



使用 Oracle HA 部署 Google Cloud NetApp Volumes

NetApp database solutions

NetApp
June 25, 2026

目錄

使用 Oracle HA 部署 Google Cloud NetApp Volumes	1
為 Google Cloud NetApp Volumes 佈建 Google Compute Engine 執行個體	1
步驟 1：建立 VM	1
步驟 2：設定 TCP 1521 連接埠的 VPC 防火牆	2
步驟 3：設定主機名稱、DNS 和 /etc/hosts	2
步驟 4：僅在資料庫主機上準備作業系統	3
步驟 5：取得 iSCSI 發起程序名稱 IQN	4
接下來呢？	5
為 Oracle Database 26ai 佈建 Google Cloud NetApp Volumes iSCSI 儲存設備	5
步驟 1：建立 GCNV iSCSI 集區	5
步驟 2：建立主機群組	5
步驟 3：建立 GCNV iSCSI 磁碟區	6
步驟 4：設定 iSCSI 和多路徑	7
步驟 5：分割 ASM 裝置	9
步驟 6：格式化並掛載 /u01	10
接下來呢？	11
在 Google Cloud NetApp Volumes 上安裝 Oracle Grid Infrastructure 和 Oracle Database 26ai	11
步驟 1：在每個資料庫主機上安裝 Grid Infrastructure	11
步驟 2：在每個資料庫主機上安裝 Oracle Database	14
接下來呢？	16
在 Google Cloud NetApp Volumes 上建立 Oracle 主資料庫	16
接下來呢？	18
使用 Google Cloud NetApp Volumes 儲存層種子建立 Oracle 備用資料庫	18
步驟 1：設定監聽器和 Data Guard 參數	18
步驟 2：準備備用 pfile 和 NOMOUNT	19
步驟 3：使用 GCNV 初始化待命儲存設備	21
步驟 4：向 Oracle Restart 註冊備用資料庫	24
接下來呢？	26
完成 Google Cloud NetApp Volumes 上 Data Guard 的備用資料庫配置	26
步驟 1：建立備用重做日誌檔案	27
步驟 2：啟用回溯功能並啟動復原	27
步驟 3：啟用重做傳送	28
步驟 4：驗證 Data Guard 狀態	29
接下來呢？	30
在 Google Cloud NetApp Volumes 上為 Oracle Database 26ai 設定 Data Guard Broker 和快速啟動容錯移轉	30
步驟 1：啟用 Data Guard Broker	30
步驟 2：確認 FSFO 的 Flashback	31
步驟 3：設定並啟用 FSFO	31

步驟 4：在 Observer 上安裝 Instant Client	32
步驟 5：將 Observer 作為 systemd 服務執行	33
步驟 6：測試 FSFO	36
接下來呢？	37

使用 Oracle HA 部署 Google Cloud NetApp Volumes

為 Google Cloud NetApp Volumes 佈建 Google Compute Engine 執行個體

配置 Google Compute Engine 虛擬機，以在 Google Cloud NetApp Volumes iSCSI 儲存上託管 Oracle Database 26ai。此程序涵蓋建立主資料庫主機、備用資料庫主機和快速啟動故障轉移觀察器虛擬機器、配置 Oracle Net 的 VPC 防火牆規則、設定主機名稱解析、準備作業系統以及取得用於 GCNV 儲存配置的 iSCSI 發起程序名稱。

步驟 1：建立 VM

在同一區域的不同可用區域建立三個 Google Compute Engine VM，以實現區域故障隔離。使用 Cloud Console、`gcloud` Terraform 或您常用的佈建工作流程。

1. 建立三個符合下表所示規格的虛擬機器。

出於整體擁有成本和永續性的考慮，優先選擇碳排放量較低的地區，同時滿足延遲和合規性要求（例如 us-west1 vs us-centrall）：

VM	區域	機器類型	開機磁碟	網路	目的
oracdb1	us-west1-a	n4-highmem-8（範例）或 c4-standard-*	OL 10、50 GB Hyperdisk Balanced（僅作業系統）	oracle-vpc / oracle-subnet、gVNIC	主要資料庫
oracdb2	us-west1-b	與主要相同	OL 10、50 GB Hyperdisk Balanced（僅作業系統）	相同	備用資料庫
oradg-obs	us-west1-c	e2-medium	OL 10、20 GB Hyperdisk Balanced	相同	FSFO Observer（僅限 Instant Client）

當延遲或輸出流量（>~200 GiB/月）至關重要時，請使用 Premium 網路層；在開發/測試環境中，為了降低 TCO，請使用 Standard 網路層。

2. 啟用受保護虛擬機器功能並驗證啟動磁碟配置：

在所有三台虛擬機器上啟用 **Secure Boot**、**vTPM** 和 **Integrity Monitoring**。

啟動磁碟僅包含作業系統 /u01，Grid/DB 主目錄、暫存環境和所有 ASM 資料都使用 GCNV iSCSI 磁碟區（請參閱 [配置 GCNV iSCSI 磁碟區](#)）

請勿為 /u01 單獨掛載 GCE 資料磁碟。

步驟 2：設定 TCP 1521 連接埠的 VPC 防火牆

建立 VPC 防火牆規則，允許所有三個虛擬機器之間透過 TCP/1521 連接埠進行 Oracle Net 重做傳輸和 Observer 連線。缺少規則會導致 Data Guard 複製中斷。

1. 建立一條 VPC 防火牆入站規則，允許所有三個 VM 內部 IP 位址之間的 TCP/1521 通訊。可以使用 VPC 防火牆規則或 Firewall Policies，並採用相同的允許清單：

Cloud Console：VPC 網路 → 防火牆 → 建立規則 `allow-oracle-net-dbhosts` on `oracle-vpc` — Ingress、Allow、sources = 三個 /32 IP、TCP 1521。如有需要，鏡像 egress。

2. 驗證每台虛擬機器的連線性，以確認防火牆規則已生效：

```
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
```

連接埠	預期	意義
22	已連接	SSH 路徑有效
1521	連線被拒絕	防火牆已開啟；Grid 監聽器在以下期間啟動 步驟 1：在每個資料庫主機上安裝 Oracle Grid Infrastructure (Oracle Restart)
任一	逾時	修復防火牆或路由

從所有三個 VM 向每個對等 IP 位址執行命令。

步驟 3：設定主機名稱、DNS 和 /etc/hosts

在所有三個 VM 上設定主機名稱和 DNS 解析，使 Oracle Net、Data Guard Broker 和 Observer 能夠正常執行正向和反向名稱解析。

1. 設定主機名稱並在所有三台主機上新增 /etc/hosts 條目。替換 GCE 內部 IP 位址（可在「Compute Engine → VM 執行個體」清單的「內部 IP」欄位中檢視）：

```
# 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. 驗證每個主機的名稱解析：

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

步驟 4：僅在資料庫主機上準備作業系統

在 `oracdb1` 和 `oracdb2` 上為 Oracle Database 26ai 準備作業系統，包括安裝預先安裝套件、建立使用者和群組、安裝 iSCSI 和多路徑軟體套件，以及設定 iSCSI 發起程式。觀察者設定已在 [步驟 4：在 Observer 主機上安裝 Oracle Instant Client](#) 中說明。



前提條件：出站 HTTPS 到 yum.oracle.com (Cloud NAT 或私有子網路上的內部鏡像)。

1. 安裝 Oracle Database 預先安裝套件，建立 `grid` 使用者和 ASM 群組，並將 `oracle` 使用者新增至 ASM 群組：

```
# 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. 安裝 iSCSI、多路徑和 JDK 套件，然後驗證 THP 和時間同步：

```
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. 設定 SELinux、防火牆和 iSCSI 啟動器設定，然後重新啟動：



安全態勢 (OL 10)： 下列指令將 SELinux 設定為寬容模式並停用 `firewalld`。這只是最低限度的實驗室態勢。如需強化的 SELinux 和防火牆組態，請參閱貴組織的安全基準。

```
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
```

步驟 5：取得 iSCSI 發起程序名稱 IQN

重新啟動後，從每個資料庫主機擷取 iSCSI 發起程式名稱 (IQN)。您將使用這些 IQN 在 [步驟 2：建立主機群組](#) 中建立 GCNV 主機群組。

1. 從 `oracdb1` 擷取 IQN 並記錄：

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

2. 重複上述步驟 `oracdb2` 並記錄其 IQN。每個主機使用一個主機群組，這樣單一主機的重新啟動或 IQN 重新產生就不會影響其他主機的 GCNV iSCSI 磁碟區可見性：



複製虛擬機器： 如果兩台主機共用同一個 IQN，則重新產生 `oracdb2`（停止 `iscsi`、清除 `/var/lib/iscsi/nodes/*`、新建 `InitiatorName` 在 `/etc/iscsi/initiatorname.iscsi`、重新啟動 `iscsid`）。

接下來呢？

若要為 Oracle 二進位和 ASM 磁碟群組提供共用儲存，請前往 [配置 Google Cloud NetApp Volumes iSCSI 集區、主機群組和磁碟區](#)。

為 Oracle Database 26ai 佈建 Google Cloud NetApp Volumes iSCSI 儲存設備

在 Google Compute Engine 上為 Oracle Database 26ai 高可用性佈建 Google Cloud NetApp Volumes iSCSI 區塊儲存。本流程涵蓋建立 GCNV Flex 統一儲存池、定義主機群組、為每個資料庫主機建立 iSCSI 磁碟區、設定 Linux iSCSI 和多路徑、對 ASM 後端裝置進行分割，以及掛載 `u01` 檔案系統。

步驟 1：建立 GCNV iSCSI 集區

建立兩個 Flex Unified 儲存池，每個資料庫區域各一個，為主要主機和備用主機提供 iSCSI 磁碟區。每個資料庫主機使用其本機區域儲存池中的磁碟區。

1. 使用 Cloud Console 建立兩個儲存池。使用下表中的規格，並對每個區域重複建立流程：

資源池名稱	區域	使用者
oracle-pool-a	us-west1-a	oracdb1 (主要)
oracle-pool-b	us-west1-b	oracdb2 (待命)

NetApp Volumes → **Storage pools** → **Create** 為每個池：

- 服務等級：Flex (非 Premium)
 - *類型：*統一
 - *Zone：*符合資料庫 VM 區域(us-west1-a / us-west1-b)
 - **PSA**: 已連接到 oracle-vpc
 - 容量：根據工作負載進行調整；當重做、備份或還原超出預設餘裕時，使用自訂佈建的處理量 / IOPS (每個集區最高可達 5120 MiB/s 或 160K IOPS，依產品限制而定)
2. 請等待兩個儲存池都達到 `READY` 狀態後再繼續。根據資料庫規模調整儲存池大小 ([步驟 3：建立 GCNV iSCSI 磁碟區](#)中的大小僅為範例)：



預設模式 (本指南)：Flex Unified 資源池使用預設模式(--mode=default)。使用 Cloud Console 或 `gcloud netapp` 建立資源池和 iSCSI 磁碟區。磁碟區複寫、快照和複本使用 Google Cloud API ([步驟 3：GCNV 待命初始化](#))。

步驟 2：建立主機群組

為每個資料庫主機建立一個主機群組，以便每個虛擬機器只能看到自己的磁碟區。主主機和備用主機不得共用 GCNV iSCSI 磁碟區，以維持儲存的獨立性。

1. 使用 Cloud Console 為 `oracdb1` 建立主機群組：

NetApp 磁碟區 → 主機群組 → 建立

- 名稱：oracdb1-hg
- 地區：us-west1
- 類型：iSCSI 啟動器
- **OS 類型**: Linux
- *主機：*從 `oracdb1` 貼上 IQN (`/etc/iscsi/initiatorname.iscsi` 的值)
- 描述："Oracle 主要主機 oracdb1"
- 建立

2. 針對 oracdb2 重複此程序，並使用名稱 oracdb2-hg 和 oracdb2 的 IQN。Observer 主機不需要任何 GCNV 資源。

步驟 3：建立 GCNV iSCSI 磁碟區

為每個資料庫主機建立五個 GCNV iSCSI 磁碟區：一個用於 /u01，四個用於 ASM 後端設備。每個主機的磁碟區必須建立在其本機區域的儲存池中，並與其對應的主機群組關聯。

1. 為 oracdb1 在 oracle-pool-a 中建立五個磁碟區，主機群組為 oracdb1-hg。請使用下表中的規格：

GCNV iSCSI 磁碟區	大小	使用	多重路徑別名
ora_<host>_u01	100 GiB	/u01 GCNV iSCSI 磁碟區 — Grid/Oracle 主目錄、暫存區	/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 (條紋)	/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

磁碟區名稱：僅使用字母、數字和底線（不使用連字號）。



最小佈局（僅用於驗證）：每個主機兩個 LUN (*_data, *_reco) with arch_01p1 → +RECO 和 arch_01p2 → +FRA 對於實驗室來說是可以接受的；生產環境每個 [步驟 3：建立 GCNV iSCSI 磁碟區](#) 使用五個磁碟區。

2. 使用相同的規格，在 oracle-pool-b 中為 oracdb2 建立五個磁碟區，並使用主機群組 oracdb2-hg。針對每個集區，使用 **NetApp Volumes** → **Volumes** → **Create** — iSCSI、正確的集區與主機群組、Linux。記錄以下資訊：
 - iSCSI 入口網站 IP 位址 → <ISCSI_PORTAL_1>, <ISCSI_PORTAL_2> (主池入口網站位於 oracdb1；備用池入口網站位於 oracdb2——它們可能不同)

- 從雲端主控台取得磁碟區序號 — 與主機發現的 WWID 一起使用 [步驟 4：為 GCNV iSCSI 磁碟區設定 Linux iSCSI 和多重路徑](#)

步驟 4：設定 iSCSI 和多路徑

在每個資料庫主機上設定 iSCSI 和 device-mapper-multipath，以便透過兩個儲存入口網站 IP 存取 GCNV 磁碟區。在 `oracdb1` 上使用主儲存池的入口網站 IP 執行這些步驟，然後在 `oracdb2` 上使用備用儲存池的入口網站 IP 重複這些步驟。如果主機輸出受到限制，請允許每個資料庫 VM 透過 TCP/3260 存取其 GCNV iSCSI 入口網站 IP（此外還允許來自 [步驟 2：VPC 防火牆](#) — 在所有三個區域中允許 TCP/1521 的 VM 間 TCP/1521）。

1. 探索目標、登入並持續節點啟動：

```
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
```

重新開機後，請在啟動 Oracle 之前重新檢查：

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

2. Configure device-mapper-multipath 使用預設值和黑名單規則進行設定：

```

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. 將主機發現的 WWID 別名加入 /etc/multipath.conf (不要猜測 — multipath.conf 不會展開 shell 變數)。發現 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

```

將該主機的具體別名附加到 /etc/multipath.conf，然後 `sudo systemctl restart multipathd`。

開啟 `oracdb1`，附加：

```

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  }
}

```

在 `oracdb2` 上，使用相同的模式搭配 `ora_oracdb2_*` 別名，然後：

```

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

```

步驟 5：分割 ASM 裝置

將四個 ASM 後端裝置（不包括 u01）分別劃分為一個 GPT 分區，供 ASM 使用，然後設定 udev 規則以實現 grid 所有權。在每個資料庫主機上執行這些步驟。

1. 使用 GPT 對四個 ASM 後端裝置進行分割，並驗證分割區：

```

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. 設定 udev 規則以指派網格所有權並觸發變更：

```

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

```

步驟 6：格式化並掛載 /u01

使用 XFS 格式化 ora_<host>_u01 GCNV 磁碟區，並使用 UUID 將其持久掛載於 /etc/fstab。此 '/u01' 檔案系統包含 Grid 主目錄、Oracle 主目錄和暫存檔案。

1. 使用 XFS 格式化多路徑裝置並取得其 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. 新增基於 UUID 的掛載條目至 '/etc/fstab' 並掛載檔案系統：

```

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. 為 Grid 和 Oracle 軟體建立具有正確所有權的目錄結構：

```
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
```

重新開機一次並確認 /u01 掛載在[安裝 Oracle 軟體](#) 之前。

接下來呢？

若要在已準備好的主機上安裝 Oracle Grid Infrastructure 和資料庫二進位檔案，請前往 [安裝 Oracle Grid Infrastructure](#) 和 [Oracle 資料庫軟體](#)（兩台主機均需執行）。

在 Google Cloud NetApp Volumes 上安裝 Oracle Grid Infrastructure 和 Oracle Database 26ai

在每個資料庫主機上，使用 Google Cloud NetApp Volumes iSCSI 儲存安裝 Oracle Grid Infrastructure、Oracle Restart 和 ASM，然後安裝 Oracle Database 26ai 軟體。此程序包括暫存 Oracle GoldImages、使用回應檔案執行靜默安裝、在 GCNV 磁碟區上建立 ASM 磁碟群組，以及在建立資料庫之前，使用相同的 Oracle 軟體準備主要主機和備用主機。

步驟 1：在每個資料庫主機上安裝 Grid Infrastructure

在每個資料庫主機上安裝 Oracle Grid Infrastructure GoldImage，以啟用 Oracle Restart 和 ASM。兩個主機都需要各自的 Grid 主目錄、ASM 執行個體和磁碟群組；Data Guard 透過 Oracle Net 複寫資料，而非透過共用儲存設備。完成所有步驟 `oracdb1` 後，再在 `oracdb2` 上重複執行。

1. 將 Oracle GoldImages、Release Update 和 OPatch 二進位檔案暫存到 /u01/stage：

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

2. 將 Grid GoldImage 就地解壓縮到目標 Grid 主目錄。26ai GoldImage 的安裝方式是直接解壓縮到目標目錄：

```
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
```

如果 Grid GoldImage 的版本低於目標 RU，請在安裝過程中使用 `gridSetup.sh -applyRU` 流程修補 Grid 主

目錄，或使用已捆綁 RU 的 GoldImage。請確保 Grid 和資料庫主目錄的修補程式層級保持一致。

3. 在每台主機上建置 gridSetup`回應檔案` /tmp/grid.rsp。替換主機名稱並使用強密碼：

```
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=WelcomeOracle!
asmsnmpPassword=WelcomeOracle!
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. Run `gridSetup.sh`以靜默模式執行，以複製二進位檔案並準備組態。預期`Successfully Setup Software with warning(s)`退出代碼為 6（警告）或 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. 以 root 使用者身分執行 orainstRoot.sh 和 root.sh。root.sh 腳本會建立 crsctl、srvctl 和 asmcmd 包裝器，並啟動 OHAS：

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

6. 執行 `gridSetup.sh -executeConfigTools` 以針對回應檔案執行設定助手 (NETCA、ASMCA、CVU) 。這將建立 ASM 執行個體和 `+DATA` 磁碟群組。NETCA / ASMCA / CVU 完成後的預期結果 `Successfully Configured Software.` :

```
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. 使用 `asmca` 建立 `+RECO` 與 `+FRA` 磁碟群組。單次安裝僅會建立 `+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. 驗證 ASM 磁碟組和 Oracle Restart 資源狀態 :

```

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. 在 `oracdb2` 上重複上述步驟。`HOST=\$(hostname -s)` 中的模式會在步驟 3 和 4 和步驟 7 中自動選擇該主機的 GCNV iSCSI 裝置。

使用相同的 ASM 磁碟群組名稱 — Data Guard 透過 Oracle Net 進行複寫，而非透過儲存設備。

步驟 2：在每個資料庫主機上安裝 Oracle Database

在每個資料庫主機上，使用靜默、僅軟體安裝的方式安裝 Oracle Database 26ai 軟體主目錄，並套用最新的版本更新。完成 `oracdb1` 上的所有步驟，再重複執行 `oracdb2` 上的步驟。

1. 將資料庫主目錄、最新的 OPatch 和 RU 修補程式解壓縮到各自的目錄中。有關 RU 目錄結構和 `applyRU` 路徑，請參閱 Oracle 文件：

```

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. 編寫安裝回應檔，並在套用 RU 的情況下執行靜默僅軟體安裝。在 OL 8/9 上，從 `runInstaller` 行中省略 `applyOneOffs`：

```

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

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

3. 執行安裝後的 root 指令碼：

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

4. 在每個資料庫主機上設定 Oracle 環境。使用 ORACLE_SID=orcl 於 `oracdb1`，以及 ORACLE_SID=orcls 於 `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
```

待命資料庫是在[建立備用資料庫](#)中建立。

接下來呢？

若要為 HA 部署建立生產主要執行個體，請前往 [建立 Oracle 主資料庫](#) 的 oracdb1。

在 Google Cloud NetApp Volumes 上建立 Oracle 主資料庫

使用 Oracle Database Configuration Assistant 以靜默模式在 Google Cloud NetApp Volumes iSCSI 儲存上建立 Oracle 主要資料庫。此程序涵蓋執行 `dbca` 以在 GCNV 支援的 ASM 磁碟群組上建立容器資料庫和可插拔資料庫、設定歸檔日誌目的地，以及在啟用 Data Guard 後新增基於角色的應用程式服務以實現透明容錯移轉。

步驟

在 oracdb1 上建立 Oracle 容器資料庫和可插拔資料庫，以靜默模式使用 `dbca`，設定歸檔日誌目的地，驗證 Oracle Restart 註冊，並新增基於角色的應用程式服務以實現透明用戶端容錯移轉。

1. 以靜默模式執行 dbca，在 ASM 磁碟群組上建立 CDB 和 PDB：

```
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. 將歸檔日誌指向 `+RECO` 並開啟和儲存可插拔資料庫狀態。備用資料庫在 [步驟 2：備用 init.ora、pfile 和 NOMOUNT](#) 中使用相符的歸檔日誌設定：

```

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. 確認資料庫正在 Oracle Restart 下運作：

```

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. 建立角色型應用程式服務，讓應用程式透過 `orclapp` 連接，並在啟用 Data Guard 時實現透明的故障移轉：

```

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
'
```

啟用 Data Guard Broker 後，`orclapp` 僅在 PRIMARY 上執行。跨 ASM 磁碟群組多工控制檔案，並根據工作負載調整記憶體大小。

接下來呢？

若要建立備用保護和容錯移轉整備，請前往[建立 Oracle 備用資料庫](#)（位於 oracdb2）。

使用 Google Cloud NetApp Volumes 儲存層種子建立 Oracle 備用資料庫

使用 Google Cloud NetApp Volumes 儲存層複寫、快照或複製來建立 Oracle 實體備用資料庫，與傳統的 RMAN 方法相比，可加速備用資料庫初始化。此程序涵蓋設定接聽程式、建立備用 pfile、使用 GCNV 複寫植入備用磁碟區、完成 Oracle 執行個體，以及向 Oracle Restart 註冊備用資料庫。所有 HA 層級皆需完成這些步驟。對於 **Prod HA (Data Guard + FSFO)** 層級，在設定 [Data Guard Broker](#)、[Fast-Start Failover](#) 與 [Observer](#) 之前，請先繼續進行 [Data Guard 最終化](#)。

步驟 1：設定監聽器和 Data Guard 參數

在兩個資料庫主機上設定監聽器以支援 Data Guard 連線，包括 `_DGMGRL` 代理程式所需的服務。在主要資料庫上設定密碼檔案並設定歸檔日誌參數。

1. 配置主監聽器並驗證環境於 oracdb1：

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

2. 在 `oracdb2` 上設定備用監聽器，以包含 `orcls` 和 `orcls_DGMGRL` 服務：

```
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. 在兩台主機上透過 Oracle Restart 重新啟動監聽器，並驗證 `_DGMGRL` 服務是否已註冊：

```

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 必須列出 <SID> 和 <SID>_DGMGRL。

步驟 2：準備備用 pfile 和 NOMOUNT

透過從主資料庫複製密碼檔案、建立包含 Data Guard 參數的最小 init.ora pfile 以及以 NOMOUNT 模式啟動執行個體來準備備用資料庫執行個體。

1. 使用 IAP 和 `gcloud compute scp` 將主密碼檔案複製到備用主機：

```

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. 從主資料庫查詢 compatible 參數值：

```

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

3. 在 oracdb2 上建立備用 pfile，設定密碼檔案的所有權，並以 NOMOUNT 模式啟動執行個體。將上一步中的 compatible 值替換為 <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
'

```

備用實例現在處於 NOMOUNT 模式，沒有資料檔案，直到 [步驟 3：使用 GCNV 初始化待命儲存設備](#)。

步驟 3：使用 GCNV 初始化待命儲存設備

在 `oracle-pool-b` 中初始化備用磁碟區，將其附加至 `oracdb2`，掛載 ASM 磁碟群組，並將備用執行個體最終設為 MOUNT 狀態。

使用 GCNV 複寫進行生產環境種子建立，並使用快照種子建立進行一次性實驗室工作流程。

選擇植入路徑

根據您的環境和復原要求選擇備用種子方法。

- *推薦用於生產環境：*使用 [複製路徑：建立和同步複製](#) 和 [複製路徑：切換並附加待命磁碟區](#) 中的複製路徑。
- 實驗室替代方案：使用 [備用路徑：從快照取得種子](#)。

所有路徑在 [掛載備用 ASM 磁碟組](#) 和 [完成備用執行個體](#) 處重新匯合。

檢查先決條件

在建立備用磁碟區之前，請確認以下先決條件。

- `gcloud netapp` 支援磁碟區複寫。
- 兩個位於不同位置的 預設模式 池(`oracle-pool-a`, `oracle-pool-b`)。
- 連接到主池的來源磁碟區 `oracdb1-hg`；由複寫建立的目的地磁碟區。
- 從 Cloud Shell 或工作站執行複寫，而不是從資料庫 VM 執行。
- 在 `oracdb2` 上，從 [步驟 4](#)、[步驟 5](#) 和 [步驟 6](#) 完成 iSCSI 和 ASM 主機設定。

```
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"
```

- 如有需要，請建立備用池：

```
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>
```

建立和同步複寫

建立從主磁碟區到備用磁碟區的複製關係，然後等待初始同步完成。

```

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"

```

+

等待 mirrorState 為 `MIRRORED`，且每次複製的初始同步完成。

切換並連接待命磁碟區

使主要主機靜默，在最後一次同步後停止複製，並將目標磁碟區附加到備用主機群組。

在主節點上，靜默寫入並擷取恢復中繼資料：

```

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

```

允許進行最後一次複製週期，然後停止複製：

```

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

```

將目標磁碟區附加到 oracdb2-hg（複製的 LUN 可以保留來源名稱）：

```

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"

```

將備用控制檔複製到 oracdb2，然後結束主要資料庫上的備份模式：

```
ALTER DATABASE END BACKUP;
```

從快照建立種子

當不需要持續複製時，請使用此路徑進行一次性實驗室植入。

對於一次性實驗室種子，請建立來源快照，並從該快照在 oracle-pool-b（雲端控制台或 API）中建立備用磁碟區。將建立的磁碟區附加到 oracdb2-hg，然後繼續執行 [掛載備用 ASM 磁碟組](#)。

掛載備用 ASM 磁碟組

在備用主機上，探索附加的儲存路徑，並在資料庫復原之前掛載 ASM 磁碟群組。

在 oracdb2，登入備用池 iSCSI 入口網站並重新掃描多路徑裝置。如果 ASM 磁碟標頭與實驗室工作流程中的主要命名相符，請使用主要樣式別名（例如 ora_oracdb1_data_01，ora_oracdb1_arch_01），設定 asm_diskstring='/dev/mapper/ora_oracdb1_*p*'，並確認分割區所有權為 grid:asmadmin，然後掛載磁碟群組：

```

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

```

完成備用執行個體

還原備用控制檔案，恢復至擷取的 SCN，轉換為實體備用，並開始受管理的恢復。

```
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;
```

此時，備用系統應該已經 `PHYSICAL STANDBY` 且 `MOUNTED` 並已啟動託管復原。

各層級後續步驟：

- **Prod HA (no Data Guard):** 直接繼續 [步驟 4：向 Oracle Restart 註冊備用資料庫](#)。
- **Prod HA (Data Guard + FSFO):** 繼續執行 [步驟 4：向 Oracle Restart 註冊備用資料庫](#)，然後繼續執行 [Data Guard 最終化步驟](#)。

步驟 4：向 Oracle Restart 註冊備用資料庫

將備用資料庫註冊到 Oracle Restart，以便重新啟動時自動復原 ASM 磁碟群組、掛載備用資料庫並重新啟動託管復原。同時，將應用程式服務新增至兩個資料庫資源。

1. 從備用資料庫擷取 spfile 位置，並將其註冊到 Oracle Restart on oracdb2。替換查詢中的 `<STANDBY_SPFILE_PATH>`（通常位於 `+DATA` 下方）：

```

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> \
  -pfile /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. 驗證並更新主資料庫資源 oracdb1，使其包含所有 ASM 磁碟組相依性：

```

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. 將應用程式服務新增至備用資料庫資源 (orcls on oracdb2)。在兩端使用 role PRIMARY，以便 `orclapp` 在切換後仍然可用：

```

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. 驗證備用資料庫資源於 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
'
```

接下來呢？

特定等級：

- **Prod HA (無 Data Guard)**：為了維護基於儲存複製的復原目標，備用資料庫初始化完成，並且備用資料庫已向 Oracle Restart 註冊為備份執行個體。
- **Prod HA (Data Guard + FSFO)**: 若要啟用代理管理的切換和快速啟動故障轉移，請繼續執行[完成 Data Guard 的備用資料庫配置](#)。

完成 Google Cloud NetApp Volumes 上 Data Guard 的備用資料庫配置

透過建立備用重做日誌檔案、啟用閃回資料庫、啟動重做傳輸及驗證 Data Guard 狀態，完成 Google Cloud NetApp Volumes 上 Oracle Data Guard 的備用資料庫。

層級特定：*此程序僅適用於 ***Prod HA (Data Guard + FSFO)** 層級。

步驟 1：建立備用重做日誌檔案

在兩個資料庫主機上建立備用重做日誌檔案，以支援 Fast-Start Failover。其大小必須大於或等於最大的主要線上重做日誌檔案，且數量應等於（每個執行緒的線上群組數）+ 1。完成 GCNV 種子部署後，刪除備用資料庫上的備用重做日誌檔案並重新建立，以修復複製路徑。

1. 在主資料庫上建立備用重做日誌檔案 (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. 在 GCNV 種子部署後，刪除並重新建立備用資料庫上的備用重做日誌檔案 (orcls)。+DATA/ORCL/... 下的複製路徑會導致 ORA-19527 / ORA-16086，直到重建為止：

```
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;
```

步驟 2：啟用回溯功能並啟動復原

在備用資料庫上啟用 Flashback 資料庫，以支援容錯移轉後的自動恢復，然後以即時套用方式啟動受管理的恢復。必須在啟動受管理的恢復之前啟用 Flashback，因為在 MRP 執行期間無法啟用 Flashback。

1. 關閉備用資料庫，以 MOUNT 模式重新啟動，並在 `oracdb2` 上啟用閃回資料庫：

```
# 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. 啟動具有即時套用功能的託管復原：

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

USING CURRENT LOGFILE 可實現即時套用（重做會在抵達 SRL 時套用）。

步驟 3：啟用重做傳送

透過啟用 LOG_ARCHIVE_DEST_STATE_2 來啟用從主要端到備用端的 redo 傳輸，這在備用端初始化程序的步驟 2 中被刻意設定為 DEFER，以抑制在建立備用端期間的 ORA-12154 錯誤。

1. 切換 LOG_ARCHIVE_DEST_STATE_2 到 ENABLE 並強制執行日誌切換以啟動重做傳送：

```
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. 驗證重做傳送是否正常運作：

```
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.
```

如果 dest_2 顯示 ORA-12154，則重新啟動主伺服器。之後步驟 1：在兩個資料庫上啟用 broker，透過 DGMGRL 管理傳輸。

步驟 4：驗證 Data Guard 狀態

確認主資料庫處於 READ WRITE 模式，且備用資料庫已掛載並啟用託管恢復，套用重做日誌。

1. 驗證主資料庫角色和開啟模式（位於 oracdb1）：

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

2. 驗證備用資料庫的角色、開放模式和託管復原狀態（位於 oracdb2）：

```
gcloud compute ssh oracdb2 --tunnel-through-iap --zone=us-west1-b  
  
sudo -u oracle bash <<'BASH'  
. ~/.bash_profile  
export ORACLE_SID=orcl  
  
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
```

待機預期：PHYSICAL STANDBY | MOUNTED；MRP0 帶有 APPLYING_LOG。

3. 如果備用復原進程報告 `MOUNTED` 但 apply 未執行，請在 `oracdb2` 上重新啟動託管復原：

```
sudo -u oracle bash -c '  
. ~/.bash_profile  
export ORACLE_SID=orcl  
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'
```

接下來呢？

若要啟動自動角色管理和故障轉移保護，請繼續執行設定 [Oracle Data Guard Broker](#)、[Fast-Start Failover](#) 和 [Observer](#)。

在 Google Cloud NetApp Volumes 上為 Oracle Database 26ai 設定 Data Guard Broker 和快速啟動容錯移轉

配置 Oracle Data Guard Broker 和快速啟動故障轉移，並配備專用觀察器，以啟用 Google Cloud NetApp Volumes 上 Oracle Database 26ai 的自動角色轉換。

層級特定：*此程序僅適用於 *Prod HA (Data Guard + FSFO) 層。

此流程涵蓋在兩個資料庫上啟用代理、建立 Data Guard 組態、啟用具有 MaxAvailability 保護模式的 FSFO、在 Observer 主機上安裝 Oracle Instant Client、使用基於錢包的憑證將 Observer 作為 systemd 服務啟動，以及測試切換和容錯移轉。之後 ENABLE CONFIGURATION，透過 **DGMGRL**（而非臨時 LOG_ARCHIVE_DEST_* SQL）管理傳輸和角色。

步驟 1：啟用 Data Guard Broker

在兩個資料庫主機上啟用 Data Guard Broker，並建立將主資料庫和備用資料庫在統一管理下連結的 Broker 組態。

1. 在主資料庫主機和備用資料庫主機上設定 dg_broker_start=TRUE：

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

2. 在主伺服器上，使用作業系統驗證連線至 DGMGRL 並建立代理程式組態：



僅在 Observer 主機上使用 dgmgrrl /@orcl，且需在自動登入錢包存在後使用。請勿在 `dgmgrrl` 命令列中輸入密碼。

```
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  
dgmgrrl /  
'
```

```
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. 驗證組態 — 在 [步驟 3：設定 FSFO 內容並啟用](#) 之前修正任何 WARNING 或非 NULL ERROR：

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

步驟 2：確認 FSFO 的 Flashback

確認兩台主機上都已啟用閃回資料庫。閃回功能是 FSFO 自動恢復所必需的，可讓原主節點在容錯移轉後自動重新加入組態，成為備用節點。

1. 確認 flashback_on 已在兩台資料庫主機上啟用 YES：

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

2. 僅在主要裝置上，如果尚未設定 flashback 保留：

```
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'
```

步驟 3：設定並啟用 FSFO

設定 SYNC 重做傳輸，配置 MaxAvailability 保護模式，在每個資料庫上定義 FSFO 目標，並啟用 Fast-Start Failover。

1. 將兩個資料庫的重做傳輸模式設定為 SYNC，並將保護模式提升至 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. 設定 FSFO 目標，使每個資料庫都將另一個資料庫指定為其容錯移轉目標，然後設定臨界值和自動重新恢復行為：

```
-- 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. 啟用 Fast-Start Failover 並確認組態：

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

步驟 4：在 Observer 上安裝 Instant Client

在專用的 Observer VM 上安裝 Oracle Instant Client (oradg-obs)，建立專用的 oracle OS 使用者，並設定 Oracle Net 環境，使 Observer 能夠透過 TCP/1521 連線至兩個資料庫成員。

1. 在觀察者主機上安裝 Oracle Instant Client 軟體套件 (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. 建立一個專用的 oracle OS 使用者，將擁有錢包和 systemd 單元：

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

3. 配置 Oracle Net 環境，並建立 `tnsnames.ora`，其中包含兩個資料庫主機的條目：

```
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
```

步驟 5：將 Observer 作為 `systemd` 服務執行

建立一個包含兩個資料庫成員憑證的自動登入錢包，然後將 Observer 設定並啟動為 `systemd` 服務，使其在重新開機後仍能持續運作並自動重新連線至設定。

將專用 Data Guard 管理帳戶（例如 `SYSDG`）的憑證儲存在錢包中，而不是 `sys`。憑證絕不能出現在 ``dgmgrl`` 命令列中，因為命令列對 ``ps`` 和 ``journalctl`` 可見；一律在 Observer 上使用 ``/@<tns_alias>`` 進行連線。

1. 建立加密錢包，並為資料庫中的兩個成員填寫憑證：

```

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

```

2. 產生自動登入電子錢包 (wallet.sso)，使 Observer systemd 服務無須提示輸入密碼即可啟動。如果 wallet.sso 在執行 mkstore 後遺失，請使用來自 Instant Client 工具套件或資料庫主目錄的 orapki 來建立它，然後重新新增儲存的認證：

```
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.pl2
```

3. 建立 systemd 單元，啟用服務，並驗證 Observer 是否已連線：

```
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

```

觀察員必須從主設備讀取 CONNECTED (a DISCONNECTED 觀察員會靜默暫停 FSFO) :

```

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

```

步驟 6：測試 FSFO

使用 `VALIDATE DATABASE` 驗證 Data Guard 配置，然後執行計劃內切換，並在測試視窗中執行計劃外 VM 重設故障轉移，以確認 FSFO 端對端運作正常。

1. 測試計劃內的切換並恢復原始拓撲結構：

```

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

```

2. 在受控的測試視窗中，使用 VM 重置來測試非計畫性容錯移轉：

使用虛擬機器 **Reset** (崩潰式測試)；普通的 **Stop** 操作可能不會觸發 FSFO。Tail /var/log/dgmgml-observer.log on oradg-obs 以監控故障移轉進度；完成後還原拓撲。

接下來呢？

此部署的 Oracle Data Guard Broker、Fast-Start Failover 和 Observer 組態現已就緒。

版權資訊

Copyright © 2026 NetApp, Inc. 版權所有。台灣印製。非經版權所有人事先書面同意，不得將本受版權保護文件的任何部分以任何形式或任何方法（圖形、電子或機械）重製，包括影印、錄影、錄音或儲存至電子檢索系統中。

由 NetApp 版權資料衍伸之軟體必須遵守下列授權和免責聲明：

此軟體以 NETAPP「原樣」提供，不含任何明示或暗示的擔保，包括但不限於有關適售性或特定目的適用性之擔保，特此聲明。於任何情況下，就任何已造成或基於任何理論上責任之直接性、間接性、附隨性、特殊性、懲罰性或衍生性損害（包括但不限於替代商品或服務之採購；使用、資料或利潤上的損失；或企業營運中斷），無論是在使用此軟體時以任何方式所產生的契約、嚴格責任或侵權行為（包括疏忽或其他）等方面，NetApp 概不負責，即使已被告知有前述損害存在之可能性亦然。

NetApp 保留隨時變更本文所述之任何產品的權利，恕不另行通知。NetApp 不承擔因使用本文所述之產品而產生的責任或義務，除非明確經過 NetApp 書面同意。使用或購買此產品並不會在依據任何專利權、商標權或任何其他 NetApp 智慧財產權的情況下轉讓授權。

本手冊所述之產品受到一項（含）以上的美國專利、國外專利或申請中專利所保障。

有限權利說明：政府機關的使用、複製或公開揭露須受 DFARS 252.227-7013（2014 年 2 月）和 FAR 52.227-19（2007 年 12 月）中的「技術資料權利 - 非商業項目」條款 (b)(3) 小段所述之限制。

此處所含屬於商業產品和 / 或商業服務（如 FAR 2.101 所定義）的資料均為 NetApp, Inc. 所有。根據本協議提供的所有 NetApp 技術資料和電腦軟體皆屬於商業性質，並且完全由私人出資開發。美國政府對於該資料具有非專屬、非轉讓、非轉授權、全球性、有限且不可撤銷的使用權限，僅限於美國政府為傳輸此資料所訂合約所允許之範圍，並基於履行該合約之目的方可使用。除非本文另有規定，否則未經 NetApp Inc. 事前書面許可，不得逕行使用、揭露、重製、修改、履行或展示該資料。美國政府授予國防部之許可權利，僅適用於 DFARS 條款 252.227-7015(b)（2014 年 2 月）所述權利。

商標資訊

NETAPP、NETAPP 標誌及 <http://www.netapp.com/TM> 所列之標章均為 NetApp, Inc. 的商標。文中所涉及的所有其他公司或產品名稱，均為其各自所有者的商標，不得侵犯。