

リージョン間バックアップを使用した ディザスタ・リカバリ

Oracle Public CloudでのObject Storageリージョン間レプリケーションを使用した、
低コストのディザスタ・リカバリ・ソリューション

2022年10月6日

Copyright © 2022, Oracle and/or its affiliates

公開

目次	
目的	3
概要	3
セキュリティ要件	4
ディザスタ・リカバリの概要	5
リージョン間レプリケーションのための初期構成	5
リージョン間レプリケーションの監視	5
プライマリ・リージョンの障害後のディザスタ・リカバリ手順	6
手順1：Cloudコンソールを使用した新規データベース（ターゲット）の作成	6
手順2：クラウド・バックアップ・モジュールのインストール	6
手順3：リストアのためのターゲット・データベースの準備	7
手順4：TDEウォレットとOracle Net構成のリストア	8
手順5：tnsnames.oraとsqlnet.oraの検証	9
手順6：ターゲット・データベースのリストアおよびリカバリ	9
手順7：データベース・リカバリ後の手順	14
結論	16
付録A：タイムスタンプを使用したポイント・イン・タイム・リカバリ	16
付録B：完全なレベル0	18

目的

本書では、Object Storageサービスのリージョン間レプリケーション機能を使用して、低コストのディザスタ・リカバリを構成するためのガイドラインとベスト・プラクティスを提供します。リモート・リージョンでのデータベースのディザスタ・リカバリを対象とした詳しい手順も含まれています。本書は、Oracle Public Cloud上にあり、Oracle Exadata Database Service on Dedicated Infrastructure（ExaDB-D）を使用するすべてのOracleデータベースに当てはまります。

概要

Oracle Object Storageサービスのリージョン間レプリケーションが提供する実装しやすい低コストのレプリケーション機能では、ローカル・リージョンにあるExadata Database Serviceのデータベース・バックアップを自動的にレプリケートできます。ローカル・リージョンで障害が発生した場合、レプリケートされたローカル・バックアップ・コピーを使用して、リモート・リージョンにあるデータベースを新規/既存のExadata Database Service（ExaDB-D）にリストアでき、データ損失やリカバリ・ポイント目標（RPO）を最小限に抑えられます。データベースとアーカイブ・バックアップがローカル・リージョンのObject Storageサービス（OSS）パケットに書き込まれると、ほぼ直後にこれらがリージョン間のOSSパケットにレプリケートされます。MAAテストで確認された遅延は5分未満でしたが、この遅延は、待機時間および帯域幅によってリージョンごとに異なる可能性があります。したがって、全体としてのRPOは基本的に、直前に成功したアーカイブ・ログ・バックアップ（デフォルトでは30分ごとに設定）よりも数分長くなります。

ExaDB-Dは、顧客が定義したObject Storageサービス（OSS）パケットを使用したバックアップをサポートしています。ExaDB-Dデータベース向けに顧客が作成および管理するOSSパケットを使用できるのは、ユーザー構成のバックアップのみです。ユーザー構成バックアップの設定と管理には、dbaascliユーティリティを利用します。Cloudコンソール、SDK、またはREST APIエンドポイントを使用して、ユーザー構成バックアップを有効化、設定、表示することはできません。ユーザー構成バックアップはコンソールからは独立してセットアップされますが、dbaascliユーティリティの使用は完全にサポートされています。ユーザー構成バックアップは、MAAのバックアップおよびリカバリのベスト・プラクティスに従いますが、これらはすべてdbaascliユーティリティ内に組み込まれています。

ソース・データベースのバックアップ保存期間は、リモート・バケットで自動的に維持されます。増分バックアップ・ピースがOracle Recovery Managerによって期限切れと判断されて削除されると、OSSは、これに対応するピースをリモート・リージョンで自動的に削除します。リモート・リージョンに必要なストレージ量には、ソース・データベースのバックアップによって消費されるストレージ量が反映されます。

リカバリ時間目標（RTO）は、障害発生時にリモート・リージョンで必要になる手順の影響を受けます。ソースと同じバージョンに維持され、パッチが適用されている既存のExaDB-Dを利用すると、RTOを短縮できます。本書内に記載されたディザスタ・リカバリ手順は、Recovery Managerのリストアおよびリカバリ・コマンドを利用しており、すべてのExaDB-Dノード間でパラレル化されています。クラスタ間でチャンネルをパラレル化することで、ディザスタ・リカバリ計画のリストアおよびリカバリ・フェーズの完了に必要な時間を短縮できるため、RTOも短縮できます。チャンネルのパフォーマンスおよびスループットに関する推奨事項については、次のMAAドキュメントを参照してください。[Object Storageを使用したExaCSのデータベース・バックアップおよびリストアのパフォーマンス測定](#)

OSSでは、コンソール、CLI、SDK、REST APIからいくつかのレプリケーション指標にアクセスできます。ただし、これらの指標では、オブジェクト数と同期時間のリアルタイム推定は提供されません。RPOを算出する場合、最後にレプリケートされた同期時間がもっとも役立ちます。**レプリケートされた同期時刻より前に正しく完了したバックアップは、すべて確実にリモート・リージョンにレプリケートされます。**詳細については、本書のレプリケーションの監視に関するセクションを参照してください。

使用しているデータベースのRTOおよびRPO要件が満たされる場合、この技術概要で説明する、OSSリージョン間レプリケーションを使用したディザスタ・リカバリ・ソリューションは妥当な選択肢になります。RPOが10秒未満でRTOが数秒から数分である場合は、Cloud Data Guardの実装を検討してください。[Exadata Cloud Database SystemsでのOracle Maximum Availability Architecture](#)も併せて参照してください。

セキュリティ要件

Oracle Cloud Infrastructure (OCI) で作成されるデータベースは、透過的データ暗号化 (TDE) が自動的に有効化されるため、暗号化された状態でデータが保存されます。dbaascliを使用して生成されるバックアップは、Recovery Managerの暗号化/圧縮を使用して、OSSへの転送前にTDEデータファイルを復号化、圧縮、再暗号化します。ExaDB-Dでは暗号化キーの保管に関して、Oracle管理キーと顧客管理キーの両方がサポートされます。ただし、本書では、Oracle管理キー (ファイルベースのOracleウォレット) を使用した検証のみを実施しています。dbaascliユーティリティは、毎日、増分レベル0/1のバックアップの一部として**ewallet.p12**を自動的にバックアップします。**cwallet.sso**はウォレット・バックアップ・ワークフローに含まれていない点に注意してください。

ディザスタ・リカバリの概要

この技術概要は、次の4つの部分に分かれています。

1. リージョン間のOSSレプリケーションのためのOracleデータベース初期構成
2. リージョン間のOSSレプリケーションと潜在的なデータ損失の監視
3. プライマリ・リージョン損失による障害シナリオでのデータベースのリストアとリカバリ
4. 新しくリストアされたデータベースのバックアップの構成

リージョン間レプリケーションのための初期構成

このセクションでは、ユーザー構成バックアップおよびOSSレプリケーション用にデータベースを構成するのに必要な手順を説明します。

1. バックアップを構成する前に、OSSバケットのレプリケーションを有効化します。
バックアップ構成後にレプリケーションを有効化した場合は、新規でレベル0のバックアップを取得する必要があります。これは、**レプリケーションを有効化しても既存バックアップはレプリケートされず、新規バックアップのみがレプリケートされる**からです。詳しくは、[付録B](#)を参照してください。
2. OSSバケットとレプリケーションをセットアップします。[OSSバケットの作成とレプリケーションの有効化](#)について、詳しくはOSSドキュメントを参照してください。
3. バックアップ用にデータベースを構成します。ExaDB-Dドキュメントの[ユーザー構成バックアップ](#)のセクションに記載された手順に従います。
4. ディザスタ・リカバリに必要な情報を収集します。
上記の構成を実行しながら、下の表に示した情報を収集します。この情報は後続のセクションで使用し、必要に応じて更新します。この情報を適切に更新し、使用できるようにしておくことで、障害発生時のRTOを短縮できます。
注：ドキュメントによっては、OSSバケットを示す用語としてコンテナが使用されています。

表1：

OSSデータ	ローカル・リージョン	リモート・リージョン
REGION_NAME	<LOCAL_REGION_NAME>	<REPL_REGION_NAME>
COMPARTMENT	<LOCAL_REGION_COMPARTMENT>	<REPL_REGION_COMPARTMENT>
SWIFT_URL	<LOCAL_REGION_SWIFT_URL>	<REPL_REGION_SWIFT_URL>
OPC_CONTAINER	<LOCAL_REGION_BUCKET>	<REPL_REGION_BUCKET>
OSS_USER	<LOCAL_REGION_OSS_USERNAME>	<REPL_REGION_OSS_USERNAME>
OSS_PASSWORD	<LOCAL_REGION_OSS_PASSWORD>	<REPL_REGION_OSS_PASSWORD>

ソース・データベースから情報を収集します。

データベース名	<DB_NAME>
データベースの一意的名前	<DB_UNIQUE_NAME>
ノード1のホスト名	<SOURCE_HOSTNAME_NODE1>
データベースID	<DBID>
自動ログイン・ウォレットのパスワード	<EWALLET_PASSWORD>

リージョン間レプリケーションの監視

レプリケーションのステータスと進捗は、ローカル・リージョンのコンソールか、[OSSレプリケーションのドキュメント](#)に記載されたその他のサポート対象アクセス方法を使用して監視できます。

コンソールでは、ソース・バケットのReplication Policyセクションに以下の情報が提供されます。

- **前回のレプリケーション**

これは、ソース・バケット内のすべてのオブジェクトが宛先に正しくレプリケートされたことが判明している最後の時刻です。前回のレプリケーションより前にソース・バケットに挿入されたバックアップは、すべて正しくレプリケートされたことがわかっています。注：この値は5～6分ごとに更新されます。

- **レプリケーション・ステータス**

ポリシーのレプリケーション・ステータスです。正しいステータスはActiveです。それ以外の場合は、コンソールで詳細を確認してください。

プライマリ・リージョンの障害後のディザスタ・リカバリ手順

このセクションでのソース・データベースとは、上の初期構成で設定したデータベースです。ターゲットは、レプリケートされたソース・データベースのバックアップ上にある新規データベースです。

手順1：Cloudコンソールを使用した新規データベース（ターゲット）の作成

- ソース・データベースと同じ<DB_NAME>および<DB_UNIQUE_NAME>を使用して、データベースを作成します。
- データベースのPDB値にはPDBTEMPを使用します。このPDBは、リストア後タスクの一部として削除されます。
- ソース・データベースで使用されているのと同じデータベース・バージョンを選択します。Oracleホーム・ディレクトリの値は異なっても構いません。
- TDEはOracle管理である必要があります。本書に示す構成例は、顧客管理キーをサポートしていません。
- ここでデータベースのバックアップを構成しないでください。バックアップの有効化は、データベース・リカバリ後のセクションの[手順h](#)に記載されています。
- ソース・データベースと同じ管理者パスワードを使用します。
これは必須ではありません。複製後タスクの手順で、クローニング後のパスワード変更について説明しています。
- ソース・データベースに含まれるPDBの数があっても、作成するPDBは、PDBTEMPという名前の1つだけです。
リカバリの完了後に、コンソールですべてのPDBが自動的に検出されます。
- データベースを作成する手順については、[コンソールを使用したデータベースの作成](#)を参照してください。

手順2：クラウド・バックアップ・モジュールのインストール

opc_installerのダウンロードとインストール

<https://www.oracle.com/database/technologies/oracle-cloud-backup-downloads.html>からopc_installer.zipをダウンロードし、ノード1の一時的な場所にコピーします。

oracleユーザーとしてディレクトリを作成し、opc_installer.zipをコピーして解凍します。

```
$ mkdir /home/oracle/tmp<DB_NAME>/down_opc_installer
$ cd /home/oracle/tmp<DB_NAME>/down_opc_installer
$ unzip opc_installer.zip
```

opc_install.jarの実行

-libDir、-configfile、-walletDirオプションについては、次の例に示すディレクトリ構造を使用します。これらのオプションはOracle Automatic Storage Management Cluster File System（Oracle ACFS）を利用するので、すべてのノードのファイルにアクセスできます。これらのファイルをOracle ACFSに配置しない場合、クラスタ内のその他のノードがリストア中に失敗します。

```
$ cd /home/oracle/tmp<DB_NAME>/down_opc_installer/opc_installer/opc_installer
$ java -jar opc_install.jar -host <REPL_REGION_SWIFT_URL> -opcid '<REPL_REGION_OSS_USERNAME>' -container '<REPL_REGION_BUCKET>' -opcPass '<REPL_REGION_OSS_PASSWORD>' -libDir /var/opt/oracle/dbaas_acfs/<DB_NAME>/opc -configfile /var/opt/oracle/dbaas_acfs/<DB_NAME>/opc/opc<DB_NAME>.ora -walletDir /acfs01/dbaas_acfs/<DB_NAME>/opc/opc_wallet
```

手順3：リストアのためのターゲット・データベースの準備

リストア準備の手順は以下のとおりです。

上の手順1で作成したデータベースについて、以下のinit.oraパラメータをメモしておきます。これらの値は、後続の手順6.bで、リストアされたinit.oraパラメータを更新するために使用します。

remote_listener	show parameter remote_listener
db_domain	show parameters db_domain
cluster_interconnects 注：cluster_interconnectsの値はインスタンスごとに異なります	select name, inst_id, value from gv\$parameter where name='cluster_interconnects';

手順1で作成したクラウド・データベースのファイルを削除します。oracleユーザーとして、次のSQLスクリプトbuild_delete_ASM_files.sqlを作成します。

```
$ cat /home/oracle/tmp<DB_NAME>/build_delete_ASM_files.sql
set head off
set echo off
set term off
set feed off
set linesize 999
set pagesize 0
set trimspool on
column value for a120
column name for a100
column HANDLE for a100
spool /tmp/delete_ASM_files.sh
select 'asmcmd rm '||name from v$datafile union all select 'asmcmd rm '||name from v$tempfile union all
select 'asmcmd rm '||member from v$logfile union all select 'asmcmd rm '||name from v$archived_log where
name is not null union all select 'asmcmd rm '||name from v$controlfile union all select 'asmcmd rm
'||trim(UPPER(value),'TO ') from V$RMAN_CONFIGURATION where NAME='SNAPSHOT CONTROLFILE NAME' union all
select 'asmcmd rm '||handle from v$backup_piece where DEVICE_TYPE='DISK' and handle is not null union all
select 'asmcmd rm '||NAME FROM V$FLASHBACK_DATABASE_LOGFILE union all select 'asmcmd rm '||value from
v$parameter where NAME='spfile';
select 'asmcmd ls -lt '|| value || '/' ||(select DB_UNIQUE_NAME from v$database) from v$parameter where
name='db_create_file_dest';
select 'asmcmd ls -lt '|| value || '/' ||(select DB_UNIQUE_NAME from v$database) from v$parameter where
name='db_recovery_file_dest';
spool off;
exit
```

oracleユーザーとして、SQL*Plusでbuild_delete_ASM_files.sqlスクリプトを実行します。
SQLによってシェル・スクリプト/tmp/delete_ASM_files.shが生成されます。

```
$ sqlplus / as sysdba
SQL> @ build_delete_ASM_files.sql
```

ファイル/tmp/delete_ASM_files.shの権限を変更します。

```
# chmod 774 /tmp/delete_ASM_files.sh
```

abortオプションを使用してデータベースを停止します。

```
$ srvctl stop database -d <DB_UNIQUE_NAME> -o abort
```

スクリプト/tmp/delete_ASM_files.shを実行して、データファイル、REDOログ、アーカイブ・ログ、制御ファイルを削除します。
gridユーザーとして、スクリプト/tmp/delete_ASM_files.shを実行します。

```
$(grid) /tmp/delete_ASM_files.sh > /tmp/delete_ASM_files.log
```

スクリプト実行の出力結果を画面またはログ・ファイルで確認します。

手順4：TDEウォレットとOracle Net構成のリストア

TDEウォレットをリストアする手順は以下のとおりです。

- ソース・データベース構成ファイルの最新バックアップを見つけてダウンロードする：
ohcfgfiles_<YYYYMMDD_HHMMI>.tar.gz
- ターゲットの現在のTDEウォレットを、ソース・データベースのTDEウォレットで置換する
- cwallet.ssoを作成する

バケット内で、ファイルohcfgfiles_<YYYYMMDD_HHMMI>.tar.gzの最新バージョンを見つけます。このファイルには、ソース・データベースのewallet.p12、tnsname.ora、sqlnet.oraが含まれています。これらのファイルは、ソース・データベースの日次増分バックアップの一部としてバックアップされています。最新の日時を持つファイルを特定します。

レプリケートされたOSSバケットからファイルを見つけてダウンロードする方法は、3つあります。以下のいずれかの方法を使用します。

ステージング・ディレクトリtmpを作成し、そこにcdを実行します。

```
$ mkdir /home/oracle/tmp<DB_NAME>/OHCFCFILES
$ cd /home/oracle/tmp<DB_NAME>/OHCFCFILES
```

方法1：CPの使用

コンソールから「Object Storage & Archive Storage」→「Bucket Details」に移動し、レプリケートされたバケットを表示します。

セクション<SOURCE_HOSTNAME_NODE1>-<DBID>-dbaastools-regularで、ohcfgfiles_<YYYYMMDD_HHMMI>.tar.gzという名前を持つ最新のオブジェクトを特定します。

ケバブ・メニュー（縦に並んだ3つのドット）からダウンロード機能を使用します。ケバブ・メニューはファイルの右側にあります。

方法2：oci-cliの使用

os object listコマンドを使用します。以下のコマンド・オプションを使用します。

bucket-name	<REPL_REGION_BUCKET>
prefix	<SOURCE_HOSTNAME_NODE1>-<DBID>-dbaastools-regular/ohcfgfiles
output	Table


```
List object using oci-cli
oci os object list --bucket-name <REPL_REGION_BUCKET> --prefix
<SOURCE_HOSTNAME_NODE1>-<DBID>-dbaastools-regular/ohcfgfiles --output table
```

上のコマンドで生成されたリストから、最新の<YYYYMMDD_HHMI>を持つファイルを特定し、ダウンロードします。

Download the specific file:

```
oci os object get --bucket-name <REPL_REGION_BUCKET> --file ohcfgfiles_<YYYYMMDD_HHMI>.tar.gz --name
<SOURCE_HOSTNAME_NODE1>-<DBID>-dbaastools-regular/ohcfgfiles_<YYYYMMDD_HHMI>.tar.gz"
```

方法3 : curlの使用

Using curl:

```
curl -u '<REPL_REGION_OSS_USERNAME>:<REPL_REGION_OSS_PASSWORD>'
<REPL_REGION_SWIFT_URL>/<REPL_REGION_BUCKET>?prefix=<SOURCE_HOSTNAME_NODE1> | python -m json.tool | grep -i
ohc
```

上のコマンドで生成されたリストから、最新の<YYYYMMDD_HHMI>を持つファイルを特定し、ダウンロードします。

Download the specific file:Use the "name:" value listed in output above for <NAME>

```
curl -u '<REPL_REGION_OSS_USERNAME>:<REPL_REGION_OSS_PASSWORD>' -X GET
<REPL_REGION_SWIFT_URL>/<REPL_REGION_BUCKET>/<NAME> -o ohcfgfiles_<YYYYMMDD_HHMI>.tar.gz
```

ダウンロードしたファイルtar.gzを解凍します。

oracleユーザーとして、ファイルohcfgfiles_<YYYYMMDD_HHMI>.tar.gzを解凍したのからファイルewallet.p12をコピーします。

Depending on the version:The wallet_root/tde directory is tde_wallet

```
$ cd /home/oracle/tmp<DB_NAME>/OHCFCFILES
$ tar -xvf ohcfgfiles_<YYYYMMDD_HHMI>.tar.gz
$ rm -rf /var/opt/oracle/dbaas_acfs/<DB_NAME>/wallet_root/tde/*
$ cp /home/oracle/tmp<DB_NAME>/OHCFCFILES/var/opt/oracle/dbaas_acfs/<DB_NAME>/wallet_root/tde/ewallet.p12
/var/opt/oracle/dbaas_acfs/<DB_NAME>/wallet_root/tde/
```

自動ログイン・ファイルcwallet.ssoを作成します。

oracleユーザーとして自動ログイン・パスワードを指定する必要があります。

Create cwallet

```
$ orapki wallet create -wallet /var/opt/oracle/dbaas_acfs/<DB_NAME>/wallet_root/tde -pwd
"<EWALLET_PASSWORD>" -auto_login
```

手順5 : tnsnames.oraとsqlnet.oraの検証

ohcfgfiles_<YYYYMMDD_HHMI>.tar.gzには、ソース・データベースのtnsnames.oraファイルおよびsqlnet.oraファイルも含まれています。これらのファイルはサービスtnsnamesのエンドポイントを作成するのに役立ちます。

ファイルの場所は次のとおりです。

```
/home/oracle/tmp<DB_NAME>/OHCFCFILES/<SOURCE_ORACLE_HOME>/network/admin/<DB_NAME>
```

ソースのtnsnames.oraファイルおよびsqlnet.oraファイルを、ターゲット・データベースの既存ファイルと比較します。

必要な場合は、既存ファイルを変更するか、情報を追加できます。たとえば、アプリケーションを別のデータベースに接続するためのカスタム・サービスを追加できます。

手順6 : ターゲット・データベースのリストアおよびリカバリ

ソース・バックアップからターゲット・データベースをリストアしてリカバリするには、以下の手順を実行します。

- 自動バックアップからspfileをpfileとしてリストアする (Recovery Manager)
- pfileを使用してデータベースをシャットダウンし、ノーマウントで起動する (SQL*Plus)

- c) 自動バックアップから制御ファイルをリストアする (Recovery Manager)
- d) pfile内のデータベース・パラメータ値 (control_files、remote_listenerなど) を更新する (SQL*Plus)
- e) pfileからspfileを作成し、データベース・クラスターリソースを更新する (SQL*Plus)
- f) バックアップ・ピースをカタログに追加する
- g) リストアを検証する (Recovery Manager)
- h) データベースを最新にリストア/リカバリする
- i) resetlogsオプションを使用してデータベースをオープンする (SQL*Plus)

a) 自動バックアップからspfileをpfileにリストアします。

```
$ rman target /
RMAN> startup force nomount;
RMAN> set dbid=<DBID>; RMAN>
run
{
allocate channel ch1 device type 'SBT_TAPE' parms
'SBT_LIBRARY=/var/opt/oracle/dbaas_acfs/<DB_NAME>/opc/libopc.so,
ENV=(OPC_PFILE=/var/opt/oracle/dbaas_acfs/<DB_NAME>/opc/opc<DB_NAME>.ora)'; RESTORE SPFILE TO
PFILE '<ORACLE_HOME>/dbs/init<DB_NAME>.ora' FROM AUTOBACKUP;
}
```

b) データベースをシャットダウンし、pfileのデータベース・パラメータを更新し、ノーマウントで起動します。

データベースのシャットダウン

```
$ sqlplus / as sysdba
SQL> shutdown;
```

pfileを更新します。以下のpfileパラメータを更新する必要があります。手順3でメモした値を使用します。

db_domain	現在のクライアントのサブネットに変更
remote_listener	現在のSCANホスト名に変更
cluster_interconnects	注：値はインスタンスごとに異なります

<ORACLE_HOME>/dbs/init<DB_NAME>.oraにリストアされたpfileを編集し、上記のパラメータを更新します。

pfileを使用してノーマウントで起動します。

```
$ sqlplus / as sysdba
SQL> startup nomount pfile='<ORACLE_HOME>/dbs/init<DB_NAME>.ora';
```

c) 自動バックアップから制御ファイルをリストアします。

制御ファイルは、pfileに指定されたASMディスク・グループにリストアされます。

リストアされた制御ファイルの名前をメモします。このファイル名は次の手順で使用します。

```
$ rman target /
RMAN> set dbid=<DBID>; RMAN>
run
{
```

```
allocate channel ch1 device type 'SBT_TAPE' parms
'SBT_LIBRARY=/var/opt/oracle/dbaas_acfs/<DB_NAME>/opc/libopc.so,
ENV=(OPC_PFILE=/var/opt/oracle/dbaas_acfs/<DB_NAME>/opc/opc<DB_NAME>.ora);
RESTORE CONTROLFILE FROM AUTOBACKUP;
}
```

d) pfileのデータベース・パラメータを新しい制御ファイルの値で更新します。

以下のpfileパラメータを更新する必要があります。

control_file	上のrestore control fileコマンドで生成された名前を使用
--------------	---------------------------------------

<ORACLE_HOME>/dbs/init<DB_NAME>.oraにリストアされたpfileを編集し、上記のパラメータを更新します。

e) spfileを作成し、データベース・クラスタ・リソースをspfileのディレクトリ/名前<SPFILE_VAL>で更新します。

```
Create spfile and shutdown database
$ sqlplus / as sysdba
SQL> create spfile from pfile='<ORACLE_HOME>/dbs/init<DB_NAME>.ora';
Shutdown database
SQL> shutdown immediate
```

<SPFILE_VAL>の値を確認するには、*asmcmd ls -lt*を使用して、
+<DATA_DG>/spfile<DB_NAME>1.oraからリンクされたファイルを見つけます。

spfileの形式は次のようになります。

+<DATA_DG>/<DB_UNIQUE_NAME>/PARAMETERFILE/spfile.<num>.<inc>
gridユーザーとして次のコマンドを実行します。

```
# asmcmd ls -lt +<DATA_DG>/spfile<DB_NAME>1.ora
```

上で見つけた場所にspfileを設定して、データベース・クラスタを変更します。変更が終わったら、マウント・モードでデータベースを起動します。

```
Modify database cluster resources
$ srvctl modify database -d <DB_UNIQUE_NAME> -spfile <SPFILE_VAL>
Mount database
$ srvctl start database -d <DB_UNIQUE_NAME> -o mount
```

f) ブロック・チェンジ・トラッキングとフラッシュバック・データベースを無効化します。

```
# sqlplus / as sysdba
SQL> alter database disable block change tracking;
SQL> alter database flashback off;
```

g) バックアップ・ピースをカタログに追加します。

OSSにバックアップされたすべてのピースをカタログに追加します。

エディタを使用して、oracleユーザーとして次に示すスクリプトcatalog_pieces.shを作成します。実行権限を設定し、スクリプトを実行して正しい引数を指定します。

```

#@/bin/bash
if (($# != 5)); then
    echo "usage: $0 odbsrmt_dir_location oss_url oss_username oss_password oss_bucket"
    exit
fi
dir_loc=$1
[[ -d ${dir_loc} ]] || { echo "${dir_loc} is not a directory or does not exist"; exit; }
oss_url=$2; oss_username=$3;oss_password=$4;oss_bucket=$5
[[ -f ${dir_loc}/odbsrmt.py ]] || { echo "odbsrmt.py not in ${dir_loc}"; exit; }
sqlplus -s / as sysdba << EOT > /dev/null
exit;
EOT
if [[ $? -ne 0 ]]; then
    echo "unable to run sqlplus"
    exit
fi
rm -f ${dir_loc}/rep_report
python2 ${dir_loc}/odbsrmt.py --mode report --ocitype swift --host ${oss_url} --dir ${dir_loc} --credential
${oss_username}/${oss_password} --container ${oss_bucket} --dbid 0 --thread 32 --format text --forcename
rep_report || { echo "odbsrmt.py failed from ${dir_loc}"; exit; }
sed -i 'N;s/\n/ /' ${dir_loc}/rep_report || { echo "sed failed against ${dir_loc}"; exit; }
rm -f ${dir_loc}/ctl_info.dat
sqlplus -s / as sysdba << EOT > /dev/null
set head off
set echo off
set term off
set feed off
set linesize 999
set pagesize 0
set trimspool on
spool ${dir_loc}/ctl_info.dat
select handle from v$backup_piece where handle like 'c-%' and handle not like 'c-%-dbaastools' order by 1;
spool off;
exit;
EOT
if [[ $? -ne 0 ]]; then
    echo "unable to run sqlplus to gather controlfile details"
    exit
fi
ctl_line=$( tail -2 ${dir_loc}/ctl_info.dat | head -1 )
ctl_line=$(echo $ctl_line)
IFS=$'\n' read -r -d "" -a cat_array <<( tail -n +2 ${dir_loc}/rep_report | sort -k6,7 | awk
"/$ctl_line/"',EOF' | awk '{print $1}')
len=${#cat_array[@]}
rm -f ${dir_loc}/catlist.rman
printf '%s' "catalog device type 'sbt_tape' backuppiece " > ${dir_loc}/catlist.rman
for catalog_piece in ${cat_array[@]:1:$len - 2}; do
    printf '%s\n' "${catalog_piece}"," >> ${dir_loc}/catlist.rman
done
printf '%s\n' "${cat_array[$len-1]}";" >> ${dir_loc}/catlist.rman

```

oracleユーザーとして権限を設定します。

```
$chmod 755 /home/oracle/tmp<DB_NAME>/catalog_pieces.sh
```

スクリプトcatalog_pieces.shで使用するパラメータ

パラメータの位置	使用法
1	/var/opt/oracle/dbaas_acfs/<DB_NAME>/opc
2	<REPL_REGION_SWIFT_URL>
3	<REPL_REGION_OSS_USERNAME>
4	<REPL_REGION_OSS_PASSWORD>
5	<REPL_REGION_BUCKET>

必要なパラメータを指定し、oracleユーザーとしてスクリプトを実行します。

注：レプリケートされたバケット内にあるオブジェクトの数によっては、スクリプトの実行に数分かかる場合があります。

```
$/home/oracle/tmp<DB_NAME>/catalog_pieces.sh <p1> <p2> <p3> <p4> <p5>
```

生成されたRMANスクリプトcatlist.rmanをoracleユーザーとして実行します。

```
$ rman target /  
RMAN> @<dir_loc>/catlist.rman
```

h) データベース・リストアを検証します。

ポイント・イン・タイム・リカバリ（PITR）の場合は、[付録A](#)を参照してください。

```
$ rman target / RMAN>  
run  
{  
allocate channel sbt1 device type 'SBT_TAPE' parms  
"SBT_LIBRARY=/var/opt/oracle/dbaas_acfs/<DB_NAME>/opc/libopc.so,  
ENV=(OPC_PFILE=/var/opt/oracle/dbaas_acfs/<DB_NAME>/opc/opc<DB_NAME>.ora)";  
RESTORE DATABASE PREVIEW VALIDATE HEADER;  
}
```

i) ターゲット・データベースをリストアおよびリカバリします。

データベースをリストアおよびリカバリするには、すべてのOracle RACインスタンス間でRMANチャネルを分散します。

これは次の条件下で実施できます。

- RMANのターゲット接続が、SCANを使用してTNSエイリアス経由で行われている
- RMANのSBTチャネルが、SCANを介してすべてのクラスタ・インスタンスに割り当てられている

```
rman target sys/<SYS_PASSWORD>@<DB_UNIQUE_NAME> run  
{  
allocate channel sbt1 device type 'SBT_TAPE' parms  
"SBT_LIBRARY=/var/opt/oracle/dbaas_acfs/<DB_NAME>/opc/libopc.so,  
ENV=(OPC_PFILE=/var/opt/oracle/dbaas_acfs/<DB_NAME>/opc/opc<DB_NAME>.ora)";
```

```

allocate channel sbt2 device type 'SBT_TAPE' parms
"SBT_LIBRARY=/var/opt/oracle/dbaas_acfs/<DB_NAME>/opc/libopc.so,
ENV=(OPC_PFILE=/var/opt/oracle/dbaas_acfs/<DB_NAME>/opc/opc<DB_NAME>.ora)";
allocate channel sbt<n> device type 'SBT_TAPE' parms
"SBT_LIBRARY=/var/opt/oracle/dbaas_acfs/<DB_NAME>/opc/libopc.so,
ENV=(OPC_PFILE=/var/opt/oracle/dbaas_acfs/<DB_NAME>/opc/opc<DB_NAME>.ora)";
RESTORE DATABASE;
RECOVER DATABASE UNTIL AVAILABLE REDO;
release channel sbt1;
release channel sbt2;
release channel sbt<n>;
}

```

j) **resetlogs**を使用してデータベースをオープンします（SQLPLUS）。

- a. データベースをシャットダウンする（SRVCTL）
- b. マウント・モードで起動する（SQL*Plus）
- c. RESETLOGSを使用してデータベースをオープンする

RESETLOGSでデータベースをオープンする場合、開始できるのは1つのインスタンスのみです。

```

$ srvctl stop database -d <DB_UNIQUE_NAME>
$ sqlplus / as sysdba
SQL> startup mount;
SQL> alter database open resetlogs;
SQL> exit
$ start database -d <DB_UNIQUE_NAME>

```

手順7：データベース・リカバリ後の手順

a) データベースで、以下のリカバリ後ウォレット・タスクを実行します。

TDE自動ログインのパスワード（必須）（dbaascli）

次のコマンドを実行して、自動化に必要なTDEパスワードがメタデータ間で同期されていることを確認します。

```
dbaascli tde changepassword --dbname <DB_NAME>
```

本番システムのTDE自動ログイン・パスワードを指定します。

プロンプトが表示されるたびに、古いパスワード、新しいパスワード、再入力として同じパスワードを入力します。

オプション：同じコマンドを使用して、TDE自動ログイン・ウォレットのTDEパスワードを変更することもできます。

TDEマスターキーのローテーション（オプション）（dbaascli）

マスターキーを変更するには、次のコマンドを実行します。

```
dbaascli tde rotate masterkey --dbname <DB_NAME>
```

TDEウォレットの検証（必須）（dbaascli）

ウォレットを検証するには、次のコマンドを実行します。

```
dbaascli tde status --dbname <DB_NAME>
```

b) 一時表領域TEMPのファイルをチェックし、追加します。

リストアされたデータベースをオープンした後で、既存の一時表領域を確認し、必要な場合は一時ファイルを追加します。

c) PDBのステータスを確認します。

データベース・サービスの起動後に、PDBがマウントされていることを確認します。必要な場合は手動でマウントします。

```
$ sqlplus / as sysdba
SQL> select INST_ID,CON_ID,NAME,OPEN_MODE from gv$pdb$cs order by INST_ID,CON_ID;
SQL> alter pluggable database all OPEN instances=all;
```

d) コントロール・プレーンでのPDBメタデータ。

コントロール・プレーンで正しいPDB情報を表示するには、最初に作成されたPDBのPDBTEMPを削除します。

コンソールで、PDBTEMPのステータスが失敗として表示されます。コンソールを使用して、PDBの右側にあるケバブ・メニュー（縦に並んだ3つのドット）から「Delete」オプションを選択します。

コントロール・プレーンのメタデータは通常、データベースをリストアした数分後に同期されますが、同期プロセスのために遅延が数時間に及ぶ可能性もあります。

e) PDBサービス（クラスタ内のリソース・チェック）。

ターゲット・データベースで、PDBとPDBサービスを確認し、PDBのステータスを調整します。

データベースでサービスとPDBをチェックするには、次のコマンドを実行します。

```
SQL> SELECT NAME, PDB FROM V$SERVICES ORDER BY PDB,NAME;
```

クラスタでデータベースとPDBサービス・リソースをチェックするには、次のコマンドを実行します。

```
$ srvctl config service database -d <DB_UNIQUE_NAME>
```

データベースのクラスタ・リソースにPDBごとのサービスがあることを確認します。<PDB_NAME>.paas.oracle.com

必要な場合は、次のコマンドを実行してPDBサービスを追加します。

```
Add service
$ srvctl add service -db <DB_UNIQUE_NAME> -service <PDB_NAME>.<domain> -pdb <PDB_NAME> -role PRIMARY - preferred
<instance1,instance2,... , instanceN>
```

新しいPDBサービスを開始するには、次のコマンドを実行します。

```
$ srvctl start service -db <DB_UNIQUE_NAME>
$ srvctl status service -db <DB_UNIQUE_NAME>
```

f) ブロック・チェンジ・トラッキング（BCT）を有効化します。

リストアされたデータベースのBCTステータスが有効になっているかどうかを確認します。有効でない場合は、有効化します。

データベースでBCTの状態を確認するには、次のコマンドを実行します。

```
SQL> select * from v$block_change_tracking;
```

データベースでBCTを有効化するには、次のコマンドを実行します。

```
SQL> alter database enable block change tracking;
```

g) フラッシュバック・データベースを有効化します。

リストアされたデータベースのフラッシュバック状態が有効になっているかどうかを確認します。無効である場合は有効化します。

データベースのフラッシュバック状態をチェックするには、次のコマンドを実行します。

```
SQL> SELECT FLASHBACK_ON FROM V$DATABASE
```

データベースのフラッシュバックを有効化するには、次のコマンドを実行します。

```
$srvctl stop database -d <DB_UNIQUE_NAME> -o immediate
$sqlplus / as sysdba
SQL> startup mount
SQL> alter database flashback on;
SQL> shutdown immediate;
$srvctl start database -d <DB_UNIQUE_NAME>
```

h) 新しくリストアされたデータベースのバックアップを構成します。

ターゲット・データベースのバックアップ先を同じOSSバケット位置に設定することで、新しくリストアされたデータベースが以前のバックアップを使用できるようになります。これができるのは、リストアされたデータベースのDBIDがソース・データベースと同じだからです。

データベースのバックアップを設定する前に、バケットの読み取り/書き込みを有効化する必要があります。

バックアップを設定するには、以下の手順を実行します。

- 1) OSSドキュメントのレプリケーションの停止に関するセクションに従って、<REPL_OSS_BUCKET>の読み取り/書き込みを有効にします。
- 2) ExaDB-Dドキュメントの[ユーザー構成バックアップ](#)に関するセクションに従い、<REMOTE_REGION_*>の値を使用します。このガイドの「リージョン間レプリケーションのための初期構成」セクションの[表1](#)を参照してください。

結論

1つのデータベースがある場合でも、多数のデータベースを1つまたは複数のExaDB-Dシステムに統合している場合でも、Oracle Object Storageのリージョン間レプリケーションを利用すると、データ損失やリカバリ・ポイント目標（RPO）を最小限に抑えた低コストのディザスタリカバリ・ソリューションを実現できます。

付録A：タイムスタンプを使用したポイント・イン・タイム・リカバリ

a) ポイント・イン・タイム・リカバリ

タイムスタンプ：<TIMESTAMP_VAL>

ある特定の時点にデータベースをリカバリする必要がある場合は、適切なTIMESTAMP_VALを指定します。

指定する<TIMESTAMP_VAL>の値は、ソース・データベースのタイムゾーンに従います。つまり、ソース・データベースのリージョンがEST（東部標準時）で、レプリケート先のリージョンがPST（太平洋標準時）である場合、レプリケート先リージョンでのPITRに指定するタイムスタンプ値はESTになります。

b) PITRのオプション：プレビュー/ヘッダー検証を使用したリストア・データベースの検証

タイムスタンプ形式：

```
$ rman target / RMAN>
run
{
set until time "to_date('<TIMESTAMP_VAL>', 'YYYY-MM-DD HH24:MI:SS')";
allocate channel sbt1 device type 'SBT_TAPE' parms
"SBT_LIBRARY=/var/opt/oracle/dbaas_acfs/<DB_NAME>/opc/libopc.so,
ENV=(OPC_PFILE=/var/opt/oracle/dbaas_acfs/<DB_NAME>/opc/opc<DB_NAME>.ora)";
RESTORE DATABASE PREVIEW VALIDATE HEADER;
}
```

c) PITRのオプション：ターゲット・データベースのリストアおよびリカバリ

リストア・データベースのプレビューが問題なく終了したら、同じ<TIMESTAMP_VAL>を使用してデータベースのリストアとリカバリを実行します。データベースをリストアおよびリカバリするには、すべてのOracle RACインスタンス間でRMANチャンネルを分散します。これは次の条件下で実施できます。

- RMANのターゲット接続が、SCANを使用してTNSエイリアス経由で行われている
- RMANのSBTチャンネルが、SCANを介してすべてのクラスタ・インスタンスに割り当てられている

```
If Restore database preview finishes successfully then restore and recover the database using the same
<TIMESTAMP_VAL>.
$ rman target / RMAN>
run
{
set until time "to_date('<TIMESTAMP_VAL>', 'YYYY-MM-DD HH24:MI:SS')";
allocate channel sbt1 device type 'SBT_TAPE' parms
"SBT_LIBRARY=/var/opt/oracle/dbaas_acfs/<DB_NAME>/opc/libopc.so,
ENV=(OPC_PFILE=/var/opt/oracle/dbaas_acfs/<DB_NAME>/opc/opc<DB_NAME>.ora)";
allocate channel sbt2 device type 'SBT_TAPE' parms
"SBT_LIBRARY=/var/opt/oracle/dbaas_acfs/<DB_NAME>/opc/libopc.so,
ENV=(OPC_PFILE=/var/opt/oracle/dbaas_acfs/<DB_NAME>/opc/opc<DB_NAME>.ora)";
allocate channel sbt<n> device type 'SBT_TAPE' parms
"SBT_LIBRARY=/var/opt/oracle/dbaas_acfs/<DB_NAME>/opc/libopc.so,
ENV=(OPC_PFILE=/var/opt/oracle/dbaas_acfs/<DB_NAME>/opc/opc<DB_NAME>.ora)";
RESTORE DATABASE;
RECOVER DATABASE;
release channel sbt1;
release channel sbt2;
release channel sbt<n>;
}
```

手順6.j) RESETLOGSを使用してデータベースをオープンするに進みます。

```
rman target sys/<SYS_PASSWORD>@<DB_UNIQUE_NAME> run {
set until time "to_date('<TIMESTAMP_VAL>', 'YYYY-MM-DD HH24:MI:SS')";
allocate channel sbt1 device type 'SBT_TAPE' parms
"SBT_LIBRARY=/var/opt/oracle/dbaas_acfs/<DB_NAME>/opc/libopc.so,
ENV=(OPC_PFILE=/var/opt/oracle/dbaas_acfs/<DB_NAME>/opc/opc<DB_NAME>.ora)";
allocate channel sbt2 device type 'SBT_TAPE' parms
"SBT_LIBRARY=/var/opt/oracle/dbaas_acfs/<DB_NAME>/opc/libopc.so,
ENV=(OPC_PFILE=/var/opt/oracle/dbaas_acfs/<DB_NAME>/opc/opc<DB_NAME>.ora)";
allocate channel sbt<n> device type 'SBT_TAPE' parms
"SBT_LIBRARY=/var/opt/oracle/dbaas_acfs/<DB_NAME>/opc/libopc.so,
ENV=(OPC_PFILE=/var/opt/oracle/dbaas_acfs/<DB_NAME>/opc/opc<DB_NAME>.ora)";
RESTORE DATABASE;
RECOVER DATABASE UNTIL AVAILABLE REDO;
release channel sbt1;
release channel sbt2;
release channel sbt<n>;
}
```

付録B : 完全なレベル0

ローカル・リージョンのバックアップ開始後にレプリケーションを有効化した場合、既存のバックアップ・ピースはどれもレプリケートされません。しかし、レプリケーションを有効化した後に取得されたバックアップはレプリケートされます。この動作により、完全なレベル0のデータベース・バックアップが必要になります。このバックアップには、すべての読取り専用表領域が含まれている必要があります。

バックアップの最適化を無効化し、dbaascliでレベル0のバックアップを作成してから、バックアップの最適化を有効化すると、上の条件を満たすことができます。

レプリケーションを有効化して完全なレベル0を実行した日時よりも前に、レプリケートされたバケットからリストアすることはできません。

oracleユーザーとして次のコマンドを実行します。

```
rman target /
RMAN> CONFIGURE BACKUP OPTIMIZATION off;
exit
```

rootユーザーとして次のコマンドを実行します。

```
dbaascli database backup --dbname <value> --start --level 0
```

oracleユーザーとして次のコマンドを実行します。

```
rman target /
RMAN> CONFIGURE BACKUP OPTIMIZATION on;
exit
```

Connect with us

+1.800.ORACLE1までご連絡いただくか、[oracle.com](https://www.oracle.com)をご覧ください。北米以外の地域では、[oracle.com/contact](https://www.oracle.com/contact)で最寄りの営業所をご確認いただけます。

 blogs.oracle.com

 facebook.com/oracle

 twitter.com/oracle

Copyright © 2022, Oracle and/or its affiliates. All rights reserved. 本文書は情報提供のみを目的として提供されており、ここに記載されている内容は予告なく変更されることがあります。本文書は、その内容に誤りがないことを保証するものではなく、また、口頭による明示的保証や法律による黙示的保証を含め、商品性ないし特定目的適合性に関する黙示的保証および条件などのいかなる保証および条件も提供するものではありません。オラクルは本文書に関するいかなる法的責任も明確に否認し、本文書によって直接的または間接的に確立される契約義務はないものとします。本文書はオラクルの書面による許可を前もって得ることなく、いかなる目的のためにも、電子または印刷を含むいかなる形式や手段によっても再作成または送信することはできません。

本デバイスは、連邦通信委員会のルールに基づいた認可を未取得です。認可を受けるまでは、このデバイスの販売またはリースを提案することも、このデバイスを販売またはリースすることはありません。

OracleおよびJavaはOracleおよびその子会社、関連会社の登録商標です。その他の名称はそれぞれの会社の商標です。

IntelおよびIntel XeonはIntel Corporationの商標または登録商標です。すべてのSPARC商標はライセンスに基づいて使用されるSPARC International, Inc.の商標または登録商標です。AMD、Opteron、AMDロゴおよびAMD Opteronロゴは、Advanced Micro Devicesの商標または登録商標です。UNIXは、The Open Groupの登録商標です。0120

免責事項：データシートにこの免責事項の記載が必要かどうか分からない場合は、収益認識方針を参照してください。本書の内容と免責事項の要件についてさらに質問がある場合は、REVREG_US@oracle.com宛てに電子メールでご連絡ください。

