



將大量 Oracle 資料庫遷移至異質環境 AWS 的

AWS 方案指引



AWS 方案指引: 將大量 Oracle 資料庫遷移至異質環境 AWS 的

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon 的商標和商業外觀不得用於任何非 Amazon 的產品或服務，也不能以任何可能造成客戶混淆、任何貶低或使 Amazon 名譽受損的方式使用 Amazon 的商標和商業外觀。所有其他非 Amazon 擁有的商標均為其各自擁有者的財產，這些擁有者可能附屬於 Amazon，或與 Amazon 有合作關係，亦或受到 Amazon 贊助。

Table of Contents

簡介	1
概觀	2
準備階段	3
向前滾動階段	3
傳輸階段	3
遷移階段	5
設定	5
準備階段	6
備份資料表空間	6
傳輸備份複本	7
還原備份複本	9
向前滾動階段	10
進行增量備份	10
傳輸備份	10
轉換並套用	11
傳輸階段	13
讓來源唯讀	13
最終增量備份	13
匯出中繼資料	14
傳輸檔案	14
匯入中繼資料	15
驗證階段	15
檢查資料表空間	15
進行讀取/寫入	16
結論	17
文件歷史紀錄	18
.....	xix

將大型 Oracle 資料庫遷移至跨平台環境 AWS 的

Incheol Roh , Amazon Web Services (AWS)

2021 年 3 月 ([文件歷史記錄](#))

將大於 100 TB 的 Oracle 資料庫從內部部署遷移至 Amazon Web Services (AWS) 雲端時，遷移停機時間是一個嚴重因素。特別是，如果系統是關鍵任務系統，例如全球企業資源規劃 (ERP) 或製造執行系統 (MES)，則您想要盡可能減少停機時間以實現業務連續性。

在具有不同終端格式的系統之間遷移 Oracle 資料庫的方法有很多，如[遷移 Oracle 資料庫的策略](#) [AWS](#)中所述。其中一種方法是 Oracle 跨平台可傳輸資料表空間 (XTTS)，您可以使用它將遷移停機時間從數天縮短為數小時。

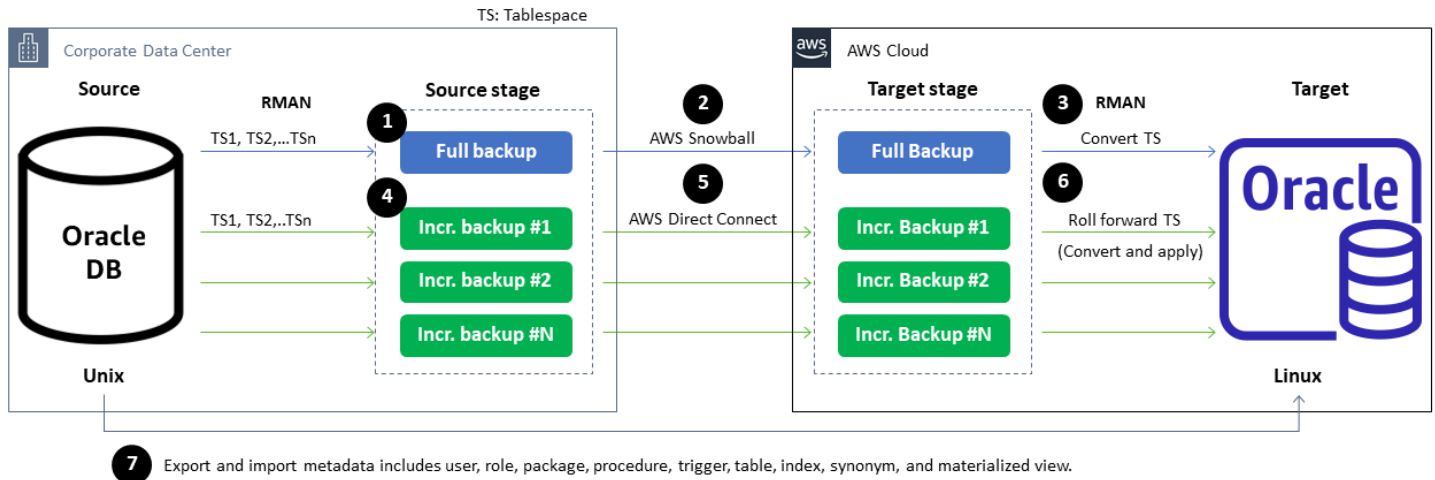
結合 Oracle XTTS 與 Oracle Recovery Manager (RMAN) 增量備份可大幅減少在執行不同終端格式的平台之間移動資料所需的停機時間。

本指南介紹如何使用 [AWS Snowball Edge](#)、[AWS Direct Connect](#)和 [Amazon FSx for Lustre](#) 搭配 Oracle XTTS 搭配 RMAN 增量備份。此方法的目標是將具有非常大型資料集的環境中的遷移停機時間降至最低。

概觀

這是將 Oracle 資料庫遷移至搭配 Snowball Edge 和 FSx for Lustre AWS 使用 Oracle XTTS Direct Connect和 RMAN 增量備份的概念程序。

下圖顯示 Oracle 資料庫跨不同終端格式的高階遷移步驟。



1. 完整備份所有資料表空間。
2. 使用 Snowball Edge 將備份從來源階段移至目標階段。
3. 將資料表空間轉換為目標資料庫。
4. 進行增量備份。
5. 使用 將增量備份從來源階段 Direct Connect 傳輸到目標階段。
6. 向前滾動增量備份，轉換它們並將其套用至目標資料庫。
7. 匯出和匯入要傳輸之所有資料表空間的中繼資料。

在切換之前，您可以執行下列動作，將停機時間降至最低：

- 匯出和匯入非區段型物件的中繼資料，包括 USER、PACKAGE_SPEC、PACKAGE_BODY、PROCEDURE和 FUNCTION
- 增加完整備份和增量備份的平行處理
- 轉換資料檔案
- 在遷移期間滾動備份

Oracle 文件使用跨平台增量備份 ([2471245.1](#)) 減少可傳輸資料表空間停機時間，說明如何搭配 RMAN 增量備份使用 Oracle XTTS。文件也包含需求和建議的詳細資訊。文件不會描述如何將 Oracle 資料庫從內部部署環境遷移至 Oracle on , AWS 或如何平行化每個遷移步驟以將停機時間降至最低。

本指南提供平行化階段的方法，以極大的資料大小將關鍵任務系統環境中的遷移停機時間降至最低。

在初始設定階段之後，搭配 RMAN 增量備份使用 Oracle XTTS 的高階步驟包括下列階段。

階段 1 – 準備階段

準備階段包含下列步驟：

1. 資料表空間的初始完整備份 (level=0) 會在來源資料庫上取得來源階段，也就是 NAS 儲存體。
2. 備份複本會使用 Snowball Edge 傳輸到目標階段，該階段是與 Amazon Simple Storage Service (Amazon S3) 整合的 FSx for Lustre。
3. 備份資料表空間會以小端數格式還原並轉換為目標資料庫。

此階段中的步驟只會在遷移期間執行一次。在此階段中，來源資料庫中可以完全存取要傳輸的資料。

階段 2 – 向前滾動階段

向前滾動階段包含下列步驟：

1. 增量備份會從來源資料庫取得至來源階段。
2. 增量備份複本會透過 傳輸到目標階段 Direct Connect。
3. 增量備份複本會以小端數格式轉換為目標資料庫。然後，複本會套用至初始目標資料庫，稱為向前滾動步驟。

您可以多次執行此階段。每個連續的增量備份都應該花費較少的時間，並將目的地資料檔案複本與來源資料庫一起更新。如同第 1 階段，在此階段中傳輸的來源資料可以完全存取。

階段 3 – 傳輸階段

第三個階段包含下列步驟：

1. 傳輸的資料表空間會變更為唯讀。

2. 最終增量備份是從來源資料庫取得。
3. 中繼資料會匯出。
4. 備份會傳輸並套用至目的地。
5. 匯入物件中繼資料。

此時，目的地資料庫的系統變更編號 (SCN) 與來源資料庫的系統變更編號 (SCN) 一致。

可傳輸資料表空間的中繼資料會從來源資料庫匯出，並匯入目的地資料庫。中繼資料包含使用者、角色、套件、程序、函數、資料表和索引的資訊。

最後，資料表空間會進行讀取/寫入，以便從應用程式完整存取目的地資料庫。

此階段後接驗證階段。

遷移階段

本指南涵蓋將 Oracle 單一執行個體和 Oracle RAC 遷移為來源和目標資料庫。如果來源和目標是 Oracle RAC，您可以為每個遷移步驟最大化平行處理。

階段 0 – 初始設定階段

1. 在目標 Amazon Elastic Compute Cloud (Amazon EC2) 執行個體上安裝 Oracle 資料庫 12.1.0.2 或更新版本。
2. 驗證目標資料庫。

您必須驗證目標資料庫執行個體是否具有與來源資料庫相同的時區和字元集。否則，此遷移方法將會失敗

若要檢查來源和目標中的時區版本是否相同，請執行下列命令。

```
SQL> select * from v$timezone_file;
FILENAME          VERSION      CON_ID
-----
timezlg14.dat      14           0
```

若要檢查來源和目標中的資料庫字元集和國家字元集是否相同，請執行下列命令。

```
SQL> select * from nls_database_parameters
      where parameter in ('NLS_CHARACTERSET', 'NLS_NCHAR_CHARACTERSET');
PARAMETER          VALUE
-----
NLS_NCHAR_CHARACTERSET      UTF8
NLS_CHARACTERSET            AL32UTF8
```

3. 設定 Oracle XTTS 公用程式。

在來源系統上，從 Oracle 文件 2471245.1 [下載並擷取指令碼檔案 rman_xttconvert_VER4.zip](#)。使用特定組態修改 xtt.properties 檔案中的參數。然後將 xtt.properties 和 xttdriver.pl 指令碼複製到目的地系統。

階段 1 – 準備階段（來源資料庫保持線上狀態）

在此階段中，將要傳輸之資料表空間的資料檔案備份在來源系統上。備份複本會傳輸到目的地系統，並透過執行 `xttdriver.pl` 指令碼來還原。

步驟 1：在來源系統上取得完整 (level=0) 資料表空間備份

若要縮短備份持續時間，請將 `xtt.properties` 和 `xttdriver.pl` 檔案複製到多個目錄。在每個目錄下的每個 `xtt.properties` 檔案中，在 `tablespaces` 參數中輸入資料表空間的名稱。然後，在每個目錄下，`xttdriver.pl` 使用 `--backup` 選項執行。除了在安裝目標資料庫期間建立的 `SYSTEM`、`UNDO`、`SYS_AUX` 和 `TEMP`，此命令會傳輸所有資料表空間。

例如，每個 `xtt01~xtt04` 目錄都有 `xtt.properties` 檔案中 `tablespaces=` 參數具有不同值的所有 `xtt` 指令碼 (`xtt.properties` 和 `xttdriver.pl`)。編輯每個 `xtt.properties` 檔案中的 `tablespaces` 參數時，請考慮分割資料表空間群組，讓每個資料表空間群組中的資料檔案總大小相似。

在本指南中，範例假設有四個資料表空間群組。如果來源資料庫是 Oracle 單一執行個體，您可以建立所有 `xtt01~04` 目錄。如果來源資料庫是 Oracle RAC，您可以在每個 Oracle RAC 伺服器中建立每個 `xtt` 目錄。

```
[oracle@erp expimp]$ ls
out xtt01 xtt02 xtt03 xtt04
[oracle@erp out]$ ls
out01 out02 out03 out04
```

下表顯示 `xtt.properties` 檔案中 `tablespaces=` 參數的範例。

xtt01	tablespaces=APPS_TS_TX_DATA
xtt02	tablespaces=APPS_TS_TX_IDX
xtt03	tablespaces=APPS_TS_SUMMARY
xtt04	tablespaces=APPS_CALCLIP, APPS_OMO, APPS_TS_ARCHIVE, APPS_TS_DISCO, APPS_TS_DISCO_OLAP , APPS_TS_INTERFACE, APPS_TS_M

```
EDIA, APPS_TS_NOLOGGING,
APPS_TS_QUEUES, APPS_TS_SEED...
```

若要防止在平行xttdriver.pl執行時刪除任何現有的資料檔案備份複本，請在 中註解下行xttdriver.pl。

```
if (@present)
{
    ## Try deleting any existing copied datafiles
    ##Comment out it for executing rman command in parallel by AWS
    ##      system("\rm -f $dfcopydir/*.tf");
    sleep 5;
}
```

使用 xttdriver.pl進行來源系統的 RMAN 備份。

如果來源系統是 Oracle RAC，則可以同時在每個 Oracle RAC 執行個體上執行。

xttdriver.pl 的輸出會儲存在 TMPDIR 環境變數中。使用 --backup選項執行每個xttdriver.pl命令，並使用 --debug選項開啟偵錯模式。

```
cd /u01/oracle/expimp/xtt<nn>
export TMPDIR=/u01/oracle/expimp/out/out<nn>
$ORACLE_HOME/perl/bin/perl xttdriver.pl --backup --debug 3
```

在此範例中，<nn> 代表資料表空間群組。如果有四個資料表空間群組，<nn> 會變成 01、03、02和 04。

步驟 2：將完整備份複本傳輸至目的地系統

使用下列兩個選項之一，將您的備份複本傳輸至目的地系統。

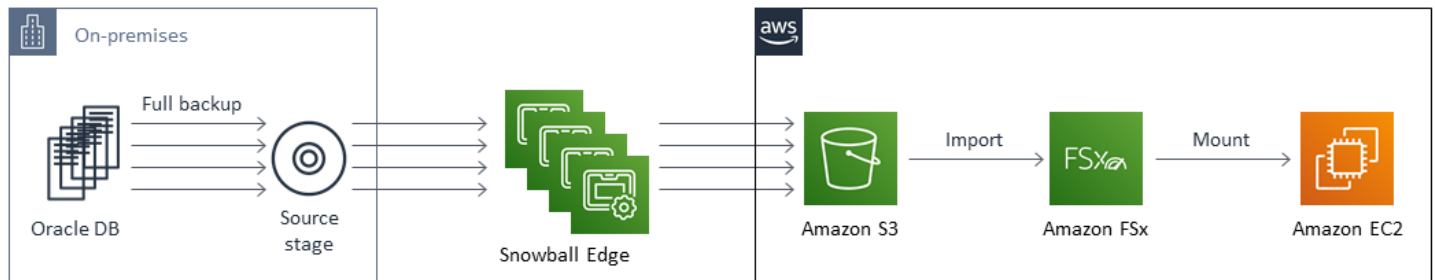
選項 1 – 使用 AWS Snowball Edge

路徑：來源資料庫 -> 來源階段 AWS Snowball Edge -> Amazon S3 -> FSx for Lustre

Note

AWS Snowball Edge 不再提供給新客戶。新客戶應探索[AWS DataSync](#)是否有線上傳輸、[AWS 資料傳輸終端機](#)是否有安全的實體傳輸，或 AWS Partner 解決方案。針對邊緣運算，探索 [AWS Outposts](#)。

下圖顯示使用 Snowball Edge 傳輸完整備份。



Snowball Edge 是堅固耐用的實體儲存和運算裝置，可用來將大規模資料移至其中 AWS。Snowball Edge 有助於克服大規模資料傳輸可能遇到的挑戰，包括長時間傳輸、缺乏可用頻寬和安全問題。

整個資料檔案備份副本由 Snowball Edge 傳輸到 Amazon S3。如果來源系統大小超過 100 TB，您必須使用多個 Snowball Edge 裝置來縮短傳輸持續時間。

Amazon FSx for Lustre 與 Amazon S3 深度整合。您可以在幾分鐘內建立連結至 S3 儲存貯體的 FSx for Lustre 檔案系統。連結至 Amazon S3 資料儲存庫時，FSx for Lustre 檔案系統會以透明的方式將 Amazon S3 物件呈現為檔案。在實際環境中，FSx for Lustre 的效能取決於其儲存容量、每個檔案的大小，以及檔案分發到檔案系統的程度。當 FSx for Lustre 用作目標階段儲存體時，您不需要將資料表空間備份複本從 Amazon S3 還原至 Amazon Elastic Block Store (Amazon EBS) 或 Amazon Elastic File System。

1. 將 S3 儲存貯體匯入 FSx for Lustre。

使用 Snowball Edge 將來源階段區域（例如 `src_backups`）傳輸並存放至 S3 儲存貯體（例如 `s3-src-backups`）。

建立 FSx for Lustre 檔案系統（例如 `dest_backups`）做為資料檔案備份複本的目的地階段區域。

在目的地系統 (Amazon EC2) 上安裝開放原始碼 Lustre 用戶端。如需詳細資訊，請參閱《Amazon FSx for Lustre 使用者指南》。

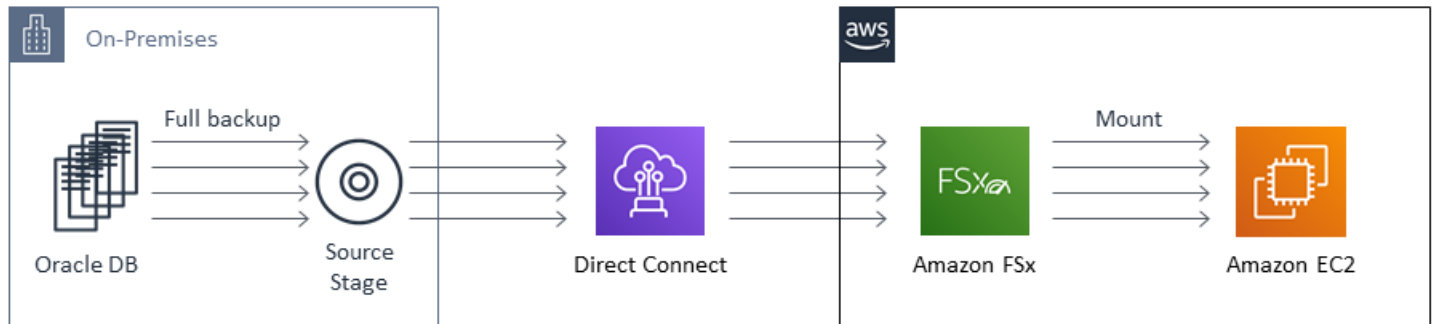
安裝 Lustre 用戶端後，將 FSx for Lustre 檔案系統掛載到目的地系統 (Amazon EC2)。

2. `res.txt` 在來源 `$TMPDIR` 上從 傳輸到目的地 `$TMPDIR` 上的。

選項 2 – 使用 AWS Direct Connect

路徑：來源資料庫 -> 來源階段 AWS Direct Connect -> FSx for Lustre

下圖顯示使用 傳輸完整備份 AWS Direct Connect。



Direct Connect 可讓您在資料中心、辦公室或主機代管環境與 之間建立私有網路連線 AWS。這些私有網路連線可降低網路成本、提高輸送量，並提供一致的體驗。

您可以將資料檔案備份副本直接從來源系統傳輸到掛載 Amazon FSx for Lustre 檔案系統的目的地系統 (Amazon EC2)。如果您可以容納的高頻寬，此選項會比 Snowball Edge 選項更快 Direct Connect。

在 Snowball Edge 或 傳輸資料檔案備份複本之後 Direct Connect，您可以在目標階段區域的 FSx for Lustre 檔案系統中看到它們。

步驟 3：還原完整備份複本並轉換資料檔案

還原完整備份複本，並將資料檔案轉換為目的地系統最終格式。若要減少還原和轉換持續時間，請針對每個資料表空間群組使用 `--restore` 選項執行 `xttdriver.pl` 命令。

```
cd /u01/oracle/expimp/xtt<nn>
export TMPDIR=/u01/oracle/expimp/out/out<nn>
$ORACLE_HOME/perl/bin/perl xttdriver.pl --restore --debug 3
```

步驟完成後，資料檔案會放置在目的地系統 `xtt.properties` 檔案中定義的 `dest_datafile_location` 中。

階段 2 – 向前滾動階段（來源資料庫保持線上狀態）

您可以視需要重複向前滾動階段多次，以將目的地資料檔案擷取到來源資料庫。

向前滾動階段包含下列步驟。

1. 從來源資料庫建立增量備份。
2. 將備份傳輸至目的地系統。
3. 將備份轉換為目的地系統最終格式。
4. 將備份套用至轉換後的目的地資料檔案副本，以將其向前滾動。

每個連續的增量備份都應該花費較少的時間，並將目的地資料檔案複本與來源資料庫一起更新。

步驟 1：平行進行增量備份

對在來源系統上平行傳輸的資料表空間群組進行增量備份。

此步驟會為 `xtt.properties` 檔案中列出的所有資料表空間= 建立增量備份。

如果您可以在來源系統上啟用 BLOCK CHANGE TRACKING 功能，則可以大幅縮短增量備份的時間。

下列命令會在完整備份後自動辨識下一個 SCN。

```
cd /u01/oracle/expimp/xtt<nn>
export TMPDIR=/u01/oracle/expimp/out/out<nn>
$ORACLE_HOME/perl/bin/perl xttdriver.pl --backup --debug 3
```

步驟 2：將增量備份和 `res.txt` 檔案傳輸至目的地系統

如果您有足夠的頻寬，您可以使用 縮短傳輸持續時間 Direct Connect。

在 `src_scratch_location` 和 之間傳輸增量備份 `dest_scratch_location`。在 `$TMPDIR` 來源系統和目的地系統 `$TMPDIR` 上的 之間傳輸 `res.txt` 檔案。

如果您進行多個增量備份，則必須在最後一個增量備份之後複製 `res.txt` 檔案，才能套用到目的地系統。

```
[source]$ scp $TMPDIR/res.txt oracle@[dest]:/u01/oracle/expimp/out/out<nn>
[source]$ scp <src_scratch_location>/* oracle@[dest]:<dest_scratch_location>
```

步驟 3：轉換並套用增量備份

將增量備份轉換為目的地系統最終格式，並將增量備份套用至目的地系統上的目的地資料檔案副本。

在本指南中，假設資料表空間群組是四個。使用每個資料表空間群組的 `--restore` 選項執行每個 `xttdriver.pl` 命令。

在此步驟中，Oracle XTTS 公用程式會重新啟動目標資料庫。若要平行執行命令，您必須在 Perl 指令碼中使用下列自訂 `xttdriver.pl`。

1. 註解以下幾行：

- 第 4867 行：`my $outputstart = `sqlplus -L -s \"/ as sysdba\"`
`@xttstartupnomount.sql`;`
- 第 4868 行：`checkError ("Error in executing xttstartupnomount.sql",
$outputstart);`
- 第 4992 行：`my $outputstart = `sqlplus -L -s \"/ as sysdba\"`
`@xttdbopen.sql`;\`;`

2. 讓目標資料庫處於 NOMOUNT 狀態。

```
sqlplus / as sysdba @xttstartupnomount.sql
```

3. 平行執行增量備份的向前滾動。

```
cd /u01/oracle/expimp/xtt<nn>  
export TMPDIR=/u01/oracle/expimp/out/out<nn>  
$ORACLE_HOME/perl/bin/perl xttdriver.pl --restore --debug 3
```

4. 滾動所有增量備份之後，將目標資料庫設定為 OPEN 狀態。

```
sqlplus / as sysdba @xttdbopen.sql
```

此時，您可以重複第 2 階段，直到來源和目的地資料庫上的 SCNs 夠接近為止。您必須決定要繼續進行階段 3 還是重複階段 2，取決於指定的目標停機時間。

若要減少第 3 階段（傳輸階段）期間的停機時間，您可以在將來源資料庫設定為之前，先匯出和匯入非區段型物件的中繼資料 FUNCTION，例如 PROCEDURE、USER PACKAGE 和 READ ONLY。

如果來源資料庫中的資料庫物件數量非常大（數百個），匯出和匯入中繼資料需要幾個小時。在此情況下，請考慮匯出中繼資料。

匯出非區段型物件會在來源系統上的讀取/寫入中執行。然後，您必須先將物件匯入目的地系統，來源系統才會設為 READ ONLY。此時，您必須保持資料庫來源物件，例如 PACKAGE、PROCEDURE 和 FUNCTION 不變。

這是傾印參數檔案的範例，用於匯出非區段型物件的中繼資料，包括 USER、PACKAGE_SPEC、PROCEDURE、PACKAGE_BODY 和 FUNCTION。

```
directory=dmpdir
dumpfile=xttmsc%U.dmp
full=y
filesize=1048576000
logfile=expmsc.log
metrics=y
exclude=TABLE,INDEX,CONSTRAINT,COMMENT,
MATERIALIZED_VIEW,MATERIALIZED_VIEW_LOG,SCHEMA_CALLOUT
```

使用下列命令匯出中繼資料。

```
SQL> create directory dmpdir as <location>;
expdp system/<system password> parfile=<parameter file>
```

這是用於匯入非區段物件中繼資料的傾印參數檔案範例。

```
directory=dmpdir
dumpfile=xttmsc%U.dmp
full=y
logfile=impmsc1.log
EXCLUDE=TABLESPACE, PROCOBJ, RLS_CONTEXT, RLS_GROUP, RLS_POLICY, TABLESPACE_QUOTA
metrics=y
remap_tablespace=
APPS_TS_ARCHIVE:SYSTEM,
APPS_TS_INTERFACE:SYSTEM,
APPS_TS_SEED:SYSTEM,
APPS_TS_SUMMARY:SYSTEM,
... .
```

目前，除了目的地系統TEMP上的 SYSTEM、UNDO、SYSAUX和 之外，沒有資料表空間。因此，您必須重新映射要匯入資料表SYSTEM空間的所有物件。

使用下列命令來匯入中繼資料。

```
SQL> create directory dmpdir as <location>;
impdp system/<system password> parfile=<parameter file>
```

現在，您可以在目的地資料庫上看到建立的物件。

```
SQL> select object_type, count(*) from dba_objects group by object_type order by
count(*) desc;
OBJECT_TYPE                                COUNT(*)
-----
SYNONYM                                    89327
PACKAGE                                    55670
PACKAGE BODY                              54447
VIEW                                       41378
JAVA CLASS                                31978
SEQUENCE                                  12766
....
```

除了 SYSTEM、SYSAUX、UNDO 和之外，沒有資料表空間 TEMP。USER 物件會以 USERS 做為預設資料表空間建立。在第 3 階段期間，將建立可傳輸的資料表空間，然後使用原始預設資料表空間來變更 USER 物件。

階段 3 – 傳輸階段（來源資料庫為唯讀）

在此階段，來源系統變成唯讀。套用最終增量備份，使目的地系統上的資料檔案與來源系統一致。然後，從來源系統匯出物件中繼資料，並將其匯入目的地系統。

步驟 1：讓來源資料庫中的資料表空間唯讀

做為 SYSDBA，將所有資料表空間傳輸到 READ ONLY 來源系統。

若要減少停機時間，您可以同時執行下列兩個步驟。

步驟 2：建立最終增量備份

在來源系統上，建立要傳輸之資料表空間的最終增量備份。

```
cd /u01/oracle/expimp/xtt<nn>
export TMPDIR=/u01/oracle/expimp/out/out<nn>
$ORACLE_HOME/perl/bin/perl xttdriver.pl --backup --debug 3
```

此步驟會傳回錯誤，例如「ORA-20001：TABLESPACE(S) IS READONLY」；預期會發生錯誤，您可以放心地予以忽略。

步驟 3：匯出中繼資料

從來源資料庫匯出可傳輸資料表空間的中繼資料。

這是匯出可傳輸資料表空間中繼資料的參數檔案範例。

```
directory=dmpdir
metrics=y
dumpfile=xttsmeta%U.dmp
filesize=1048576000
logfile=expxtts.log
transport_tablespaces=
APPS_TS_ARCHIVE,
APPS_TS_INTERFACE,
APPS_TS_MEDIA,
APPS_TS_NOLOGGING,
...
exclude=table_statistics,index_statistics
```

此外，如果來源系統有許多資料表和索引，您可以透過排除其統計資料來節省匯出期間的時間。匯入可傳輸資料表空間之後，請將統計資料匯入目的地系統。

執行之前expdp，請先建立資料庫目錄，其中傾印檔案會存放在來源系統上。

```
SQL> create directory dmpdir as <location>;
expdp system/<system password> parfile=<parameter file>
```

下列兩個步驟是具有 RMAN 增量備份的跨平台可傳輸資料表空間的最後步驟。這些步驟必須依序執行。

步驟 4：傳輸檔案並套用最終增量備份

傳輸最終增量備份，並將傾印檔案匯出至目的地系統、轉換和套用最終增量備份。

使用 將最終增量備份複本和 res.txt 檔案 Direct Connect 傳輸至目的地。您可以使用 VPN 連線，但如果足夠的頻寬，使用 Direct Connect 會大幅減少停機時間。

若要還原最終增量備份，請在目的地系統上，使用 --restore 選項為每個資料表空間群組執行下列命令。

```
cd /u01/oracle/expimp/xtt<nn>
export TMPDIR=/u01/oracle/expimp/out/out<nn>
$ORACLE_HOME/perl/bin/perl xttdriver.pl --restore --debug 3
```

步驟 5：匯入物件中繼資料

使用 Oracle Data Pump 將物件中繼資料匯入目的地系統。執行下列命令，以取得目的地系統上 `transport_datafiles=` 參數的資料檔案清單。

```
cd /u01/oracle/expimp/xtt<nn>
export TMPDIR=/u01/oracle/expimp/out/out<nn>
$ORACLE_HOME/perl/bin/perl xttdriver.pl -e
```

每當您執行先前的命令時，就會取得具有 `transport_datafiles=` 參數的 `xttplugin.txt` 檔案。從所有 `xttplugin.txt` 檔案 `transport_datafiles=` 合併一行，並將資料檔案清單放入匯入中繼資料的參數檔案 `transport_datafiles` 引數中。

下列程式碼片段顯示用於在目的地系統上匯入可傳輸資料表空間的參數檔案。

```
directory=dmpdir
metrics=y
dumpfile=xttsmeta%U.dmp
logfile=impxtts.log
exclude=TYPE
transport_datafiles= '+EBSDATA/APPS_TS_TX_DATA_2.dbf', '+EBSDATA/
APPS_TS_TX_DATA_11.dbf', '+EBSDATA/APPS_TS_TX_DATA_22.dbf', '+EBSDATA/
APPS_TS_TX_DATA_183.dbf', '+EBSDATA/APPS_TS_TX_DATA_204.dbf', '+EBSDATA/
APPS_TS_TX_DATA_219.dbf', '+EBSDATA/APPS_TS_TX_DATA_227.dbf'....
```

在執行之前 `impdp`，請建立指向匯出傾印檔案位置的資料庫目錄。

```
SQL> create directory dmpdir as <location>;
impdp system/<system password> parfile=<parameter file>
```

階段 4 – 驗證傳輸的資料

步驟 1：檢查資料表空間是否損毀

在此步驟中，傳輸的資料表空間只會在目的地資料庫中讀取。您可以執行 `RMAN` 命令來驗證資料表空間是否有效，如下所示。

```
sqlplus / as sysdba;
set feedback off
set pagesize 0
set heading off
spool tbs_val.sql;
select 'validate tablespace '||tablespace_name||' check
logical;' from dba_tablespaces where tablespace_name not in
('SYSTEM','SYSAUX','TEMP','USERS','UNDOTBS1','UNDOTBS2','UNDOTBS3','UNDOTBS3','UNDOTBS4');
spool off;
exit;

--Remove the first and last line in the spool file, tbs_val.sql
rman target /
RMAN> @tbs_val.sql
```

此外，您可以檢查物件的計數，例如資料表、索引、同義詞、檢視和套件物件。

步驟 2. 將目的地資料庫中的所有傳輸資料表空間設為讀取/寫入

```
sqlplus / as sysdba;
set feedback off
set pagesize 0
set heading off
spool tbs_rw.sql
select 'alter tablespace '||tablespace_name||' read write;' from dba_tablespaces where
status='READ ONLY';
spool off;
exit;
--Remove the first and last line in the spool file, tbs_rw.sql
sqlplus / as sysdba;
SQL> @tbs_rw.sql
```

結論

在本指南中，您了解如何透過使用 Oracle 跨平台可傳輸資料表空間和 RMAN 增量備份 AWS Snowball Edge，以及 Amazon FSx for Lustre AWS Direct Connect，將停機時間降至最低。

將 Oracle 資料庫從現場部署遷移到 時 AWS，請考慮下列變數：

- 所需的網路頻寬 Direct Connect
- FSx for Lustre 的容量
- Oracle 資料庫的中繼資料大小
- 增量備份大小

使用本指南中建議的 方法，從在 Unix 系統上執行的 100 TB Oracle 資料庫遷移到 時 AWS，您可以將停機時間縮短到大約 10 小時。

文件歷史記錄

下表說明本指南的重大變更。如果您想收到有關未來更新的通知，可以訂閱 [RSS 摘要](#)。

變更	描述	日期
二	變更簡介和概觀中的異質提及。	2021 年 7 月 14 日
二	初次出版	2021 年 3 月 2 日

本文為英文版的機器翻譯版本，如內容有任何歧義或不一致之處，概以英文版為準。