



TR-4981: Amazon FSx ONTAPによる Oracle Active Data Guard のコスト削減

NetApp database solutions

NetApp
August 18, 2025

目次

TR-4981: Amazon FSx ONTAPによる Oracle Active Data Guard のコスト削減	1
目的	1
観客	1
ソリューションのテストおよび検証環境	2
アーキテクチャ	2
ハードウェアおよびソフトウェアコンポーネント	2
仮想的なニューヨークからロサンゼルスへの DR セットアップによる Oracle Data Guard 構成	3
導入検討の重要な要素	3
ソリューションの展開	4
展開の前提条件	4
Data Guard用のプライマリデータベースを準備する	5
スタンバイデータベースを準備し、Data Guardをアクティブ化する	12
Data Guard Brokerのセットアップ	21
他のユースケース用にスタンバイ データベースをクローンする	24
詳細情報の入手方法	38

TR-4981: Amazon FSx ONTAPによる Oracle Active Data Guard のコスト削減

アレン・カオ、ニヤズ・モハメド、NetApp

このソリューションでは、AWS FSx ONTAP をスタンバイサイトの Oracle データベースストレージとして使用して Oracle Data Guard を構成し、AWS での Oracle Data Guard HA/DR ソリューションのライセンスおよび運用コストを削減するための概要と詳細を示します。

目的

Oracle Data Guard は、プライマリ データベースとスタンバイ データベースのレプリケーション構成におけるエンタープライズ データの高可用性、データ保護、および災害復旧を保証します。Oracle Active Data Guard により、プライマリ データベースからスタンバイ データベースへのデータ レプリケーションがアクティブな状態でも、ユーザーはスタンバイ データベースにアクセスできるようになります。Data Guard は、Oracle Database Enterprise Edition の機能です。別途ライセンスは必要ありません。一方、Active Data Guard は Oracle Database Enterprise Edition オプションであるため、別途ライセンスが必要です。Active Data Guard セットアップでは、複数のスタンバイ データベースがプライマリ データベースからデータ レプリケーションを受信できます。ただし、追加のスタンバイ データベースごとに、Active Data Guard ライセンスと、プライマリ データベースのサイズと同じ追加のストレージが必要になります。運用コストはすぐに増加します。

Oracle データベース運用のコストを削減することに熱心で、AWS に Active Data Guard を設定する予定の場合は、代替案を検討する必要があります。Active Data Guard の代わりに、Data Guard を使用して、プライマリデータベースからAmazon FSx ONTAPストレージ上の単一の物理スタンバイデータベースにレプリケートします。その後、このスタンバイ データベースの複数のコピーをクローン化し、読み取り/書き込みアクセス用に開いて、レポート、開発、テストなどの他の多くのユース ケースに使用できます。その結果、Active Data Guard の機能が効果的に提供されると同時に、追加のスタンバイ データベースごとに Active Data Guard ライセンスと追加のストレージ コストが不要になります。このドキュメントでは、AWS の既存のプライマリデータベースを使用して Oracle Data Guard を設定し、Amazon FSx ONTAPストレージに物理スタンバイデータベースを配置する方法を説明します。スタンバイ データベースはスナップショットを介してバックアップされ、必要に応じてユースケースの読み取り/書き込みアクセス用にクローン化されます。

このソリューションは、次のユースケースに対応します。

- AWS の任意のストレージ上のプライマリデータベースとAmazon FSx ONTAPストレージ上のスタンバイデータベース間の Oracle Data Guard。
- レポート、開発、テストなどのユースケースに対応するために、データ レプリケーションのために閉じた状態でスタンバイ データベースのクローンを作成します。

観客

このソリューションは次の人々を対象としています。

- 高可用性、データ保護、災害復旧のために AWS に Oracle Active Data Guard をセットアップした DBA。
- AWS クラウドでの Oracle Active Data Guard 構成に関心のあるデータベース ソリューション アーキテクト。

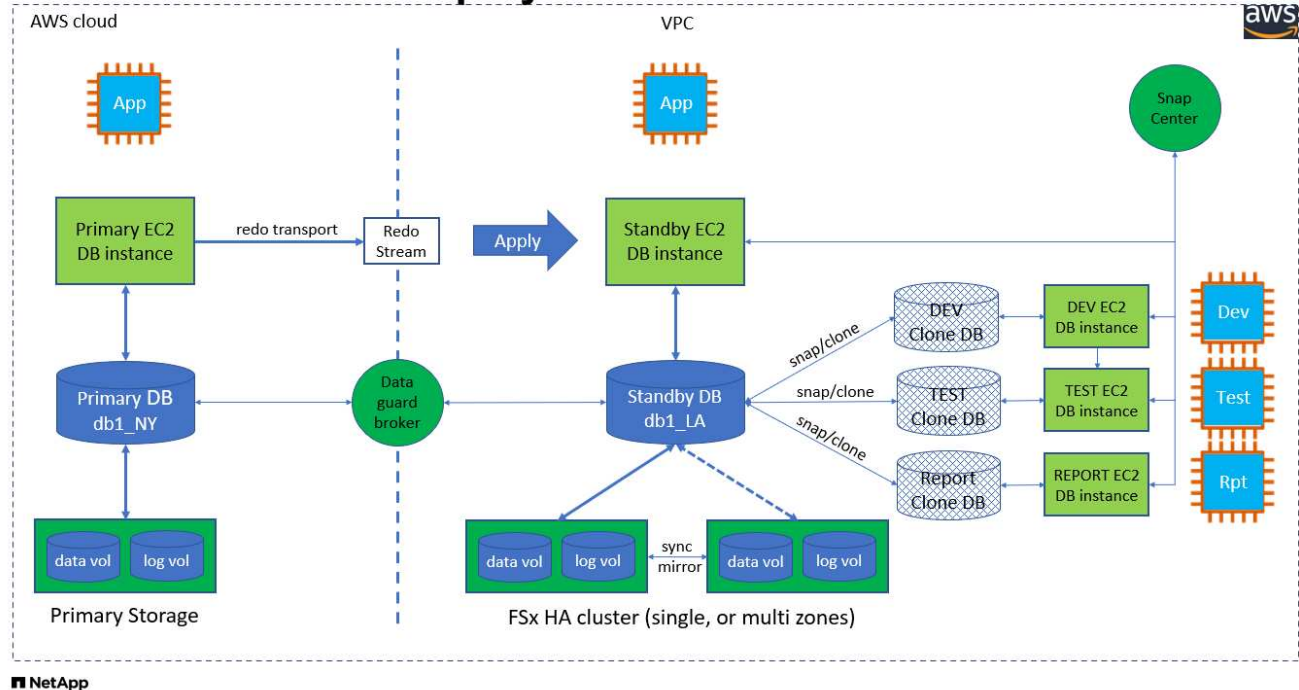
- Oracle Data Guard をサポートする AWS FSx ONTAPストレージを管理するストレージ管理者。
- AWS FSx/EC2 環境で Oracle Data Guard を立ち上げたいアプリケーション所有者。

ソリューションのテストおよび検証環境

このソリューションのテストと検証は、最終的な展開環境と一致しない可能性のある AWS FSx ONTAPおよび EC2 ラボ環境で実行されました。詳細については、セクションをご覧ください。[導入検討の重要な要素]

アーキテクチャ

Oracle Data Guard Deployment with Amazon FSx for ONTAP



ハードウェアおよびソフトウェアコンポーネント

ハードウェア		
FSx ONTAPストレージ	AWSが提供する現在のバージョン	同じ VPC およびアベイラビリティゾーン内の 1 つの FSx HA クラスター
コンピューティング用のEC2インスタンス	t2.xlarge/4vCPU/16G	3 つの EC2 T2 xlarge EC2 インスタンス (1 つはプライマリ DB サーバー、1 つはスタンバイ DB サーバー、3 つ目はクローン DB サーバー)
ソフトウェア		
レッドハットリナックス	RHEL-8.6.0_HVM-20220503-x86_64-2-Hourly2-GP2	テスト用にRedHatサブスクリプションを導入

Oracle グリッド・インフラストラクチャ	バージョン19.18	RUパッチ チp34762026_190000_Linux-x86-64.zipを適用しました
Oracle Database	バージョン19.18	RUパッチ チp34765931_190000_Linux-x86-64.zipを適用しました
Oracle OPatch	バージョン 12.2.0.1.36	最新パッチ p6880880_190000_Linux-x86-64.zip

仮想的なニューヨークからロサンゼルスへの **DR** セットアップによる **Oracle Data Guard** 構成

データベース	DB_UNIQUE_NAME	Oracle ネットサービス名
プライマリ	db1_NY	db1_NY.demo.netapp.com
物理スタンバイ	db1_LA	db1_LA.demo.netapp.com

導入検討の重要な要素

- **Oracle** スタンバイ データベース**FlexClone** の仕組み。AWS FSx ONTAP FlexClone は、書き込み可能な同じスタンバイデータベースボリュームの共有コピーを提供します。ボリュームのコピーは、実際には、クローン上で新しい書き込みが開始されるまで、元のデータ ブロックにリンクするポインタです。その後、ONTAP は新しい書き込み用に新しいストレージ ブロックを割り当てます。すべての読み取り IO は、アクティブなレプリケーションの下で元のデータ ブロックによって処理されます。したがって、クローンのストレージ効率は非常に高く、新しい書き込み IO に対して最小限かつ増分的な新しいストレージ割り当てで、他の多くのユースケースに使用できます。これにより、Active Data Guard のストレージ フットプリントが大幅に削減され、ストレージ コストが大幅に削減されます。NetApp、Oracle のパフォーマンスを高いレベルで維持するために、データベースがプライマリ ストレージからスタンバイ FSx ストレージに切り替わる場合、FlexCloneアクティビティを最小限に抑えることを推奨しています。
- *Oracle ソフトウェア要件*通常、フィジカルスタンバイデータベースは、Oracle Data Guard Standby-First Patch Applyプロセスが進行中でない限り（My Oracle Supportノート1265700.1に記載されているように）、パッチセット例外（PSE）、クリティカルパッチアップデート（CPU）、パッチセットアップデート（PSU）を含め、プライマリデータベースと同じデータベースホームバージョンである必要があります。["サポート"](#)
- *スタンバイ データベースのディレクトリ構造に関する考慮事項*可能であれば、プライマリ システムとスタンバイ システムのデータ ファイル、ログ ファイル、および制御ファイルの名前とパス名を同じにし、Optimal Flexible Architecture (OFA) 命名規則を使用する必要があります。スタンバイ データベース上のアーカイブ ディレクトリも、サイズや構造を含め、サイト間で同一である必要があります。この戦略により、バックアップ、スイッチオーバー、フェイルオーバーなどの他の操作でも同じ手順を実行できるようになり、メンテナンスの複雑さが軽減されます。
- *強制ログモード。*スタンバイ データベースに伝播できないプライマリ データベースでのログに記録されない直接書き込みを防ぐには、スタンバイ作成用のデータ ファイルのバックアップを実行する前に、プライマリ データベースで FORCE LOGGING をオンにします。
- *データベース ストレージ管理*操作を簡素化するために、Oracle Data Guard 構成で Oracle Automatic Storage Management (Oracle ASM) と Oracle Managed Files (OMF) を設定する場合は、プライマリ データベースとスタンバイ データベースで対称的に設定することをお勧めします。

- ***EC2 コンピューティングインスタンス。***これらのテストと検証では、Oracle データベース コンピューティング インスタンスとして AWS EC2 t2.xlarge インスタンスを使用しました。NetApp、データベース ワークロードに最適化されているため、本番環境での Oracle のコンピューティング インスタンスとして M5 タイプの EC2 インスタンスを使用することを推奨しています。実際のワークロード要件に基づいて、vCPU の数と RAM の量に応じて EC2 インスタンスのサイズを適切に設定する必要があります。
- ***FSx ストレージ HA クラスターの単一またはマルチゾーン展開。***これらのテストと検証では、単一の AWS アベイラビリティゾーンに FSx HA クラスターをデプロイしました。本番環境での導入では、NetApp は2つの異なるアベイラビリティゾーンに FSx HA ペアを導入することを推奨しています。FSx クラスターは常に、アクティブ/パッシブ ファイル システムのペアで同期ミラーリングされた HA ペアでプロビジョニングされ、ストレージ レベルの冗長性を提供します。マルチゾーン展開により、単一の AWS ゾーンで障害が発生した場合でも高可用性がさらに強化されます。
- **FSx ストレージ クラスターのサイズ設定。**Amazon FSx ONTAPストレージファイルシステムは、最大 160,000 の生の SSD IOPS、最大 4GBps のスループット、最大 192TiB の容量を提供します。ただし、デプロイ時の実際の要件に基づいて、プロビジョニングされた IOPS、スループット、およびストレージ制限 (最小 1,024 GiB) の観点からクラスターのサイズを決定できます。アプリケーションの可用性に影響を与えることなく、容量を動的に調整できます。

ソリューションの展開

Data Guard をセットアップするための開始点として、VPC 内の AWS EC2 環境にプライマリ Oracle データベースがすでにデプロイされているものと想定されます。プライマリ データベースは、ストレージ管理用に Oracle ASM を使用して展開されます。Oracle データファイル、ログファイル、制御ファイルなどのために、2つの ASM ディスク グループ (+DATA と +LOGS) が作成されます。ASM を使用した AWS での Oracle のデプロイメントの詳細については、次の技術レポートを参照してください。

- ["EC2 および FSx での Oracle データベースのデプロイメントのベストプラクティス"](#)
- ["AWS FSx/EC2 における iSCSI/ASM を使用した Oracle データベースのデプロイメントと保護"](#)
- ["AWS FSx/EC2 で NFS/ASM を使用して Oracle 19c をスタンドアロンで再起動する"](#)

プライマリ Oracle データベースは、FSx ONTAPまたは AWS EC2 エコシステム内の任意のストレージ上で実行できます。次のセクションでは、ASM ストレージを備えたプライマリ EC2 DB インスタンスと ASM ストレージを備えたスタンバイ EC2 DB インスタンス間に Oracle Data Guard を設定するためのデプロイメント手順を段階的に説明します。

展開の前提条件

展開には次の前提条件が必要です。

1. AWS アカウントが設定され、必要な VPC とネットワークセグメントが AWS アカウント内に作成されています。
2. AWS EC2 コンソールから、少なくとも 3 つの EC2 Linux インスタンスをデプロイする必要があります。1 つはプライマリ Oracle DB インスタンス、1 つはスタンバイ Oracle DB インスタンス、そしてレポート、開発、テストなどのためのクローンターゲット DB インスタンスです。環境設定の詳細については、前のセクションのアーキテクチャ図を参照してください。AWS も確認しましょう "[Linux インスタンスのユーザーガイド](#)" 詳細についてはこちらをご覧ください。
3. AWS EC2 コンソールから、Amazon FSx ONTAP ストレージ HA クラスターをデプロイして、Oracle スタンバイデータベースを格納する Oracle ボリュームをホストします。FSx ストレージの導入に慣れていない場合は、ドキュメントを参照してください。"[FSx ONTAP ファイルシステムの作成](#)" ステップバイステップの手順については、こちらをご覧ください。
4. ステップ 2 と 3 は、次の Terraform 自動化ツールキットを使用して実行できます。このツールキットは、次の名前の EC2 インスタンスを作成します。ora_01 FSx ファイルシステムは fsx_01。実行する前に、手順を注意深く確認し、環境に合わせて変数を変更してください。テンプレートは、独自の展開要件に合わせて簡単に変更できます。

```
git clone https://github.com/NetApp-  
Automation/na_aws_fsx_ec2_deploy.git
```



Oracle インストール ファイルをステージングするための十分なスペースを確保するために、EC2 インスタンスのルート ボリュームに少なくとも 50G が割り当てられていることを確認します。

Data Guard用のプライマリデータベースを準備する

このデモでは、プライマリ EC2 DB インスタンスに db1 というプライマリ Oracle データベースをセットアップし、スタンドアロンの再起動構成で 2 つの ASM ディスク グループをセットアップしました。データ ファイルは ASM ディスク グループ +DATA に、フラッシュ リカバリ領域は ASM ディスク グループ +LOGS に含まれています。以下に、Data Guard のプライマリ データベースを設定するための詳細な手順を示します。すべての手順は、データベース所有者 (Oracle ユーザー) として実行する必要があります。

1. プライマリ EC2 DB インスタンス ip-172-30-15-45 上のプライマリ データベース db1 構成。ASM ディスク グループは、EC2 エコシステム内のあらゆるタイプのストレージに配置できます。

```
[oracle@ip-172-30-15-45 ~]$ cat /etc/oratab

# This file is used by ORACLE utilities.  It is created by root.sh
# and updated by either Database Configuration Assistant while
creating
# a database or ASM Configuration Assistant while creating ASM
instance.

# A colon, ':', is used as the field terminator.  A new line
terminates
# the entry.  Lines beginning with a pound sign, '#', are comments.
#
# Entries are of the form:
#   $ORACLE_SID:$ORACLE_HOME:<N|Y>:
#
# The first and second fields are the system identifier and home
# directory of the database respectively.  The third field indicates
# to the dbstart utility that the database should , "Y", or should
not,
# "N", be brought up at system boot time.
#
# Multiple entries with the same $ORACLE_SID are not allowed.
#
#
+ASM:/u01/app/oracle/product/19.0.0/grid:N
db1:/u01/app/oracle/product/19.0.0/db1:N

[oracle@ip-172-30-15-45 ~]$
/u01/app/oracle/product/19.0.0/grid/bin/crsctl stat res -t
-----
-----
Name                Target    State          Server          State
details
-----
-----
Local Resources
```

```

-----
ora.DATA.dg
      ONLINE  ONLINE      ip-172-30-15-45      STABLE
ora.LISTENER.lsnr
      ONLINE  ONLINE      ip-172-30-15-45      STABLE
ora.LOGS.dg
      ONLINE  ONLINE      ip-172-30-15-45      STABLE
ora.asm
      ONLINE  ONLINE      ip-172-30-15-45
Started,STABLE
ora.ons
      OFFLINE OFFLINE      ip-172-30-15-45      STABLE
-----

```

Cluster Resources

```

-----
ora.cssd
      1      ONLINE  ONLINE      ip-172-30-15-45      STABLE
ora.dbf1.db
      1      ONLINE  ONLINE      ip-172-30-15-45
Open,HOME=/u01/app/o
racle/product/19.0.0
/dbf1,STABLE
ora.diskmon
      1      OFFLINE OFFLINE      STABLE
ora.driver.afd
      1      ONLINE  ONLINE      ip-172-30-15-45      STABLE
ora.evmd
      1      ONLINE  ONLINE      ip-172-30-15-45      STABLE
-----

```

2. sqlplus から、プライマリでの強制ログ記録を有効にします。

```
alter database force logging;
```

3. sqlplus から、プライマリでフラッシュバックを有効にします。フラッシュバックを使用すると、フェイルオーバー後にプライマリ データベースをスタンバイとして簡単に復元できます。

```
alter database flashback on;
```

4. Oracle パスワード ファイルを使用して REDO トランспорт認証を構成します。設定されていない場合は orapwd ユーティリティを使用してプライマリで pwd ファイルを作成し、スタンバイ データベースの \$ORACLE_HOME/dbs ディレクトリにコピーします。
5. 現在のオンライン ログ ファイルと同じサイズのスタンバイ REDO ログをプライマリ DB に作成します。ログ グループは、オンライン ログ ファイル グループより 1 つ多くなります。プライマリ データベースは、必要に応じてすぐにスタンバイ ロールに移行し、REDO データの受信を開始できます。

```
alter database add standby logfile thread 1 size 200M;
```

Validate after standby logs addition:

```
SQL> select group#, type, member from v$logfile;
```

GROUP#	TYPE	MEMBER
3	ONLINE	+DATA/DB1/ONLINELOG/group_3.264.1145821513
2	ONLINE	+DATA/DB1/ONLINELOG/group_2.263.1145821513
1	ONLINE	+DATA/DB1/ONLINELOG/group_1.262.1145821513
4	STANDBY	+DATA/DB1/ONLINELOG/group_4.286.1146082751
4	STANDBY	+LOGS/DB1/ONLINELOG/group_4.258.1146082753
5	STANDBY	+DATA/DB1/ONLINELOG/group_5.287.1146082819
5	STANDBY	+LOGS/DB1/ONLINELOG/group_5.260.1146082821
6	STANDBY	+DATA/DB1/ONLINELOG/group_6.288.1146082825
6	STANDBY	+LOGS/DB1/ONLINELOG/group_6.261.1146082827
7	STANDBY	+DATA/DB1/ONLINELOG/group_7.289.1146082835
7	STANDBY	+LOGS/DB1/ONLINELOG/group_7.262.1146082835

11 rows selected.

6. sqlplus から、編集用に spfile から pfile を作成します。

```
create pfile='/home/oracle/initdb1.ora' from spfile;
```

7. pfile を修正し、次のパラメータを追加します。

```
DB_NAME=db1
DB_UNIQUE_NAME=db1_NY
LOG_ARCHIVE_CONFIG='DG_CONFIG=(db1_NY,db1_LA) '
LOG_ARCHIVE_DEST_1='LOCATION=USE_DB_RECOVERY_FILE_DEST
VALID_FOR=(ALL_LOGFILES,ALL_ROLES) DB_UNIQUE_NAME=db1_NY'
LOG_ARCHIVE_DEST_2='SERVICE=db1_LA ASYNC
VALID_FOR=(ONLINE_LOGFILES,PRIMARY_ROLE) DB_UNIQUE_NAME=db1_LA'
REMOTE_LOGIN_PASSWORDFILE=EXCLUSIVE
FAL_SERVER=db1_LA
STANDBY_FILE_MANAGEMENT=AUTO
```

8. sqlplus から、/home/oracle ディレクトリ内の修正された pfile から ASM +DATA ディレクトリに spfile を作成します。

```
create spfile='+DATA' from pfile='/home/oracle/initdb1.ora';
```

9. +DATA ディスク グループの下に新しく作成された spfile を見つけます (必要に応じて asmcmd ユーティリティを使用します)。以下に示すように、srvctl を使用してグリッドを変更し、新しい spfile からデータベースを起動します。

```
[oracle@ip-172-30-15-45 db1]$ srvctl config database -d db1
Database unique name: db1
Database name: db1
Oracle home: /u01/app/oracle/product/19.0.0/db1
Oracle user: oracle
Spfile: +DATA/DB1/PARAMETERFILE/spfile.270.1145822903
Password file:
Domain: demo.netapp.com
Start options: open
Stop options: immediate
Database role: PRIMARY
Management policy: AUTOMATIC
Disk Groups: DATA
Services:
OSDBA group:
OSOPER group:
Database instance: db1
[oracle@ip-172-30-15-45 db1]$ srvctl modify database -d db1 -spfile
+DATA/DB1/PARAMETERFILE/spfiledb1.ora
[oracle@ip-172-30-15-45 db1]$ srvctl config database -d db1
Database unique name: db1
Database name: db1
Oracle home: /u01/app/oracle/product/19.0.0/db1
Oracle user: oracle
Spfile: +DATA/DB1/PARAMETERFILE/spfiledb1.ora
Password file:
Domain: demo.netapp.com
Start options: open
Stop options: immediate
Database role: PRIMARY
Management policy: AUTOMATIC
Disk Groups: DATA
Services:
OSDBA group:
OSOPER group:
Database instance: db1
```

10. 名前解決のために db_unique_name を追加するように tnsnames.ora を変更します。

```
# tnsnames.ora Network Configuration File:
/u01/app/oracle/product/19.0.0/db1/network/admin/tnsnames.ora
# Generated by Oracle configuration tools.

db1_NY =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP) (HOST = ip-172-30-15-
45.ec2.internal) (PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SID = db1)
    )
  )

db1_LA =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP) (HOST = ip-172-30-15-
67.ec2.internal) (PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SID = db1)
    )
  )

LISTENER_DB1 =
  (ADDRESS = (PROTOCOL = TCP) (HOST = ip-172-30-15-
45.ec2.internal) (PORT = 1521))
```

11. プライマリ データベースのデータ ガード サービス名 db1_NY_DGMGRL.demo.netapp を listener.ora ファイルに追加します。

```
#Backup file is /u01/app/oracle/crsdata/ip-172-30-15-45/output/listener.ora.bak.ip-172-30-15-45.oracle line added by Agent
# listener.ora Network Configuration File:
/u01/app/oracle/product/19.0.0/grid/network/admin/listener.ora
# Generated by Oracle configuration tools.
```

```
LISTENER =
  (DESCRIPTION_LIST =
    (DESCRIPTION =
      (ADDRESS = (PROTOCOL = TCP) (HOST = ip-172-30-15-45.ec2.internal) (PORT = 1521))
      (ADDRESS = (PROTOCOL = IPC) (KEY = EXTPROC1521))
    )
  )
```

```
SID_LIST_LISTENER =
  (SID_LIST =
    (SID_DESC =
      (GLOBAL_DBNAME = db1_NY_DGMGRL.demo.netapp.com)
      (ORACLE_HOME = /u01/app/oracle/product/19.0.0/db1)
      (SID_NAME = db1)
    )
  )
```

```
ENABLE_GLOBAL_DYNAMIC_ENDPOINT_LISTENER=ON # line added by Agent
VALID_NODE_CHECKING_REGISTRATION_LISTENER=ON # line added by Agent
```

1. `srvctl` を使用してデータベースをシャットダウンして再起動し、データ ガード パラメータがアクティブになっていることを確認します。

```
srvctl stop database -d db1
```

```
srvctl start database -d db1
```

これで、Data Guard のプライマリ データベースのセットアップが完了します。

スタンバイデータベースを準備し、**Data Guard**をアクティブ化する

Oracle Data Guard では、プライマリ EC2 DB インスタンスと一致させるために、スタンバイ EC2 DB インスタンス上の OS カーネル構成とパッチセットを含む Oracle ソフトウェア スタックが必要です。管理を容易にし、簡素化するために、スタンバイ EC2 DB インスタンスのデータベース ストレージ構成は、ASM ディスク グループの名前、数、サイズなど、プライマリ EC2 DB インスタンスと一致させるのが理想的です。以下は、Data Guard のスタンバイ EC2 DB インスタンスを設定するための詳細な手順です。すべてのコマンドは Oracle 所有者のユーザー ID として実行する必要があります。

1. まず、プライマリ EC2 インスタンス上のプライマリ データベースの構成を確認します。このデモでは、スタンドアロンの再起動構成で、2 つの ASM ディスク グループ +DATA と +LOGS を使用して、プライマリ EC2 DB インスタンス上に db1 というプライマリ Oracle データベースをセットアップしました。プライマリ ASM ディスク グループは、EC2 エコシステム内の任意のタイプのストレージ上に配置できます。
2. ドキュメントの手順に従う"[TR-4965: AWS FSx/EC2 における iSCSI/ASM を使用した Oracle データベースの導入と保護](#)"プライマリ データベースと一致するように、スタンバイ EC2 DB インスタンスにグリッドと Oracle をインストールして構成します。データベース ストレージは、プライマリ EC2 DB インスタンスと同じストレージ容量を持つ FSx ONTAPからスタンバイ EC2 DB インスタンスにプロビジョニングおよび割り当てられる必要があります。



ステップ10で停止 `Oracle database installation` セクション。スタンバイ データベースは、dbca データベース複製機能を使用してプライマリ データベースからインスタンス化されます。

3. Oracle ソフトウェアをインストールして構成したら、スタンバイ \$ORACLE_HOME dbs ディレクトリから、プライマリ データベースの Oracle パスワードをコピーします。

```
scp
oracle@172.30.15.45:/u01/app/oracle/product/19.0.0/db1/dbs/orapwdb1
.
```

4. 次のエントリを含む tnsnames.ora ファイルを作成します。

```
# tnsnames.ora Network Configuration File:
/u01/app/oracle/product/19.0.0/db1/network/admin/tnsnames.ora
# Generated by Oracle configuration tools.

db1_NY =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP) (HOST = ip-172-30-15-
45.ec2.internal) (PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SID = db1)
    )
  )

db1_LA =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP) (HOST = ip-172-30-15-
67.ec2.internal) (PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SID = db1)
    )
  )
```

5. DB データ ガード サービス名を listener.ora ファイルに追加します。

```
#Backup file is /u01/app/oracle/crsdata/ip-172-30-15-
67/output/listener.ora.bak.ip-172-30-15-67.oracle line added by
Agent
# listener.ora Network Configuration File:
/u01/app/oracle/product/19.0.0/grid/network/admin/listener.ora
# Generated by Oracle configuration tools.

LISTENER =
  (DESCRIPTION_LIST =
    (DESCRIPTION =
      (ADDRESS = (PROTOCOL = TCP) (HOST = ip-172-30-15-
67.ec2.internal) (PORT = 1521))
      (ADDRESS = (PROTOCOL = IPC) (KEY = EXTPROC1521))
    )
  )

SID_LIST_LISTENER =
  (SID_LIST =
    (SID_DESC =
      (GLOBAL_DBNAME = db1_LA_DGMGRL.demo.netapp.com)
      (ORACLE_HOME = /u01/app/oracle/product/19.0.0/db1)
      (SID_NAME = db1)
    )
  )

ENABLE_GLOBAL_DYNAMIC_ENDPOINT_LISTENER=ON # line added
by Agent
VALID_NODE_CHECKING_REGISTRATION_LISTENER=ON # line added
by Agent
```

6. Oracle ホームとパスを設定します。

```
export ORACLE_HOME=/u01/app/oracle/product/19.0.0/db1
```

```
export PATH=$PATH:$ORACLE_HOME/bin
```

7. dbca を使用して、プライマリ データベース db1 からスタンバイ データベースをインスタンス化します。

```
[oracle@ip-172-30-15-67 bin]$ dbca -silent -createDuplicateDB
-gdbName db1 -primaryDBConnectionString ip-172-30-15-
45.ec2.internal:1521/db1_NY.demo.netapp.com -sid db1 -initParams
fal_server=db1_NY -createAsStandby -dbUniqueName db1_LA
Enter SYS user password:

Prepare for db operation
22% complete
Listener config step
44% complete
Auxiliary instance creation
67% complete
RMAN duplicate
89% complete
Post duplicate database operations
100% complete

Look at the log file
"/u01/app/oracle/cfgtoollogs/dbca/db1_LA/db1_LA.log" for further
details.
```

8. 複製されたスタンバイ データベースを検証します。新しく複製されたスタンバイ データベースは、最初は READ ONLY モードで開かれます。

```
[oracle@ip-172-30-15-67 bin]$ export ORACLE_SID=db1
[oracle@ip-172-30-15-67 bin]$ sqlplus / as sysdba

SQL*Plus: Release 19.0.0.0.0 - Production on Wed Aug 30 18:25:46
2023
Version 19.18.0.0.0

Copyright (c) 1982, 2022, Oracle. All rights reserved.

Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 -
Production
Version 19.18.0.0.0

SQL> select name, open_mode from v$database;

NAME          OPEN_MODE
-----
DB1           READ ONLY
```

```
SQL> show parameter name
```

NAME	TYPE	VALUE
-----	-----	

cdb_cluster_name	string	
cell_offloadgroup_name	string	
db_file_name_convert	string	
db_name	string	db1
db_unique_name	string	db1_LA
global_names	boolean	FALSE
instance_name	string	db1
lock_name_space	string	
log_file_name_convert	string	
pdb_file_name_convert	string	
processor_group_name	string	

NAME	TYPE	VALUE
-----	-----	

service_names	string	
db1_LA.demo.netapp.com		

```
SQL>
```

```
SQL> show parameter log_archive_config
```

NAME	TYPE	VALUE
-----	-----	

log_archive_config	string	
DG_CONFIG=(db1_NY,db1_LA)		

```
SQL> show parameter fal_server
```

NAME	TYPE	VALUE
-----	-----	

fal_server	string	db1_NY

```
SQL> select name from v$datafile;
```

NAME

+DATA/DB1_LA/DATAFILE/system.261.1146248215
+DATA/DB1_LA/DATAFILE/sysaux.262.1146248231
+DATA/DB1_LA/DATAFILE/undotbs1.263.1146248247
+DATA/DB1_LA/03C5C01A66EE9797E0632D0F1EAC5F59/DATAFILE/system.264.11

```

46248253
+DATA/DB1_LA/03C5C01A66EE9797E0632D0F1EAC5F59/DATAFILE/sysaux.265.11
46248261
+DATA/DB1_LA/DATAFILE/users.266.1146248267
+DATA/DB1_LA/03C5C01A66EE9797E0632D0F1EAC5F59/DATAFILE/undotbs1.267.
1146248269
+DATA/DB1_LA/03C5EFD07C41A1FAE0632D0F1EAC9BD8/DATAFILE/system.268.11
46248271
+DATA/DB1_LA/03C5EFD07C41A1FAE0632D0F1EAC9BD8/DATAFILE/sysaux.269.11
46248279
+DATA/DB1_LA/03C5EFD07C41A1FAE0632D0F1EAC9BD8/DATAFILE/undotbs1.270.
1146248285
+DATA/DB1_LA/03C5EFD07C41A1FAE0632D0F1EAC9BD8/DATAFILE/users.271.114
6248293

```

NAME

```

-----
-----
+DATA/DB1_LA/03C5F0DDF35CA2B6E0632D0F1EAC8B6B/DATAFILE/system.272.11
46248295
+DATA/DB1_LA/03C5F0DDF35CA2B6E0632D0F1EAC8B6B/DATAFILE/sysaux.273.11
46248301
+DATA/DB1_LA/03C5F0DDF35CA2B6E0632D0F1EAC8B6B/DATAFILE/undotbs1.274.
1146248309
+DATA/DB1_LA/03C5F0DDF35CA2B6E0632D0F1EAC8B6B/DATAFILE/users.275.114
6248315
+DATA/DB1_LA/03C5F1C9B142A2F1E0632D0F1EACF21A/DATAFILE/system.276.11
46248317
+DATA/DB1_LA/03C5F1C9B142A2F1E0632D0F1EACF21A/DATAFILE/sysaux.277.11
46248323
+DATA/DB1_LA/03C5F1C9B142A2F1E0632D0F1EACF21A/DATAFILE/undotbs1.278.
1146248331
+DATA/DB1_LA/03C5F1C9B142A2F1E0632D0F1EACF21A/DATAFILE/users.279.114
6248337

```

19 rows selected.

```
SQL> select name from v$controlfile;
```

NAME

```

-----
-----
+DATA/DB1_LA/CONTROLFILE/current.260.1146248209
+LOGS/DB1_LA/CONTROLFILE/current.257.1146248209

```

```
SQL> select name from v$tempfile;
```

NAME

```
-----  
-----  
+DATA/DB1_LA/TEMPFILE/temp.287.1146248371  
+DATA/DB1_LA/03C5C01A66EE9797E0632D0F1EAC5F59/TEMPFILE/temp.288.1146  
248375  
+DATA/DB1_LA/03C5EFD07C41A1FAE0632D0F1EAC9BD8/TEMPFILE/temp.290.1146  
248463  
+DATA/DB1_LA/03C5F0DDF35CA2B6E0632D0F1EAC8B6B/TEMPFILE/temp.291.1146  
248463  
+DATA/DB1_LA/03C5F1C9B142A2F1E0632D0F1EACF21A/TEMPFILE/temp.292.1146  
248463
```

```
SQL> select group#, type, member from v$logfile order by 2, 1;
```

GROUP#	TYPE	MEMBER
1	ONLINE	+LOGS/DB1_LA/ONLINELOG/group_1.259.1146248349
1	ONLINE	+DATA/DB1_LA/ONLINELOG/group_1.280.1146248347
2	ONLINE	+DATA/DB1_LA/ONLINELOG/group_2.281.1146248351
2	ONLINE	+LOGS/DB1_LA/ONLINELOG/group_2.258.1146248353
3	ONLINE	+DATA/DB1_LA/ONLINELOG/group_3.282.1146248355
3	ONLINE	+LOGS/DB1_LA/ONLINELOG/group_3.260.1146248355
4	STANDBY	+DATA/DB1_LA/ONLINELOG/group_4.283.1146248357
4	STANDBY	+LOGS/DB1_LA/ONLINELOG/group_4.261.1146248359
5	STANDBY	+DATA/DB1_LA/ONLINELOG/group_5.284.1146248361
5	STANDBY	+LOGS/DB1_LA/ONLINELOG/group_5.262.1146248363
6	STANDBY	+LOGS/DB1_LA/ONLINELOG/group_6.263.1146248365
6	STANDBY	+DATA/DB1_LA/ONLINELOG/group_6.285.1146248365
7	STANDBY	+LOGS/DB1_LA/ONLINELOG/group_7.264.1146248369
7	STANDBY	+DATA/DB1_LA/ONLINELOG/group_7.286.1146248367

14 rows selected.

```
SQL> select name, open_mode from v$database;
```

NAME	OPEN_MODE
DB1	READ ONLY

9. スタンバイデータベースを再起動します `mount` ステージングし、次のコマンドを実行して、スタンバイ データベースの管理リカバリをアクティブ化します。

```
alter database recover managed standby database disconnect from  
session;
```

```
SQL> shutdown immediate;  
Database closed.  
Database dismounted.  
ORACLE instance shut down.  
SQL> startup mount;  
ORACLE instance started.
```

```
Total System Global Area 8053062944 bytes  
Fixed Size                  9182496 bytes  
Variable Size               1291845632 bytes  
Database Buffers            6744440832 bytes  
Redo Buffers                 7593984 bytes
```

```
Database mounted.
```

```
SQL> alter database recover managed standby database disconnect from  
session;
```

```
Database altered.
```

10. スタンバイ データベースのリカバリ ステータスを検証します。注意してください `recovery logmerger` で `APPLYING\_LOG` アクション。

```
SQL> SELECT ROLE, THREAD#, SEQUENCE#, ACTION FROM
V$DATAGUARD_PROCESS;
```

ROLE	THREAD#	SEQUENCE#	ACTION
recovery apply slave	0	0	IDLE
recovery apply slave	0	0	IDLE
recovery apply slave	0	0	IDLE
recovery apply slave	0	0	IDLE
recovery logmerger	1	30	APPLYING_LOG
RFS ping	1	30	IDLE
RFS async	1	30	IDLE
archive redo	0	0	IDLE
archive redo	0	0	IDLE
archive redo	0	0	IDLE
gap manager	0	0	IDLE

ROLE	THREAD#	SEQUENCE#	ACTION
managed recovery	0	0	IDLE
redo transport monitor	0	0	IDLE
log writer	0	0	IDLE
archive local	0	0	IDLE
redo transport timer	0	0	IDLE

16 rows selected.

```
SQL>
```

これにより、管理されたスタンバイ リカバリが有効になった状態で、プライマリからスタンバイへの db1 の Data Guard 保護のセットアップが完了します。

Data Guard Brokerのセットアップ

Oracle Data Guard Broker は、Oracle Data Guard 構成の作成、保守、監視を自動化および一元化する分散管理フレームワークです。次のセクションでは、Data Guard 環境を管理するために Data Guard Broker を設定する方法を説明します。

1. sqlplus 経由で次のコマンドを使用して、プライマリ データベースとスタンバイ データベースの両方でデータ ガード ブローカーを起動します。

```
alter system set dg_broker_start=true scope=both;
```

2. プライマリ データベースから、SYSDBA として Data Guard Broker に接続します。

```
[oracle@ip-172-30-15-45 db1]$ dgmgrl sys@db1_NY
DGMGRL for Linux: Release 19.0.0.0.0 - Production on Wed Aug 30
19:34:14 2023
Version 19.18.0.0.0

Copyright (c) 1982, 2019, Oracle and/or its affiliates. All rights
reserved.

Welcome to DGMGRL, type "help" for information.
Password:
Connected to "db1_NY"
Connected as SYSDBA.
```

3. Data Guard Broker 構成を作成して有効にします。

```
DGMGRL> create configuration dg_config as primary database is db1_NY
connect identifier is db1_NY;
Configuration "dg_config" created with primary database "db1_ny"
DGMGRL> add database db1_LA as connect identifier is db1_LA;
Database "db1_la" added
DGMGRL> enable configuration;
Enabled.
DGMGRL> show configuration;

Configuration - dg_config

Protection Mode: MaxPerformance
Members:
db1_ny - Primary database
db1_la - Physical standby database

Fast-Start Failover: Disabled

Configuration Status:
SUCCESS (status updated 28 seconds ago)
```

4. Data Guard Broker 管理フレームワーク内でデータベースのステータスを検証します。

```
DGMGRL> show database db1_ny;
```

Database - db1_ny

```
Role:                PRIMARY
Intended State:       TRANSPORT-ON
Instance(s):         db1
```

```
Database Status:
SUCCESS
```

```
DGMGRL> show database db1_la;
```

Database - db1_la

```
Role:                PHYSICAL STANDBY
Intended State:       APPLY-ON
Transport Lag:        0 seconds (computed 1 second ago)
Apply Lag:            0 seconds (computed 1 second ago)
Average Apply Rate:   2.00 KByte/s
Real Time Query:      OFF
Instance(s):         db1
```

```
Database Status:
SUCCESS
```

```
DGMGRL>
```

障害が発生した場合、Data Guard Broker を使用してプライマリ データベースをスタンバイに瞬時にフェイルオーバーできます。

他のユースケース用にスタンバイ データベースをクローンする

Data Guard の AWS FSx ONTAP にスタンバイデータベースをステージングする主な利点は、最小限の追加ストレージ投資で、FlexCloned を使用して他の多くのユースケースに対応できることです。次のセクションでは、NetApp SnapCenter ツールを使用して、DEV、TEST、REPORT などの他の目的で、FSx ONTAP にマウントされリカバリ中のスタンバイ データベース ボリュームのスナップショットを作成し、クローンを作成する方法を説明します。

以下は、SnapCenter を使用して Data Guard の管理対象フィジカル スタンバイ データベースから READ/WRITE データベースをクローンするための高レベルの手順です。SnapCenter のセットアップと構成方法の詳細については、以下を参照してください。["SnapCenter を使用したハイブリッド クラウド データベース ソリューション"](#) 関連する Oracle セクション。

1. まず、テスト テーブルを作成し、プライマリ データベースのテスト テーブルに行を挿入します。次に、トランザクションがスタンバイまで通過し、最終的にクローンまで到達するかどうかを検証します。

```
[oracle@ip-172-30-15-45 db1]$ sqlplus / as sysdba

SQL*Plus: Release 19.0.0.0.0 - Production on Thu Aug 31 16:35:53
2023
Version 19.18.0.0.0

Copyright (c) 1982, 2022, Oracle. All rights reserved.

Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 -
Production
Version 19.18.0.0.0

SQL> alter session set container=db1_pdb1;

Session altered.

SQL> create table test(
  2   id integer,
  3   dt timestamp,
  4   event varchar(100));

Table created.

SQL> insert into test values(1, sysdate, 'a test transaction on
primary database db1 and ec2 db host: ip-172-30-15-
45.ec2.internal');

1 row created.

SQL> commit;
```

Commit complete.

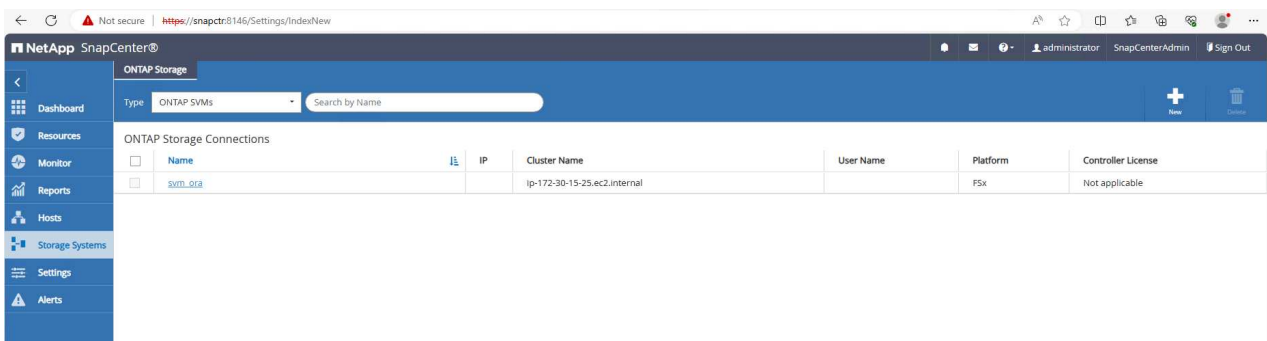
```
SQL> select * from test;
```

```
          ID
-----
DT
-----
EVENT
-----
          1
31-AUG-23 04.49.29.000000 PM
a test transaction on primary database db1 and ec2 db host: ip-172-
30-15-45.ec2.
internal
```

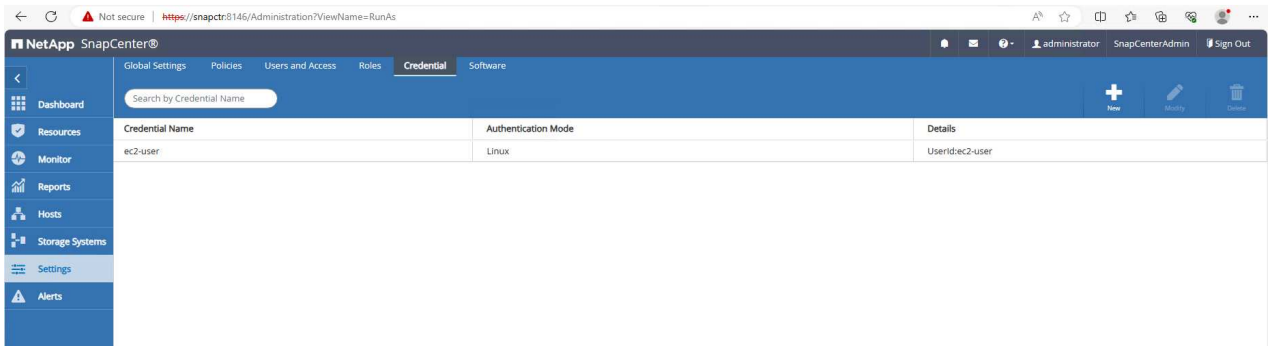
```
SQL> select instance_name, host_name from v$instance;
```

```
INSTANCE_NAME
-----
HOST_NAME
-----
db1
ip-172-30-15-45.ec2.internal
```

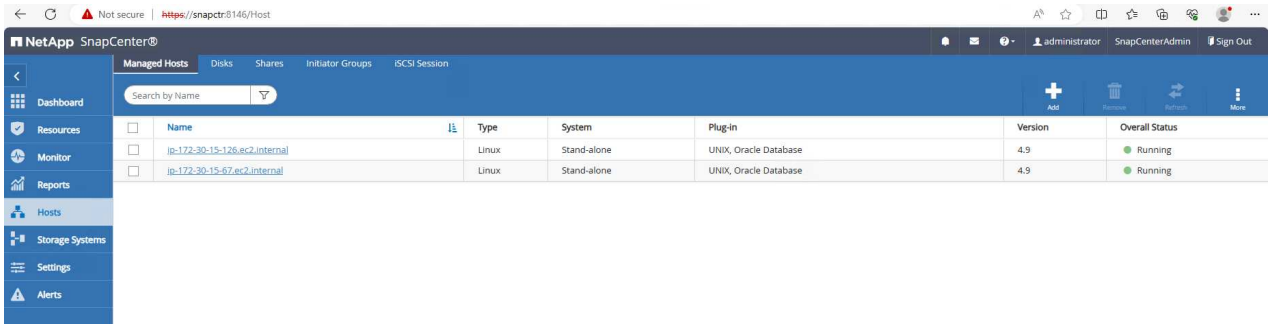
2. FSxストレージクラスターを追加する `Storage Systems` FSx クラスター管理 IP と fsxadmin 資格情報を使用してSnapCenterにアクセスします。



3. AWS ec2-user を追加する Credential`で `Settings。

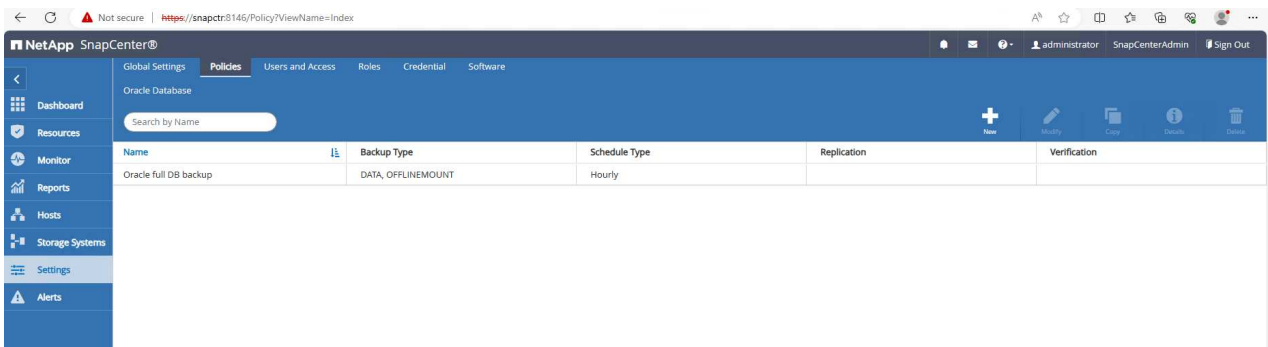


4. スタンバイEC2 DBインスタンスを追加し、EC2 DBインスタンスのクローンを作成して Hosts。

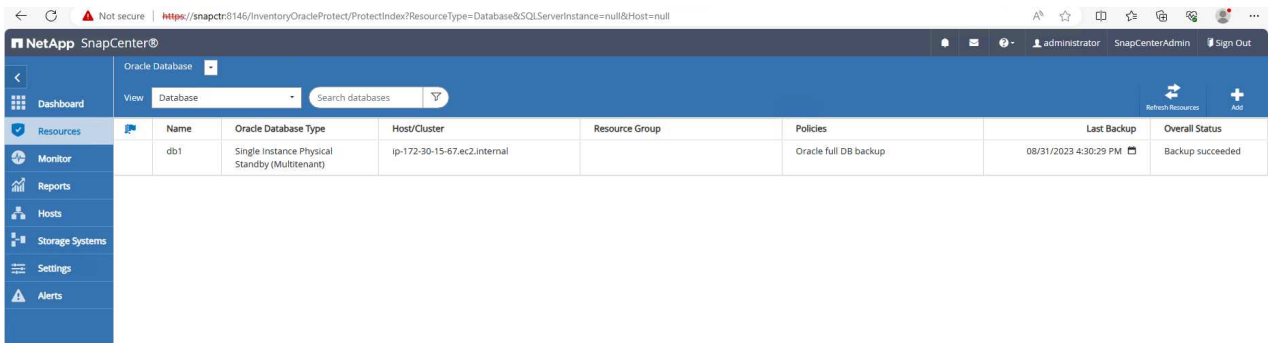


クローン EC2 DB インスタンスには、同様の Oracle ソフトウェア スタックがインストールおよび設定されている必要があります。私たちのテストケースでは、グリッドインフラストラクチャと Oracle 19C はインストールおよび構成されましたが、データベースは作成されていません。

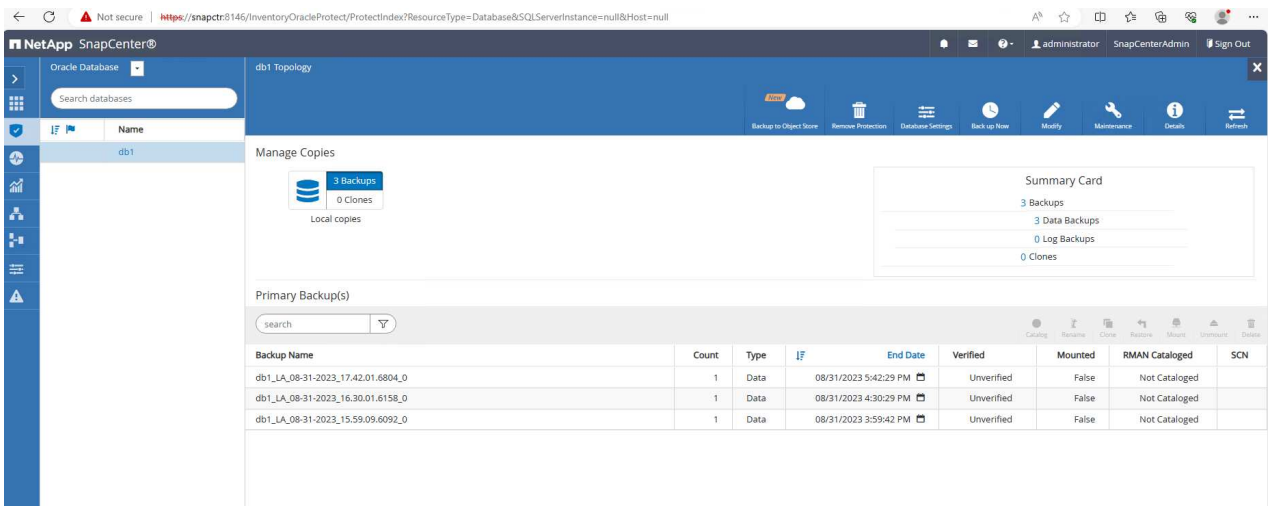
5. オフライン/マウントの完全データベース バックアップに合わせてカスタマイズされたバックアップポリシーを作成します。



6. スタンバイデータベースを保護するためにバックアップポリシーを適用する `Resources` タブ。



7. データベース名をクリックすると、データベースのバックアップ ページが開きます。データベース クローンに使用するバックアップを選択し、クリックします。`Clone` クローンワークフローを起動するボタン。



8. 選択 `Complete Database Clone` クローンインスタンスの SID に名前を付けます。

Clone from db1

1 Name

2 Locations

3 Credentials

4 PreOps

5 PostOps

6 Notification

7 Summary

☒ Complete Database Clone

Clone SID

db1dev

Exclude PDBs

Type to find PDBs

☐ PDB Clone

Previous

Next

9. スタンバイ DB からクローンされたデータベースをホストするクローン ホストを選択します。データ ファイル、制御ファイル、および REDO ログについてはデフォルトを受け入れます。スタンバイデータベースのディスク グループに対応する 2 つの ASM ディスク グループがクローン ホスト上に作成されます。

Clone from db1

1 Name

2 Locations

3 Credentials

4 PreOps

5 PostOps

6 Notification

7 Summary

Select the host to create a clone

Clone host

ip-172-30-15-126.ec2.internal

Datafile locations

+SC_2090922_db1dev

+SC_2342319_db1dev

Reset

Control files

+SC_2090922_db1dev/db1dev/control/control01.ctl

+SC_2090922_db1dev/db1dev/control/control02.ctl

Reset

Redo logs

Group		Size	Unit	Number of files	
RedoGroup 1	×	200	MB	2	+
RedoGroup 2	×	200	MB	2	+
RedoGroup 3	×	200	MB	2	+

Reset

Previous

Next

10. OS ベースの認証にはデータベース資格情報は必要ありません。Oracle ホームの設定を、クローン EC2 データベース インスタンスで構成されているものと一致させます。

Clone from db1

1 Name

2 Locations

3 Credentials

4 PreOps

5 PostOps

6 Notification

7 Summary

Database Credentials for the clone

Credential name for sys user

None

+

i

ASM instance Credential name

None

+

i

Database port

1521

ASM Port

1521

Oracle Home Settings

i

Oracle Home

/u01/app/oracle/product/19.0.0/dev

Oracle OS User

oracle

Oracle OS Group

oinstall

Previous

Next

11. 必要に応じてクローン データベース パラメータを変更し、クローン前に実行するスクリプトがあれば指定します。

Clone from db1

1 Name

2 Locations

3 Credentials

4 PreOps

5 PostOps

6 Notification

7 Summary

Specify scripts to run before clone operation

Prescript full path

/var/opt/snapcenter/spl/scripts/

Enter Prescript path

Arguments

Script timeout

60

secs

Database Parameter settings

audit_file_dest	/u01/app/oracle/admin/db1dev_LA/adump	X
audit_trail	DB	X
open_cursors	300	X
pga_aggregate_target	2684354560	X

+

Reset

Previous

Next

12. クローン後に実行する SQL を入力します。デモでは、dev/test/report データベースのデータベースアーカイブ モードをオフにするコマンドを実行しました。

Clone from db1

1 Name

2 Locations

3 Credentials

4 PreOps

5 PostOps

6 Notification

7 Summary

Until Cancel recovery will be performed for Physical Standby Dataguard/Active Dataguard database.

☒ Create new DBID

☒ Create tempfile for temporary tablespace

☐ Enter SQL queries to apply when clone is created

shutdown immediate ; startup mount ; alter database noarchivelog ; alter database open ;

+

Reset

☐ Enter scripts to run after clone operation

Previous

Next

13. 必要に応じて電子メール通知を設定します。

Clone from db1

1 Name

2 Locations

3 Credentials

4 PreOps

5 PostOps

6 Notification

7 Summary

Provide email settings ⓘ

Email preference

Never

From

From email

To

Email to

Subject

Notification

☐ Attach job report

Previous

Next

14. 概要を確認し、クリック `Finish` クローンを開始します。

Clone from db1

1 Name

2 Locations

3 Credentials

4 PreOps

5 PostOps

6 Notification

7 Summary

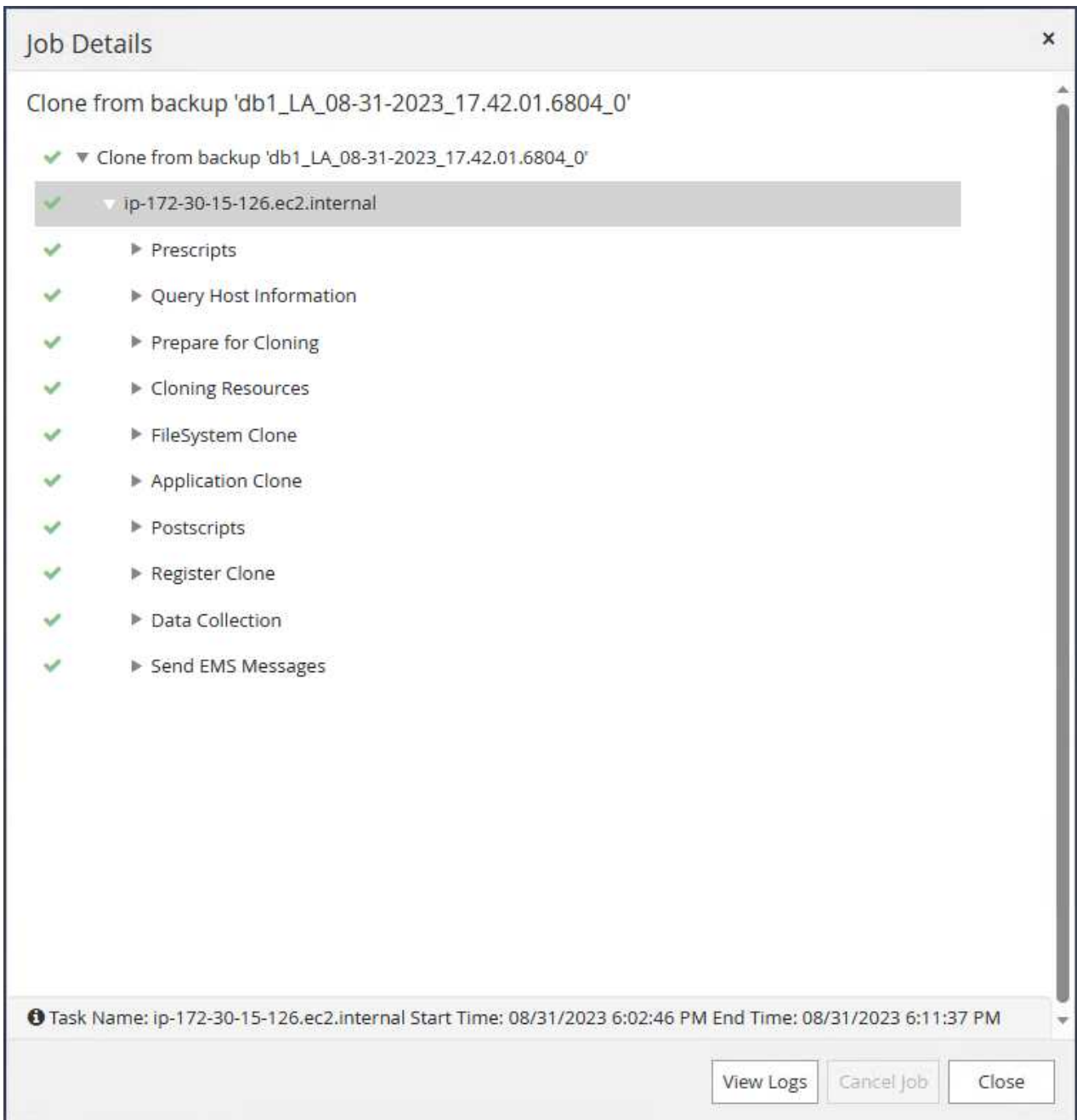
Summary

Clone from backup	db1_LA_08-31-2023_17.42.01.6804_0
Clone SID	db1dev
Clone server	ip-172-30-15-126.ec2.internal
Exclude PDBs	none
Oracle home	/u01/app/oracle/product/19.0.0/dev
Oracle OS user	oracle
Oracle OS group	oinstall
Datafile mountpaths	+SC_2090922_db1dev +SC_2342319_db1dev
Control files	+SC_2090922_db1dev/db1dev/control/control01.ctl +SC_2090922_db1dev/db1dev/control/control02.ctl
Redo groups	RedoGroup =1 TotalSize =200 Path =+SC_2090922_db1dev/db1dev/redolog/redo01_01.log RedoGroup =1 TotalSize =200 Path =+SC_2090922_db1dev/db1dev/redolog/redo01_02.log RedoGroup =2 TotalSize =200 Path =+SC_2090922_db1dev/db1dev/redolog/redo02_01.log RedoGroup =2 TotalSize =200 Path =+SC_2090922_db1dev/db1dev/redolog/redo02_02.log RedoGroup =3 TotalSize =200 Path =+SC_2090922_db1dev/db1dev/redolog/redo03_01.log RedoGroup =3 TotalSize =200 Path =+SC_2090922_db1dev/db1dev/redolog/redo03_02.log RedoGroup =4 TotalSize =200 Path =+SC_2090922_db1dev/db1dev/redolog/redo04_01.log RedoGroup =4 TotalSize =200 Path =+SC_2090922_db1dev/db1dev/redolog/redo04_02.log RedoGroup =5 TotalSize =200 Path =+SC_2090922_db1dev/db1dev/redolog/redo05_01.log RedoGroup =5 TotalSize =200 Path =+SC_2090922_db1dev/db1dev/redolog/redo05_02.log RedoGroup =6 TotalSize =200 Path =+SC_2090922_db1dev/db1dev/redolog/redo06_01.log RedoGroup =6 TotalSize =200 Path =+SC_2090922_db1dev/db1dev/redolog/redo06_02.log

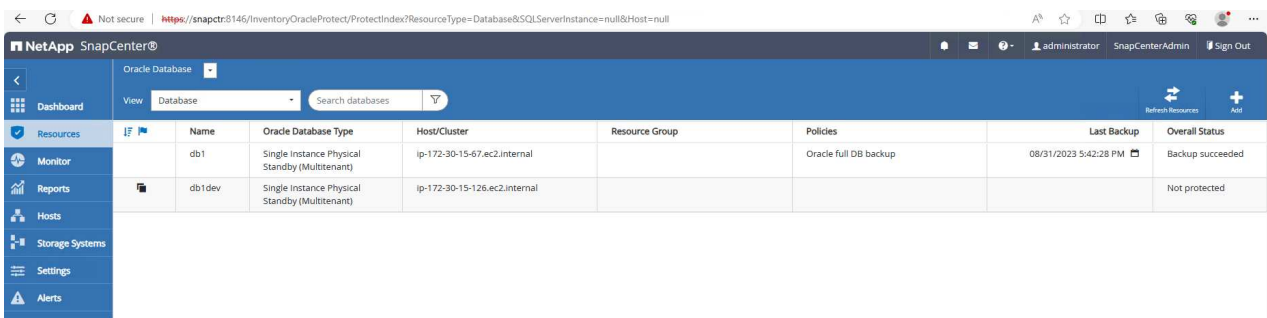
Previous

Finish

15. クローンジョブを監視する `Monitor` タブ。データベース ボリューム サイズが約 300 GB のデータベースのクローン作成には約 8 分かかりました。



16. SnapCenterからクローンデータベースを検証します。これはすぐに登録されます。`Resources` クローン操作直後のタブ。



17. クローン EC2 インスタンスからクローン データベースをクエリします。プライマリ データベースで発生したテスト トランザクションがクローン データベースまで渡されたことを検証しました。

```
[oracle@ip-172-30-15-126 ~]$ export
ORACLE_HOME=/u01/app/oracle/product/19.0.0/dev
[oracle@ip-172-30-15-126 ~]$ export ORACLE_SID=db1dev
[oracle@ip-172-30-15-126 ~]$ export PATH=$PATH:$ORACLE_HOME/bin
[oracle@ip-172-30-15-126 ~]$ sqlplus / as sysdba

SQL*Plus: Release 19.0.0.0.0 - Production on Wed Sep 6 16:41:41 2023
Version 19.18.0.0.0

Copyright (c) 1982, 2022, Oracle. All rights reserved.
```

```
Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 -
Production
Version 19.18.0.0.0
```

```
SQL> select name, open_mode, log_mode from v$database;
```

NAME	OPEN_MODE	LOG_MODE
DB1DEV	READ WRITE	NOARCHIVELOG

```
SQL> select instance_name, host_name from v$instance;
```

INSTANCE_NAME	HOST_NAME
db1dev	ip-172-30-15-126.ec2.internal

```
SQL> alter session set container=db1_pdb1;
```

```
Session altered.
```

```
SQL> select * from test;
```

ID	DT	EVENT

```
1
31-AUG-23 04.49.29.000000 PM
a test transaction on primary database db1 and ec2 db host: ip-172-
30-15-45.ec2.
internal

SQL>
```

これにより、DEV、TEST、REPORT、またはその他のユースケース向けの FSx ストレージ上の Data Guard のスタンバイ データベースからの新しい Oracle データベースのクローン作成と検証が完了します。Data Guard では、同じスタンバイ データベースから複数の Oracle データベースのクローンを作成できます。

詳細情報の入手方法

このドキュメントに記載されている情報の詳細については、次のドキュメントや Web サイトを参照してください。

- Data Guard の概念と管理

["https://docs.oracle.com/en/database/oracle/oracle-database/19/sbydb/index.html#Oracle%C2%AE-Data-Guard"](https://docs.oracle.com/en/database/oracle/oracle-database/19/sbydb/index.html#Oracle%C2%AE-Data-Guard)

- WP-7357: EC2 および FSx における Oracle データベースのデプロイメントのベストプラクティス

["はじめに"](#)

- Amazon FSx ONTAP

["https://aws.amazon.com/fsx/netapp-ontap/"](https://aws.amazon.com/fsx/netapp-ontap/)

- Amazon EC2

https://aws.amazon.com/pm/ec2/?trk=36c6da98-7b20-48fa-8225-4784bced9843&sc_channel=ps&s_kwcid=AL!4422!3!467723097970!e!!g!!aws%20ec2&ef_id=Cj0KCQiA54KfBhCKARIsAJzSrdqwQrghn6I71jiWzSeaT9Uh1-vY-VfhJixF-xnv5rWwn2S7RqZOTQ0aAh7eEALw_wcB:G:s&s_kwcid=AL!4422!3!467723097970!e!!g!!aws%20ec2

著作権に関する情報

Copyright © 2025 NetApp, Inc. All Rights Reserved. Printed in the U.S. このドキュメントは著作権によって保護されています。著作権所有者の書面による事前承諾がある場合を除き、画像媒体、電子媒体、および写真複写、記録媒体、テープ媒体、電子検索システムへの組み込みを含む機械媒体など、いかなる形式および方法による複製も禁止します。

ネットアップの著作物から派生したソフトウェアは、次に示す使用許諾条項および免責条項の対象となります。

このソフトウェアは、ネットアップによって「現状のまま」提供されています。ネットアップは明示的な保証、または商品性および特定目的に対する適合性の暗示的保証を含み、かつこれに限定されないいかなる暗示的な保証も行いません。ネットアップは、代替品または代替サービスの調達、使用不能、データ損失、利益損失、業務中断を含み、かつこれに限定されない、このソフトウェアの使用により生じたすべての直接的損害、間接的損害、偶発的損害、特別損害、懲罰的損害、必然的損害の発生に対して、損失の発生の可能性が通知されていたとしても、その発生理由、根拠とする責任論、契約の有無、厳格責任、不法行為（過失またはそうでない場合を含む）にかかわらず、一切の責任を負いません。

ネットアップは、ここに記載されているすべての製品に対する変更を随時、予告なく行う権利を保有します。ネットアップによる明示的な書面による合意がある場合を除き、ここに記載されている製品の使用により生じる責任および義務に対して、ネットアップは責任を負いません。この製品の使用または購入は、ネットアップの特許権、商標権、または他の知的所有権に基づくライセンスの供与とはみなされません。

このマニュアルに記載されている製品は、1つ以上の米国特許、その他の国の特許、および出願中の特許によって保護されている場合があります。

権利の制限について：政府による使用、複製、開示は、DFARS 252.227-7013（2014年2月）およびFAR 5252.227-19（2007年12月）のRights in Technical Data -Noncommercial Items（技術データ - 非商用品目に関する諸権利）条項の(b)(3)項、に規定された制限が適用されます。

本書に含まれるデータは商用製品および / または商用サービス（FAR 2.101の定義に基づく）に関係し、データの所有権はNetApp, Inc.にあります。本契約に基づき提供されるすべてのネットアップの技術データおよびコンピュータ ソフトウェアは、商用目的であり、私費のみで開発されたものです。米国政府は本データに対し、非独占的かつ移転およびサブライセンス不可で、全世界を対象とする取り消し不能の制限付き使用权を有し、本データの提供の根拠となった米国政府契約に関連し、当該契約の裏付けとする場合にのみ本データを使用できます。前述の場合を除き、NetApp, Inc.の書面による許可を事前に得ることなく、本データを使用、開示、転載、改変するほか、上演または展示することはできません。国防総省にかかる米国政府のデータ使用权については、DFARS 252.227-7015(b)項（2014年2月）で定められた権利のみが認められます。

商標に関する情報

NetApp、NetAppのロゴ、<http://www.netapp.com/TM>に記載されているマークは、NetApp, Inc.の商標です。その他の会社名と製品名は、それを所有する各社の商標である場合があります。