional em `oracdb1` e `oracdb2` para o Oracle Database 26ai instalando o pacote de pré-instalação, criando usuários e grupos, instalando os pacotes iSCSI e multipath, e configurando o iniciador iSCSI. A configuração do Observer é abordada em [Etapa 4: Instale o Oracle Instant Client no host do Observer](#).



Pré-requisito: HTTPS de saída para `yum.oracle.com` (NAT na nuvem ou espelhamento interno em sub-redes privadas).

1. Instale o pacote de pré-instalação do Oracle Database, crie o `grid` usuário e os grupos ASM, e adicione o `oracle` usuário aos 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. Instale os pacotes iSCSI, multipath e JDK e, em seguida, verifique o THP e a sincronização de tempo:

```
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. Configure as configurações do SELinux, do firewall e do iniciador iSCSI e reinicialize:



Postura de segurança (OL 10): Os comandos abaixo configuram o SELinux como permissivo e desativam `firewalld`. Esta é apenas uma postura mínima de laboratório. Para configurações mais robustas de SELinux e firewall, consulte a política de segurança da sua organização.

```
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
```

Etapa 5: Capture o nome IQN do iniciador iSCSI

Capture o nome do iniciador iSCSI (IQN) de cada host do banco de dados após a reinicialização. Você usará esses IQNs para criar os grupos de hosts GCNV em [Etapa 2: criar os grupos de hosts](#).

1. Capture o IQN de `oracdb1` e registre-o:

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

2. Repita em `oracdb2` e registre seu IQN. Use um grupo de hosts por host para que a reinicialização ou regeneração do IQN de um único host não afete a visibilidade do volume iSCSI GCNV de outro host.



VMs clonadas: Se ambos os hosts compartilharem o mesmo IQN, regenere em `oracdb2` (`stop iscsi`, `clear /var/lib/iscsi/nodes/*`, `new InitiatorName in /etc/iscsi/initiatorname.iscsi`, reiniciar `iscsid`).

Qual é o próximo passo?

Para fornecer storage compartilhado para binários do Oracle e grupos de discos ASM, acesse [Provisione pools iSCSI, grupos de hosts e volumes do Google Cloud NetApp Volumes](#).

Provisionar armazenamento iSCSI do Google Cloud NetApp Volumes para Oracle Database 26ai

Provisione o armazenamento em bloco iSCSI do Google Cloud NetApp Volumes para alta disponibilidade do Oracle Database 26ai no Google Compute Engine. Este procedimento abrange a criação de pools de storage unificados GCNV Flex, definição de grupos de hosts, criação de volumes iSCSI para cada host de banco de dados, configuração do iSCSI e multipath no Linux, particionamento dos dispositivos de suporte ASM e montagem do `/u01` filesystem.

Etapa 1: criar pools iSCSI GCNV

Crie dois pools de armazenamento Flex Unified, um em cada zona de banco de dados, para fornecer volumes iSCSI para os hosts primário e de espera. Cada host de banco de dados usa volumes do pool de sua zona local.

1. Crie dois pools de armazenamento usando o Cloud Console. Utilize as especificações na tabela abaixo e repita o processo de criação para cada zona:

Nome do pool	Zona	Usado por
oracle-pool-a	us-west1-a	oracdb1 (primário)
oracle-pool-b	us-west1-b	oracdb2 (espera)

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

- **Nível de serviço:** Flex (não Premium)
 - **Tipo:** Unificado
 - **Zona:** corresponde à zona da VM do banco de dados (`us-west1-a/ us-west1-b`)
 - **Aviso:** conectado a `oracle-vpc`
 - **Capacidade:** dimensionada para a carga de trabalho; utilize a taxa de transferência/IOPS provisionada personalizada quando refazer, backup ou restaurar exceder a capacidade padrão (até 5120 MiB/s ou 160K IOPS por pool, conforme os limites do produto)
2. Aguarde até que ambos os pools atinjam o status `READY` antes de prosseguir. Ajuste o tamanho dos pools de acordo com o tamanho do seu banco de dados (os tamanhos em [Etapa 3: Crie os volumes iSCSI do GCNV](#) são exemplos):



Modo padrão (este guia): Os pools Flex Unified usam o modo padrão (`--mode=default`). Crie pools e volumes iSCSI com o Cloud Console ou `gcloud netapp`. A replicação de volumes, snapshots e clones usa as APIs do Google Cloud ([Etapa 3: inicialização do modo de espera do GCNV](#)).

Etapa 2: criar grupos de hosts

Crie um grupo de hosts por host de banco de dados para que cada VM veja apenas seus próprios volumes. Os hosts primário e de espera não devem compartilhar volumes iSCSI GCNV para manter o storage independente.

1. Crie o grupo de hosts para `oracdb1` usando o Console da Nuvem:

NetApp Volumes → Grupos de hosts → Criar

- **Nome:** `oracdb1-hg`
 - **Região:** `us-west1`
 - **Tipo:** iniciador iSCSI
 - **Tipo de SO:** Linux
 - **Anfitriões:** cole o IQN de `oracdb1` (o valor de `/etc/iscsi/initiatorname.iscsi`)
 - **Descrição:** "Host primário do Oracle `oracdb1`"
 - **Criar**
2. Repita o processo `oracdb2` com o nome `oracdb2-hg` e o IQN de `oracdb2`. O host do Observer não requer recursos do GCNV.

Etapa 3: criar volumes iSCSI GCNV

Crie cinco volumes iSCSI GCNV para cada host de banco de dados: um para `/u01` e quatro para os dispositivos de suporte ASM. Os volumes de cada host devem ser criados no pool de storage da zona local, com o grupo de hosts correspondente.

1. Crie os cinco volumes para `oracdb1` em `oracle-pool-a` com o grupo de hosts `oracdb1-hg`. Use as especificações na tabela abaixo:

Volume iSCSI GCNV	Tamanho	Usar	Alias de multipath
<code>ora_<host>_u01</code>	100 GiB	<code>/u01</code> Volume iSCSI GCNV — Grid/Oracle homes, staging	<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 (listrado)	<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>
<code>ora_<host>_fra_01</code>	100 GiB	ASM +FRA	<code>/dev/mapper/ora_<host>_fra_01</code>

Nomes de volumes: apenas letras, números e sublinhados (sem hífen).



Layout mínimo (apenas para validação): Dois LUNs por host (*_data, *_reco) com arch_01p1→+RECO e arch_01p2→+FRA é aceitável para laboratório; produção usa cinco volumes por [Etapa 3: Crie os volumes iSCSI do GCNV](#).

2. Crie os cinco volumes para oracdb2 em oracle-pool-b com o grupo de hosts oracdb2-hg usando as mesmas especificações. Para cada pool, use **NetApp Volumes** → **Volumes** → **Create** — iSCSI, pool e grupo de hosts corretos, Linux. Registre as seguintes informações:
 - IPs do portal iSCSI → <ISCSI_PORTAL_1>, <ISCSI_PORTAL_2> (portais do pool primário em oracdb1; portais do pool de espera em oracdb2 — eles podem ser diferentes)
 - Número de série do volume do Console da Nuvem — use com o WWID descoberto pelo host em [Etapa 4: configure o iSCSI e o multipath do Linux para volumes iSCSI do GCNV](#)

Etapa 4: Configurar iSCSI e multipath

Configure iSCSI e device-mapper-multipath em cada host de banco de dados para acessar os volumes GCNV por meio de ambos os IPs do portal de storage. Execute estas etapas em oracdb1 usando os IPs do portal do pool primário e, em seguida, repita em oracdb2 usando os IPs do portal do pool de standby. Se a saída do host estiver restrita, permita TCP/3260 de cada VM de banco de dados para seus IPs do portal iSCSI GCNV (além de TCP/1521 entre VMs [Etapa 2: firewall da VPC — adicione TCP/1521 à lista de permissões em todas as três zonas](#)).

1. Descubra os alvos, faça login e persista a inicialização do nó:

```
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
```

Após reinicializar, verifique novamente antes de iniciar Oracle:

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

2. Configurar `device-mapper-multipath` com valores padrão e regras de 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. Adicione aliases WWID descobertos pelo host /etc/multipath.conf (não tente adivinhar — multipath.conf **não** expande variáveis de shell). Descubrir WWIDs:

```

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

```

Adicione aliases concretos para esse host a /etc/multipath.conf, depois `sudo systemctl restart multipathd`.

Em `oracdb1`, anexe:

```

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  }
}

```

Em oracdb2, use o mesmo padrão com ora_oracdb2_* aliases, então:

```

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

```

Etapa 5: particionar dispositivos ASM

Particione os quatro dispositivos de suporte ASM (excluindo u01) com uma partição GPT cada para consumo do ASM e, em seguida, configure as regras udev para a propriedade do grid. Execute estas etapas em cada host do banco de dados.

1. Particione os quatro dispositivos de suporte ASM com GPT e verifique as partições:

```

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. Configure as regras udev para atribuir a propriedade de grid e acionar as alterações:

```

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

```

Etapa 6: formatar e montar /u01

Formate o ora_<host>_u01 volume GCNV com XFS e monte-o persistentemente usando UUID em /etc/fstab. O /u01 sistema de arquivos contém Grid home, Oracle home e arquivos de staging.

1. Formate o dispositivo multipath com XFS e capture seu 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. Adicione a entrada de montagem baseada em UUID /etc/fstab e monte o sistema de arquivos:

```

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. Crie a estrutura de diretórios com a propriedade adequada para o 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
```

Reinicie uma vez e confirme /u01 montagens antes de [instalando o software Oracle](#).

Qual é o próximo passo?

Para instalar os binários do Oracle Grid Infrastructure e do banco de dados nos hosts preparados, acesse [Instale o software Oracle Grid Infrastructure e Oracle Database](#) em ambos os hosts.

Instale o Oracle Grid Infrastructure e o Oracle Database 26ai no Google Cloud NetApp Volumes

Instale o Oracle Grid Infrastructure com Oracle Restart e ASM no armazenamento iSCSI do Google Cloud NetApp Volumes para cada host de banco de dados e, em seguida, instale o software Oracle Database 26ai. Este procedimento inclui a preparação do Oracle GoldImages, a execução de instalações silenciosas com arquivos de resposta, a criação de grupos de discos ASM em volumes GCNV e a preparação dos hosts primário e de espera com o mesmo software Oracle antes da criação do banco de dados.

Etapa 1: Instale a infraestrutura de Grid em cada host do banco de dados

Instale a Oracle Grid Infrastructure GoldImage em cada host de banco de dados para habilitar o Oracle Restart e o ASM. Ambos os hosts requerem seu próprio Grid home, instância ASM e grupos de discos; o Data Guard replica dados pela Oracle Net, não por storage compartilhado. Conclua todas as etapas em `oracdb1` antes de repeti-las em `oracdb2`.

1. Prepare os binários do Oracle GoldImages, da Release Update e do OPatch em /u01/stage:

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

2. Descompacte o Grid GoldImage no local de destino do Grid home. O 26ai GoldImage é instalado descompactando diretamente no diretório 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

```

Se o Grid GoldImage for mais antigo que a RU de destino, aplique o patch no diretório raiz do Grid durante a configuração usando o `gridSetup.sh -applyRU` fluxo, ou use uma GoldImage com a RU incluída. Mantenha os diretórios raiz do Grid e do banco de dados no mesmo nível de patch pretendido.

3. Crie o `gridSetup` arquivo de resposta `/tmp/grid.rsp` em cada host. Substitua o nome do host e use senhas fortes:

```

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. Execute `gridSetup.sh` em modo silencioso para copiar os binários e preparar a configuração. Espere `Successfully Setup Software with warning(s)` e código de saída 6 (avisos) ou 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. Execute `oraInstRoot.sh` e `root.sh` como `root`. O script `root.sh` cria os wrappers `crsctl`, `srvctl` e `asmcmd` e inicia o OHAS:

```

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

6. Execute `gridSetup.sh -executeConfigTools` para executar os assistentes de configuração (NETCA, ASMCA, CVU) em [o arquivo de resposta](#). Isso cria a instância ASM e o +DATA grupo de discos. Espere Successfully Configured Software. após 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. Crie os +RECO e +FRA grupos de discos usando `asmca`. A instalação única cria apenas +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. Verifique os grupos de discos ASM e o status dos recursos do 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. Repita os passos acima em oracdb2. O HOST=\$(hostname -s) padrão em [passos 3 e 4](#) e [passo 7](#) seleciona automaticamente os dispositivos iSCSI GCNV desse host.

Use os mesmos nomes de grupo de discos ASM — o Data Guard replica pela Oracle Net, não pelo armazenamento.

Etapa 2: instale o Oracle Database em cada host do banco de dados

Instale o diretório de instalação do software Oracle Database 26ai em cada host de banco de dados usando uma instalação silenciosa, somente de software, com a Release Update mais recente aplicada. Conclua todas as etapas em oracdb1 antes de repetir em oracdb2.

1. Descompacte o diretório raiz do banco de dados, o OPatch mais recente e o patch RU em seus respectivos diretórios. Consulte a documentação da Oracle para obter informações sobre o layout do diretório RU e o -applyRU caminho:

```
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. Grave o arquivo de resposta da instalação e execute a instalação somente de software em modo silencioso com a RU aplicada. No OL 8/9, omita -applyOneOffs da linha 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. Execute o script root pós-instalação:

```

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

4. Defina o ambiente Oracle em cada host do banco de dados. Use ORACLE_SID=orcl em oracdb1 e ORACLE_SID=orcls em 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
```

O banco de dados standby é criado em [Crie o banco de dados de espera](#).

Qual é o próximo passo?

Para criar a instância primária de produção para sua implantação de alta disponibilidade, acesse [Crie o banco de dados primário do Oracle](#) em oracdb1.

Crie o banco de dados primário do Oracle no Google Cloud NetApp Volumes

Crie o banco de dados primário do Oracle no armazenamento iSCSI do Google Cloud NetApp Volumes usando o Assistente de Configuração do Banco de Dados Oracle no modo silencioso. Este procedimento abrange executar `dbca` para criar o banco de dados de contêiner e o banco de dados plugável em grupos de discos ASM com suporte do GCNV, configurar os destinos de log de arquivamento e adicionar um serviço de aplicativo baseado em funções para failover transparente após a ativação do Data Guard.

Passos

Crie o banco de dados de contêiner Oracle e o banco de dados plugável em oracdb1 usando `dbca` no modo silencioso, configure os destinos de log de arquivamento, verifique o registro do Oracle Restart e adicione um serviço de aplicativo baseado em funções para failover transparente do cliente.

1. Execute `dbca` em modo silencioso para criar o CDB e o PDB nos 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. Aponte os logs de arquivamento para +RECO e abra e salve o estado do banco de dados plugável. O standby usa configurações de log de arquivamento correspondentes em [Etapa 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. Verifique se o banco de dados está em execução no 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. Crie um serviço de aplicativo baseado em funções para que os aplicativos se conectem via orclapp e o failover seja transparente quando o Data Guard estiver ativado:

```

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
'
```

Após a ativação do Data Guard Broker, orclapp é executado somente no PRIMARY. Multiplixe os arquivos de controle entre os grupos de discos ASM e dimensione a memória de acordo com a carga de trabalho.

Qual é o próximo passo?

Para configurar a proteção em espera e a prontidão para failover, acesse [Crie o banco de dados standby do Oracle](#) em `oracdb2`.

Crie o banco de dados standby do Oracle com o seeding da camada de storage do Google Cloud NetApp Volumes

Crie o banco de dados standby físico do Oracle usando a replicação na camada de storage do Google Cloud NetApp Volumes, snapshots ou clones para acelerar a inicialização do standby em comparação com os métodos tradicionais do RMAN. Este procedimento abrange configurar o listener, criar o pfile do standby, preencher os volumes do standby com replicação GCNV, finalizar a instância do Oracle e registrar o standby com o Oracle Restart. Todas as camadas de HA concluem essas etapas. Para a camada **Prod HA (Data Guard + FSFO)**, continue com [Finalização do Data Guard](#) antes de configurar [Data Guard Broker](#), [Fast-Start Failover](#) e o [Observer](#).

Etapa 1: Configurar os parâmetros do listener e do Data Guard

Configure o listener em ambos os hosts do banco de dados para suportar conexões Data Guard, incluindo o `_DGMGRL` serviço necessário para o broker. Configure o arquivo de senhas e configure os parâmetros de arquivamento de logs no banco de dados primário.

1. Configure o ouvinte primário e verifique o ambiente em `oracdb1`:

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

2. Configure o ouvinte em espera em `oracdb2` para incluir os `orcls` e `orcls_DGMGRL` serviços:

```

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. Reinicie o listener através do Oracle Restart em ambos os hosts e verifique se o `_DGMGRL` serviço está 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` deve listar `<SID>` e `<SID>_DGMGRL`.

Etapa 2: prepare o pfile de standby e o NOMOUNT

Prepare a instância de banco de dados em espera copiando o arquivo de senha do primário, criando um arquivo `init.ora` mínimo com parâmetros do Data Guard e iniciando a instância no modo NOMOUNT.

1. Copie o arquivo de senha principal para o host em espera usando 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. Consulte o valor do parâmetro `compatible` no banco de dados principal:

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

3. Crie o arquivo pfile de standby em `oracdb2`, defina a propriedade do arquivo de senha e inicie a instância no modo `NOMOUNT`. Substitua o valor de `compatible` da etapa 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
PFILE=/u01/app/oracle/product/26ai/db_1/dbs/initorcls.ora;
EXIT
SQL
'

```

A instância em espera agora está no modo NOMOUNT, sem arquivos de dados até [Etapa 3: inicialize o armazenamento em espera com GCNV](#).

Etapa 3: inicialize o armazenamento em espera com GCNV

Inicialize os volumes de standby em `oracle-pool-b`, anexe-os a `oracdb2`, monte os grupos de discos ASM e finalize a instância de standby no estado MOUNT.

Use a replicação GCNV para o seeding de produção e o seeding por snapshot para fluxos de trabalho de laboratório pontuais.

Escolha o caminho de seeding

Escolha o método de semeadura de standby com base no seu ambiente e nos requisitos de recuperação.

- **Recomendado para produção:** Use o caminho de replicação em [Caminho de replicação: criar e sincronizar replicações](#) e [Caminho de replicação: migrar e anexar volumes em espera](#).
- **Alternativa para laboratórios:** Use [Caminho alternativo: seed a partir de um snapshot](#).

Todos os caminhos se reencontram em [Montar grupos de discos ASM em espera](#) e [Finalize a instância em espera](#).

Verifique os pré-requisitos

Confirme os seguintes pré-requisitos antes de inicializar os volumes standby.

- gcloud netapp com suporte para replicação de volume.
- Duas pools em **Default-mode** em locais diferentes (oracle-pool-a, oracle-pool-b).
- Volumes de origem no pool primário anexados a oracdb1-hg; volumes de destino criados por replicação.
- Execute a replicação a partir do Cloud Shell ou da estação de trabalho — não a partir de DB VMs.
- Em oracdb2, conclua a configuração completa do host iSCSI e ASM a partir de [Etapa 4](#), [Etapa 5](#) e [Etapa 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"
```

- Crie um pool de espera se necessário:

```
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>
```

Criar e sincronizar replicações

Crie relações de replicação dos volumes primários para os volumes em espera, depois aguarde a conclusão da sincronização 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"
```

+

Aguarde até mirrorState estar MIRRORED e a sincronização inicial ser concluída para cada replicação.

Corte e anexe volumes standby

Coloque o host primário em estado quiescente, interrompa a replicação após a sincronização final e anexe os volumes de destino ao grupo de hosts em espera.

No servidor primário, suspenda as gravações e capture os metadados de recuperação:

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

Permita um último ciclo de replicação e, em seguida, interrompa as replicações:

```
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
```

Anexe os volumes de destino a oracdb2-hg (os LUNs replicados podem manter os nomes de origem):

```
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"
```

Copie o arquivo de controle de espera para oracdb2, depois encerre o modo de backup no servidor primário:

```
ALTER DATABASE END BACKUP;
```

Semear a partir de um snapshot

Use este caminho para semear o laboratório uma única vez, quando a replicação contínua não for necessária.

Para uma configuração inicial de laboratório única, crie um snapshot de origem e crie volumes de espera a partir desse snapshot em oracle-pool-b (Console do Cloud ou API). Anexe os volumes criados a oracdb2-hg, depois continue com [Montar grupos de discos ASM em espera](#).

Montar grupos de discos ASM em espera

No host de espera, descubra os caminhos de storage conectados e monte os grupos de discos ASM antes da recuperação de banco de dados.

Em `oracdb2`, faça login nos portais iSCSI do pool de espera e faça uma nova varredura nos dispositivos multipath. Se os cabeçalhos de disco ASM corresponderem à nomenclatura primária em um fluxo de trabalho de laboratório, use aliases no estilo primário (por exemplo, `ora_oracdb1_data_01`, `ora_oracdb1_arch_01`), defina `asm_diskstring='/dev/mapper/ora_oracdb1_*p*'` e confirme se a propriedade da partição é `grid:asmadmin`, depois monte os grupos de discos:

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

Finalize a instância em espera

Restaure o controlfile em standby, recupere até o SCN capturado, converta para standby físico e inicie a recuperação gerenciada.

```
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;
```

Neste ponto, o standby deve estar `PHYSICAL STANDBY` e `MOUNTED` com a recuperação gerenciada iniciada.

Próximos passos específicos para cada tier:

- **Produção HA (sem Data Guard):** Continue diretamente para [Etapa 4: registre o standby com o Oracle Restart](#).
- **Prod HA (Data Guard + FSFO):** Continue para [Etapa 4: registre o standby com o Oracle Restart](#), depois prossiga para [Etapas de finalização do Data Guard](#).

Etapa 4: registre o standby com o Oracle Restart

Registre o banco de dados standby no Oracle Restart para que as reinicializações recuperem automaticamente os grupos de discos ASM, montem o banco de dados standby e reiniciem a recuperação gerenciada. Adicione também o serviço de aplicativo aos dois recursos de banco de dados.

1. Capture a localização do spfile do banco de dados standby e registre-a com o Oracle Restart em `oracdb2`. Substitua `<STANDBY_SPFILE_PATH>` da consulta (geralmente em `+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. Verifique e atualize o recurso de banco de dados primário em `oracdb1` para incluir todas as dependências do grupo 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. Adicione o serviço de aplicação ao recurso de banco de dados em espera (`orcls` em `oracdb2`). Use `role PRIMARY` em ambos os lados para que `orclapp` esteja disponível após switchover:

```

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. Verifique o recurso de banco de dados em espera em 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
'
```

Qual é o próximo passo?

Específico para cada tier:

- **Prod HA (sem Data Guard):** Para manter um destino de recuperação baseado em replicação de armazenamento, a inicialização do standby está concluída e o banco de dados standby está registrado no Oracle Restart como uma instância de backup.
- **Prod HA (Data Guard + FSFO):** Para habilitar o switchover gerenciado pelo broker e o failover de inicialização rápida, continue com [Finalize o banco de dados standby para Data Guard](#).

Finalize o banco de dados standby para Data Guard no Google Cloud NetApp Volumes

Finalize o banco de dados standby para o Oracle Data Guard no Google Cloud NetApp Volumes criando arquivos de redo log standby, habilitando o flashback database, ativando o redo shipping e verificando o estado do Data Guard.

Específico para cada tier: Este procedimento é necessário apenas para o tier **Prod HA (Data Guard + FSFO)**.

Etapa 1: criar arquivos de redo log em espera

Crie arquivos de redo log em espera em ambos os hosts do banco de dados para suportar o Fast-Start Failover. O tamanho deve ser maior ou igual ao maior redo log online primário, e a contagem deve ser igual a (grupos online por thread) + 1. Após a inicialização do GCNV, exclua e recrie os redo logs em espera no standby para corrigir os caminhos replicados.

1. Crie arquivos de standby redo log no banco de dados primário (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. Exclua e recrie os arquivos de redo log em espera no banco de dados em espera (orcls após o seeding do GCNV. Caminhos replicados sob +DATA/ORCL/... causam ORA-19527 / ORA-16086 até serem reconstruídos:

```
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;
```

Passo 2: ative o flashback e inicie a recuperação

Habilite o flashback database no standby para suportar a reinstalação automática após o failover e, em seguida, inicie a recuperação gerenciada com aplicação em tempo real. O flashback deve ser habilitado antes de iniciar a recuperação gerenciada, pois não pode ser habilitado enquanto o MRP estiver ativo.

1. Desligue o banco de dados em espera, reinicie no modo MOUNT e habilite o flashback do banco de dados em 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. Inicie a recuperação gerenciada com aplicação em tempo 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 a aplicação em tempo real (redo log aplicado assim que chega aos SRLs).

Etapa 3: Ative o envio de redo log

Habilite o transporte de redo do primário para o standby ativando LOG_ARCHIVE_DEST_STATE_2, que foi definido deliberadamente como DEFER em [Etapa 2](#) do procedimento de inicialização do standby para suprimir ORA-12154 erros durante a criação do standby.

1. Alterne LOG_ARCHIVE_DEST_STATE_2 para ENABLE e force uma troca de log para iniciar o envio de redo:

```
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. Verifique se o redo log está funcionando corretamente:

```
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 mostrar ORA-12154, reinicie o primário. Após [Passo 1: Habilite o broker em ambos os bancos de dados](#), gerencie o transporte via DGMGRL.

Etapa 4: Verifique o estado do Data Guard

Verifique se o banco de dados primário está no modo READ WRITE e se o banco de dados em espera está montado com recuperação gerenciada aplicando redo logs.

1. Verifique a função primária do banco de dados e o modo de abertura em oracdb1:

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

2. Verifique a função do banco de dados em espera, o modo aberto e o status da recuperação gerenciada em 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

```

Esperado em standby: PHYSICAL STANDBY | MOUNTED; MRP0 com APPLYING_LOG.

3. Se o standby reportar MOUNTED, mas o apply não estiver em execução, reinicie a recuperação gerenciada em 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'

```

Qual é o próximo passo?

Para ativar o gerenciamento automático de funções e a proteção contra falhas, continue com [Configurar o Oracle Data Guard Broker, Fast-Start Failover e Observer](#).

Configure o Data Guard Broker e o Fast-Start Failover para o Oracle Database 26ai no Google Cloud NetApp Volumes

Configure o Oracle Data Guard Broker e o Fast-Start Failover com um Observer dedicado para habilitar transições automáticas de função para o Oracle Database 26ai no Google Cloud NetApp Volumes.

Específico para cada camada: Este procedimento se aplica apenas à camada **Prod HA (Data Guard + FSFO)**.

Este procedimento abrange a ativação do broker em ambos os bancos de dados, a criação da configuração do Data Guard, a ativação do FSFO com o modo de proteção MaxAvailability, a instalação do Oracle Instant Client no host do Observer, a inicialização do Observer como um serviço systemd com credenciais baseadas em wallet e o teste de switchover e failover. Após `ENABLE CONFIGURATION`, gerencie o transporte e as funções por meio do **DGMGRL** (não por `SQL LOG_ARCHIVE_DEST_*` ad-hoc).

Etapa 1: habilitar o Data Guard Broker

Habilite o Data Guard Broker em ambos os hosts do banco de dados e crie a configuração do broker que vincula os bancos de dados primário e standby sob gerenciamento unificado.

1. Configurar `dg_broker_start=TRUE` nos hosts de banco de dados primário e de espera:

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

2. No servidor primário, conecte-se ao DGMGRL com autenticação do sistema operacional e crie a configuração do broker:



Somente no host do Observer, use `dgmgrl /@orcl` após a carteira de login automático existir. Não insira senhas na `dgmgrl` linha de 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. Valide a configuração — corrija qualquer WARNING ou valor não nulo ERROR antes de [Etapa 3: configure as propriedades do FSFO e habilite](#):

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

Etapa 2: Confirme o flashback para FSFO

Confirme se o flashback database está habilitado em ambos os hosts. O flashback é necessário para o auto-reinstate do FSFO, que permite que o antigo primário se junte automaticamente à configuração como standby após um failover.

1. Confirme `flashback\_on` está `YES` em ambos os hosts do banco de dados:

```
sudo -u oracle bash -c '
. ~/.bash_profile
sqlplus -s / as sysdba <<<"SELECT flashback_on FROM v\\$database;"
'

# Expected on both hosts: YES
```

2. Somente no primário, se a retenção de flashback ainda não estiver configurada:

```
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'
```

Etapa 3: configurar e ativar o FSFO

Defina o transporte de redo SYNC, configure o modo de proteção MaxAvailability, defina os destinos FSFO em cada banco de dados e habilite o Fast-Start Failover.

1. Defina o modo de transporte de refazer para SYNC em ambos os bancos de dados e aumente o modo de proteção para 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. Configure os destinos FSFO para que cada banco de dados nomeie o outro como seu destino de failover e, em seguida, configure o limite e o comportamento de reinstalação automática:

```
-- 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. Ative o Fast-Start Failover e confirme a configuração:

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

Passo 4: Instalar o Instant Client no Observer

Instale o Oracle Instant Client na VM dedicada do Observer (oradg-obs, crie um usuário dedicado do sistema operacional `oracle` e configure o ambiente Oracle Net para que o Observer possa se conectar a ambos os membros do banco de dados na porta TCP/1521.

1. Instale os pacotes do Oracle Instant Client no 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. Crie um usuário dedicado `oracle` do sistema operacional que será o proprietário da wallet e da unidade `systemd`:

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

3. Configure o ambiente Oracle Net e crie `tnsnames.ora` com entradas para ambos os hosts de banco de dados:

```

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

```

Etapa 5: execute o Observer como um serviço systemd

Crie uma carteira de login automático com credenciais para ambos os membros do banco de dados, depois configure e inicie o Observer como um serviço systemd para que ele sobreviva a reinicializações e se reconecte automaticamente à configuração.

Armazene as credenciais de uma conta administrativa dedicada do Data Guard (por exemplo, SYSDG) na carteira em vez de SYS. As credenciais nunca devem aparecer em uma `dgmgrl` linha de comando, onde ficam visíveis para `ps` e `journalctl`; sempre conecte usando `/@<tns_alias>` no Observer.

1. Crie a carteira criptografada e insira as credenciais para ambos os membros do banco de dados:

```

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.pl2

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 dgmlr1 /@orcl 'SHOW CONFIGURATION;'
sudo -iu oracle dgmlr1 /@orcls 'SHOW CONFIGURATION;'

```

2. Gere a carteira de login automático `cwallet.sso` para que o serviço `systemd` do Observer possa iniciar sem solicitar senha. Se `cwallet.sso` estiver faltando após executar `mkstore`, use `orapki` do pacote de ferramentas do Instant Client ou de um diretório raiz do banco de dados para criá-la e, em seguida, adicione novamente as credenciais armazenadas.

```

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. Crie a unidade systemd, habilite o serviço e verifique se o 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

```

O observador deve ler CONNECTED do primário (um DISCONNECTED observador suspende silenciosamente o FSFO):

```

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

```

Etapa 6: Testar FSFO

Valide a configuração do Data Guard com `VALIDATE DATABASE`, depois execute uma switchover planejada e, em uma janela de teste, um failover de reinicialização de VM não planejado para confirmar que o FSFO está funcionando de ponta a ponta.

1. Teste um switchover planejado e restaure a topologia original:

```

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

```

2. Teste uma falha não planejada usando uma reinicialização da VM em uma janela de teste controlada:

Use um **Reset** da VM (teste de falha simulada); uma **Parada** normal pode não acionar o FSFO. Acompanhe o `/var/log/dgmgml-observer.log`tail `oradg-obs` para monitorar o progresso do failover; restaure a topologia quando terminar.

Qual é o próximo passo?

A configuração do Oracle Data Guard Broker, Fast-Start Failover e Observer já está em vigor para esta implantação.

Informações sobre direitos autorais

Copyright © 2026 NetApp, Inc. Todos os direitos reservados. Impresso nos EUA. Nenhuma parte deste documento protegida por direitos autorais pode ser reproduzida de qualquer forma ou por qualquer meio — gráfico, eletrônico ou mecânico, incluindo fotocópia, gravação, gravação em fita ou storage em um sistema de recuperação eletrônica — sem permissão prévia, por escrito, do proprietário dos direitos autorais.

O software derivado do material da NetApp protegido por direitos autorais está sujeito à seguinte licença e isenção de responsabilidade:

ESTE SOFTWARE É FORNECIDO PELA NETAPP "NO PRESENTE ESTADO" E SEM QUAISQUER GARANTIAS EXPRESSAS OU IMPLÍCITAS, INCLUINDO, SEM LIMITAÇÕES, GARANTIAS IMPLÍCITAS DE COMERCIALIZAÇÃO E ADEQUAÇÃO A UM DETERMINADO PROPÓSITO, CONFORME A ISENÇÃO DE RESPONSABILIDADE DESTES DOCUMENTOS. EM HIPÓTESE ALGUMA A NETAPP SERÁ RESPONSÁVEL POR QUALQUER DANO DIRETO, INDIRETO, INCIDENTAL, ESPECIAL, EXEMPLAR OU CONSEQUENCIAL (INCLUINDO, SEM LIMITAÇÕES, AQUISIÇÃO DE PRODUTOS OU SERVIÇOS SOBRESSALIENTES; PERDA DE USO, DADOS OU LUCROS; OU INTERRUPÇÃO DOS NEGÓCIOS), INDEPENDENTEMENTE DA CAUSA E DO PRINCÍPIO DE RESPONSABILIDADE, SEJA EM CONTRATO, POR RESPONSABILIDADE OBJETIVA OU PREJUÍZO (INCLUINDO NEGLIGÊNCIA OU DE OUTRO MODO), RESULTANTE DO USO DESTES SOFTWARES, MESMO SE ADVERTIDA DA RESPONSABILIDADE DE TAL DANO.

A NetApp reserva-se o direito de alterar quaisquer produtos descritos neste documento, a qualquer momento e sem aviso. A NetApp não assume nenhuma responsabilidade nem obrigação decorrentes do uso dos produtos descritos neste documento, exceto conforme expressamente acordado por escrito pela NetApp. O uso ou a compra deste produto não representam uma licença sob quaisquer direitos de patente, direitos de marca comercial ou quaisquer outros direitos de propriedade intelectual da NetApp.

O produto descrito neste manual pode estar protegido por uma ou mais patentes dos EUA, patentes estrangeiras ou pedidos pendentes.

LEGENDA DE DIREITOS LIMITADOS: o uso, a duplicação ou a divulgação pelo governo estão sujeitos a restrições conforme estabelecido no subparágrafo (b)(3) dos Direitos em Dados Técnicos - Itens Não Comerciais no DFARS 252.227-7013 (fevereiro de 2014) e no FAR 52.227- 19 (dezembro de 2007).

Os dados aqui contidos pertencem a um produto comercial e/ou serviço comercial (conforme definido no FAR 2.101) e são de propriedade da NetApp, Inc. Todos os dados técnicos e software de computador da NetApp fornecidos sob este Contrato são de natureza comercial e desenvolvidos exclusivamente com despesas privadas. O Governo dos EUA tem uma licença mundial limitada, irrevogável, não exclusiva, intransferível e não sublicenciável para usar os Dados que estão relacionados apenas com o suporte e para cumprir os contratos governamentais desse país que determinam o fornecimento de tais Dados. Salvo disposição em contrário no presente documento, não é permitido usar, divulgar, reproduzir, modificar, executar ou exibir os dados sem a aprovação prévia por escrito da NetApp, Inc. Os direitos de licença pertencentes ao governo dos Estados Unidos para o Departamento de Defesa estão limitados aos direitos identificados na cláusula 252.227-7015(b) (fevereiro de 2014) do DFARS.

Informações sobre marcas comerciais

NETAPP, o logotipo NETAPP e as marcas listadas em <http://www.netapp.com/TM> são marcas comerciais da NetApp, Inc. Outros nomes de produtos e empresas podem ser marcas comerciais de seus respectivos proprietários.