



TR-5002: Azure NetApp Filesによる Oracle Active Data Guard のコスト削減

NetApp database solutions

NetApp
August 18, 2025

目次

TR-5002: Azure NetApp Files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	7
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	16
Data Guard Brokerのセットアップ	26
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	29
詳細情報の入手方法	43

TR-5002: Azure NetApp Filesによる Oracle Active Data Guard のコスト削減

このソリューションでは、Microsoft Azure NetApp Files (ANF) をプライマリおよびスタンバイ 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、Azure クラウドでの 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、Azure クラウドに 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 Azure NetApp Files 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 ケースに対応できます。最終的な結果として、Active Data Guard ライセンスを排除しながら、Active Data Guard の機能を効果的に提供できます。このドキュメントでは、ANF ストレージ上の既存のプライマリ データベースとフィジカル スタンバイ データベースを使用して Oracle Data Guard を設定する方法を説明します。スタンバイ データベースは、NetApp SnapCenter データベース管理ツールを使用して、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バックアップおよびクローン作成されます。NetApp ソリューション エンジニアリング チームは、ユーザーの介入を必要とせず、データベース クローンのライフサイクル管理を完全に自動化するために、ユーザーが定義したスケジュールでクローンを更新する自動化ツールキットも提供しています。

このソリューションは、次のユースケースに対応します。

- Azure リージョン全体の Microsoft Azure NetApp Files ストレージ上のプライマリ データベースと物理スタンバイ データベース間の Oracle Data Guard の実装。
- 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、フィジカル スタンバイ データベースをバックアップしてクローンを作成します。
- 自動化による Oracle データベース クローン更新ライフサイクル管理。

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、および災害復旧のために Azure クラウドに Oracle Active Data Guard を設定する DBA。

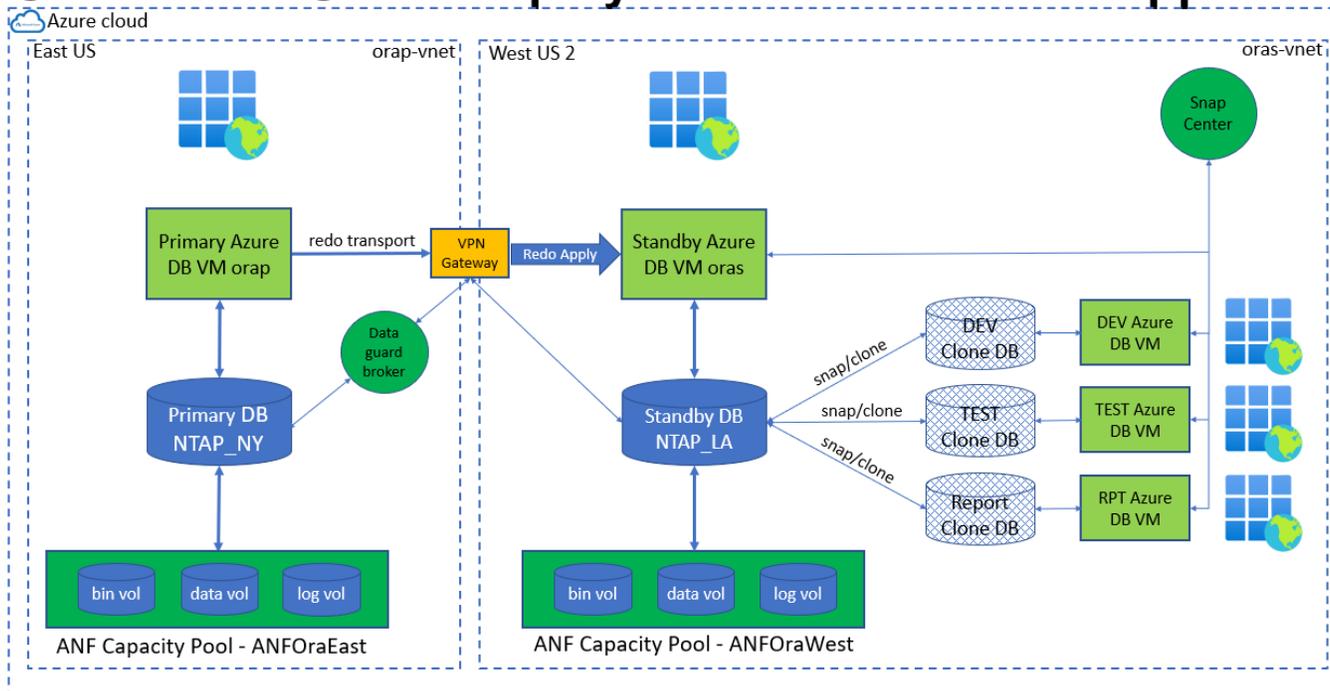
- Azure クラウドでの Oracle Active Data Guard 構成に関心のあるデータベース ソリューション アーキテクト。
- Oracle Data Guard をサポートする Azure NetApp Files ストレージを管理するストレージ管理者。
- Azure クラウド環境で Oracle Data Guard を立ち上げたいアプリケーション所有者。

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、実際のユーザー展開環境と一致しない可能性のある Azure クラウド ラボ設定で実行されました。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 Azure NetApp Files



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		
Azure NetApp Files	Microsoftが提供する現在のバージョン	2つの3 TiB容量プール、標準サービスレベル、自動QoS
DB サーバー用の Azure VM	標準 B4ms (4 vCPU、16 GiB メモリ)	3 つの DB VM (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	Red Hat Enterprise Linux 8.6 (LVM) - x64 Gen2	テスト用にRedHatサブスクリプションを導入

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
SnapCenter	バージョン6.0.1	ビルド 6.0.1.4487
NFS	バージョン3.0	Oracle で dNFS が有効

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	NTAP_NY	NTAP_NY.internal.cloudapp.net
Standby	NTAP_LA	NTAP_LA.internal.cloudapp.net

導入検討の重要な要素

- *スタンバイ データベース クローン。*プライマリ データベースからトランザクション ログを受け取って適用しながら、フィジカル スタンバイ データベースを複製して DB VM にマウントし、DEV、TEST、レポートなどの他のワークロードをサポートできます。クローンには薄いクローンと厚いクローンがあります。現時点では、ANF はスタンバイ データベースの完全なコピーであるシック クローンのみをサポートしています。ANF シン クローン オプションはまもなくリリースされる予定です。データベース ボリュームのシン クローン コピーでは、スタンバイ データベースの同じ DB ボリュームを共有し、コピー オン ライト テクノロジーを利用して書き込み IO を処理します。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 ANF 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。
- *Azure VM のサイズ設定*これらのテストと検証では、4 つの vCPU と 16 GiB のメモリを備えた Azure VM - Standard_B4ms を使用しました。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 Azure DB VM のサイズを適切に設定する必要があります。

- * Azure NetApp Files の構成。* Azure NetApp FilesはAzure NetAppストレージアカウントに次のように割り当てられます。Capacity Pools。これらのテストと検証では、East リージョンで Oracle プライマリをホストし、West2 リージョンでスタンバイ データベースをホストするために 3 TiB の容量プールを展開しました。ANF 容量プールには、Standard、Premium、Ultra の 3 つのサービス レベルがあります。ANF 容量プールの IO 容量は、容量プールのサイズとサービス レベルによって決まります。本番環境での導入では、NetAppOracle データベースのスループット要件を完全に評価し、それに応じてデータベース容量プールのサイズを設定することを推奨しています。容量プールの作成時に、QoS を自動または手動に設定し、保存時のデータの暗号化をシングルまたはダブルに設定できます。
- **dNFS** 構成。dNFS を使用すると、ANF ストレージを備えた Azure 仮想マシン上で実行される Oracle データベースは、ネイティブ NFS クライアントよりも大幅に多くの I/O を実行できます。NetApp自動化ツールキットを使用した自動化された Oracle デプロイメントでは、NFSv3 上に dNFS が自動的に構成されます。

ソリューションの展開

Oracle Data Guard をセットアップするための開始点として、プライマリ Oracle データベースが既に VNet 内の Azure クラウド環境にデプロイされているものと想定されます。理想的には、プライマリ データベースは、NFS マウントを使用して ANF ストレージにデプロイされます。Oracle データベース ストレージ用に 3 つの NFS マウント ポイントが作成されます。Oracle バイナリ ファイル用のマウント /u01、Oracle データ ファイルと制御ファイル用のマウント /u02、Oracle の現在のログ ファイルとアーカイブ ログ ファイル、および冗長制御ファイル用のマウント /u03 です。

プライマリ Oracle データベースは、NetApp ONTAPストレージ、または Azure エコシステム内またはプライベート データ センター内の任意の他のストレージで実行することもできます。次のセクションでは、ANF ストレージを備えた Azure のプライマリ Oracle DB と ANF ストレージを備えた Azure の物理スタンバイ Oracle DB の間に Oracle Data Guard を設定するための段階的な展開手順について説明します。

展開の前提条件

展開には次の前提条件が必要です。

1. Azure クラウド アカウントが設定され、必要な VNet とネットワーク サブネットが Azure アカウント内に作成されています。
2. Azure クラウド ポータル コンソールから、少なくとも 3 台の Azure Linux VM をデプロイする必要があります。1 台はプライマリ Oracle DB サーバーとして、1 台はスタンバイ Oracle DB サーバーとして、もう 1 台はレポート、開発、テストなどのためのクローン ターゲット DB サーバーとしてデプロイします。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。また、Microsoft ["Azure 仮想マシン"](#) 詳細についてはこちらをご覧ください。
3. プライマリ Oracle データベースは、プライマリ Oracle DB サーバーにインストールされ、構成されている必要があります。一方、スタンバイ Oracle DB サーバーまたはクローン Oracle DB サーバーには、Oracle ソフトウェアのみがインストールされ、Oracle データベースは作成されません。理想的には、Oracle ファイル ディレクトリのレイアウトは、すべての Oracle DB サーバー上で完全に一致する必要があります。Azure クラウドおよび ANF での Oracle の自動展開に関する NetApp の推奨事項の詳細については、次の技術レポートを参照してください。

◦ ["TR-4987: NFS を使用した Azure NetApp Files への簡素化された自動 Oracle デプロイメント"](#)



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領域を確保するために、Azure VM のルート ボリュームに少なくとも 128 G が割り当てられていることを確認します。

4. Azure クラウド ポータル コンソールから、Oracle データベース ボリュームをホストするための 2 つの ANF ストレージ容量プールをデプロイします。真の DataGuard 構成を模倣するには、ANF ストレージ容量プールを異なるリージョンに配置する必要があります。ANF ストレージの展開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。"[クイックスタート: Azure NetApp Files をセットアップして NFS ボリュームを作成する](#)" 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。

Name	Type	Resource group	Location	Subscription
ANFOraEast	NetApp account	ANFAVSRG	East US	Hybrid Cloud TME Onprem
ANFOraWest	NetApp account	ANFAVSRG	West US 2	Hybrid Cloud TME Onprem

5. プライマリ Oracle データベースとスタンバイ Oracle データベースが 2 つの異なるリージョンに配置されている場合は、2 つの別々の VNet 間のデータ トラフィック フローを許可するように VPN ゲートウェイを構成する必要があります。Azure での詳細なネットワーク構成については、このドキュメントの範囲外です。次のスクリーンショットは、VPN ゲートウェイがどのように構成され、接続され、ラボでデータ トラフィック フローがどのように確認されるかについての参考資料を提供します。

ラボ VPN ゲートウェイ

:

Microsoft Azure

Virtual network gateways

Name	Virtual network	Gateway type	Resource group	Location	Subscription
orap-vnet-gw	orap-vnet	Vpn	ANFAVSRG	East US	Hybrid Cloud TME Onprem
oras-vnet-gw	oras-vnet	Vpn	ANFAVSRG	West US 2	Hybrid Cloud TME Onprem
vNetgw	EHCvNet	Vpn	NSOL	Central US	Hybrid Cloud TME Onprem

プライマリ VNET ゲートウェイ

Microsoft Azure

Virtual network gateways > orap-vnet-gw

Overview

Essentials

- Resource group: ANFAVSRG
- Location: East US
- Subscription: Hybrid Cloud TME Onprem
- Subscription ID: Defa2dfb-917c-4497-b56a-b3f4eadb8111
- Tags: database: oracle, product_line: Field use - various
- SKU: VpnGw2AZ
- Gateway type: VPN
- VPN type: Route-based
- Virtual network: orap-vnet
- Public IP address: 27.152.7.193 (orap-vnet-gw-nic1)

Health check: Perform a quick health check to detect possible gateway issues. [Go to Resource health](#)

Advisor Recommendations: Check Critical, Warning, and Informational Recommendations. [Go to Advisor](#)

Advanced troubleshooting: Run a troubleshooting tool to investigate failure causes and perform repair actions. [Go to VPN Troubleshooting](#)

Documentation: View guidance on helpful topics related to VPN gateway. [View documentation](#)

Total tunnel ingress

Total tunnel egress

Vnet ゲートウェイの接続ステータス

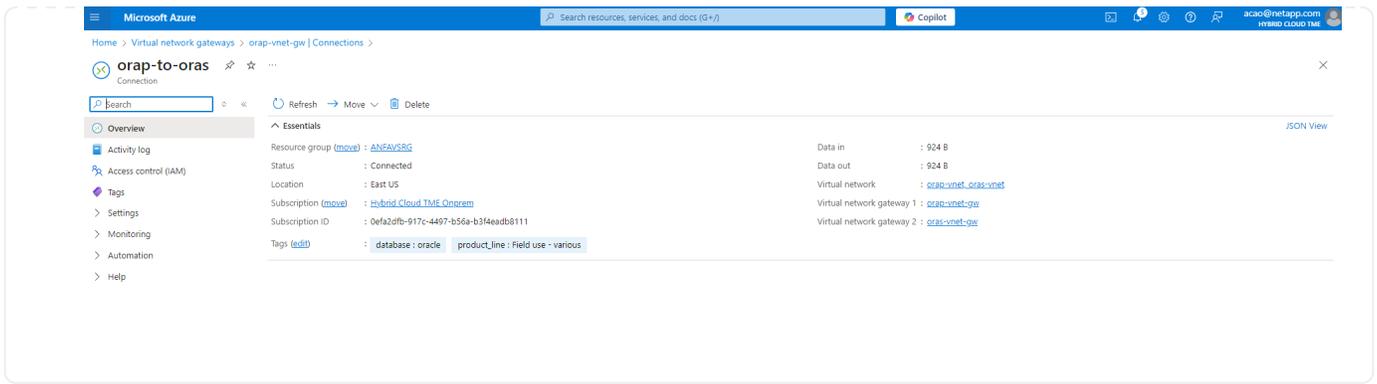
Microsoft Azure

Virtual network gateways > orap-vnet-gw

Virtual network gateway | Connections

Name	Status	Connection type	Peer
orap-to-oras	Connected	VNet-to-VNet	oras-vnet-gw
oras-to-orap	Connected	VNet-to-VNet	orap-vnet-gw

トラフィック フローが確立されていることを確認します (3つのドットをクリックしてページを開きます)



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 Azure DB サーバー上に NTAP と呼ばれるプライマリ Oracle データベースを 3 つの NFS マウント ポイント (Oracle バイナリ用の /u01、Oracle データ ファイルと Oracle 制御ファイル用の /u02、Oracle アクティブ ログ、アーカイブ ログ ファイル、および冗長 Oracle 制御ファイル用の /u03) でセットアップしました。以下は、Oracle Data Guard 保護用のプライマリ データベースを設定するための詳細な手順を示しています。すべての手順は、Oracle データベース所有者またはデフォルトのユーザーとして実行する必要があります。`oracle` ユーザー。

1. プライマリ Azure DB サーバー orap.internal.cloudapp.net 上のプライマリ データベース NTAP は、最初は ANF をデータベース ストレージとして使用するスタンドアロン データベースとしてデプロイされます。

```
orap.internal.cloudapp.net:
resource group: ANFAVSRG
Location: East US
size: Standard B4ms (4 vcpus, 16 GiB memory)
OS: Linux (redhat 8.6)
pub_ip: 172.190.207.231
pri_ip: 10.0.0.4

[oracle@orap ~]$ df -h
Filesystem                Size      Used Avail Use% Mounted on
devtmpfs                  7.7G      4.0K   7.7G   1% /dev
tmpfs                     7.8G         0   7.8G   0% /dev/shm
tmpfs                     7.8G    209M   7.5G   3% /run
tmpfs                     7.8G         0   7.8G   0% /sys/fs/cgroup
/dev/mapper/rootvg-rootlv 22G     413M   22G    2% /
/dev/mapper/rootvg-usrlv  10G     2.1G   8.0G   21% /usr
/dev/sda1                 496M    181M   315M   37% /boot
/dev/mapper/rootvg-homelv 2.0G     47M   2.0G    3% /home
/dev/sda15                495M     5.8M   489M    2% /boot/efi
/dev/mapper/rootvg-varlv  8.0G    1.1G   7.0G   13% /var
/dev/mapper/rootvg-tmplv  12G    120M   12G    1% /tmp
/dev/sdb1                 32G     49M   30G    1% /mnt
10.0.2.36:/orap-u02       500G    7.7G  493G    2% /u02
10.0.2.36:/orap-u03       450G    6.1G  444G    2% /u03
10.0.2.36:/orap-u01       100G    9.9G   91G   10% /u01

[oracle@orap ~]$ 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.
#
#
NTAP:/u01/app/oracle/product/19.0.0/NTAP:N
```

2. Oracle ユーザーとしてプライマリ DB サーバーにログインします。sqlplus 経由でデータベースにログインし、プライマリで強制ログを有効にします。

```
alter database force logging;
```

```
[oracle@orap admin]$ sqlplus / as sysdba

SQL*Plus: Release 19.0.0.0.0 - Production on Tue Nov 26 20:12:02
2024
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 database force logging;

Database altered.
```

3. sqlplus から、プライマリ DB 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;
```

```
SQL> alter database flashback on;
```

```
Database altered.
```

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。プライマリで pwd ファイルが設定されていない場合は orapwd ユーティリティを使用して 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。フェイルオーバーが発生し、REDO データの受信を開始すると、プライマリ データベースはすぐにスタンバイ ロールに移行できます。次のコマンドを 4 回繰り返して、4 つのスタンバイ ログ ファイルを作成します。

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

```

SQL> alter database add standby logfile thread 1 size 200M;

Database altered.

SQL> /

Database altered.

SQL> /

Database altered.

SQL> /

Database altered.

SQL> set lin 200
SQL> col member for a80
SQL> select group#, type, member from v$logfile;

   GROUP# TYPE      MEMBER
-----
-----
-----
          3 ONLINE  /u03/orareco/NTAP/onlinelog/redo03.log
          2 ONLINE  /u03/orareco/NTAP/onlinelog/redo02.log
          1 ONLINE  /u03/orareco/NTAP/onlinelog/redo01.log
          4 STANDBY
/u03/orareco/NTAP/onlinelog/o1_mf_4__2m115vkv_.log
          5 STANDBY
/u03/orareco/NTAP/onlinelog/o1_mf_5__2m3c5cyd_.log
          6 STANDBY
/u03/orareco/NTAP/onlinelog/o1_mf_6__2m4d7dhh_.log
          7 STANDBY
/u03/orareco/NTAP/onlinelog/o1_mf_7__2m5ct7g1_.log

```

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

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

7. pfile を修正し、次のパラメータを追加します。

```
vi /home/oracle/initNTAP.ora
```

Update the following parameters if not set:

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

8. sqlplus から、修正された pfile から spfile を再作成し、\$ORACLE_HOME/dbs ディレクトリ内の既存の spfile を上書きします。

```
create spfile='$ORACLE_HOME/dbs/spfileNTAP.ora' from
pfile='/home/oracle/initNTAP.ora';
```

9. \$ORACLE_HOME/network/admin ディレクトリの Oracle tnsnames.ora を変更して、名前解決用の db_unique_name を追加します。

```
vi $ORACLE_HOME/network/admin/tnsnames.ora
```

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

NTAP_NY =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP) (HOST =
orap.internal.cloudapp.net) (PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SID = NTAP)
    )
  )

NTAP_LA =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP) (HOST =
oras.internal.cloudapp.net) (PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SID = NTAP)
    )
  )

LISTENER_NTAP =
  (ADDRESS = (PROTOCOL = TCP) (HOST =
orap.internal.cloudapp.net) (PORT = 1521))
```



Azure DB サーバーにデフォルトとは異なる名前を付ける場合は、ホスト名解決のためにローカル ホスト ファイルに名前を追加します。

10. プライマリ データベースのデータ ガード サービス名 NTAP_NY_DGMGRL.internal.cloudapp.net を listener.ora ファイルに追加します。

```
vi $ORACLE_HOME/network/admin/listener.ora
```

```

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

LISTENER.NTAP =
  (DESCRIPTION_LIST =
    (DESCRIPTION =
      (ADDRESS = (PROTOCOL = TCP) (HOST =
orap.internal.cloudapp.net) (PORT = 1521))
      (ADDRESS = (PROTOCOL = IPC) (KEY = EXTPROC1521))
    )
  )

SID_LIST_LISTENER.NTAP =
  (SID_LIST =
    (SID_DESC =
      (GLOBAL_DBNAME = NTAP_NY_DGMGRL.internal.cloudapp.net)
      (ORACLE_HOME = /u01/app/oracle/product/19.0.0/NTAP)
      (SID_NAME = NTAP)
    )
  )
)

```

11. sqlplus 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。

```
shutdown immediate;
```

```
startup;
```

```
SQL> show parameter name
```

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

cdb_cluster_name	string	
cell_offloadgroup_name	string	
db_file_name_convert	string	
db_name	string	NTAP
db_unique_name	string	NTAP_NY
global_names	boolean	FALSE
instance_name	string	NTAP
lock_name_space	string	

```

log_file_name_convert      string
pdb_file_name_convert     string
processor_group_name       string

```

```

NAME                        TYPE                VALUE
-----

```

```

service_names              string
NTAP_NY.internal.cloudapp.net

```

```
SQL> sho parameter log_archive_dest
```

```

NAME                        TYPE                VALUE
-----

```

```

log_archive_dest           string
log_archive_dest_1        string
LOCATION=USE_DB_RECOVERY_FILE_

```

```
DEST
```

```
VALID_FOR=(ALL_LOGFILES,A
```

```
LL_ROLES)
```

```
DB_UNIQUE_NAME=NTAP_
```

```
NY
```

```

log_archive_dest_10       string
log_archive_dest_11       string
log_archive_dest_12       string
log_archive_dest_13       string
log_archive_dest_14       string
log_archive_dest_15       string

```

```

NAME                        TYPE                VALUE
-----

```

```

log_archive_dest_16       string
log_archive_dest_17       string
log_archive_dest_18       string
log_archive_dest_19       string
log_archive_dest_2        string
ASYNC VALID_FO

```

```
SERVICE=NTAP_LA
```

```
R=(ONLINE_LOGFILES,PRIMARY_ROL
```

```
E)
```

```
DB_UNIQUE_NAME=NTAP_LA
```

```

log_archive_dest_20       string
log_archive_dest_21       string

```

```
.
```

```
.
```

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 DB サーバーと一致させるために、スタンバイ DB サーバー上の 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 DB サーバーのデータベース ストレージ構成 (データベース ディレクトリのレイアウトや NFS マウント ポイントのサイズなど) は、プライマリ DB サーバーと一致させることが理想的です。スタンバイ Oracle DB サーバーを設定し、HA/DR 保護のために Oracle DataGuard をアクティブ化する詳細な手順を次に示します。すべてのコマンドはデフォルトの Oracle 所有者ユーザー ID として実行する必要があります `oracle`。

1. まず、プライマリ Oracle DB サーバー上のプライマリ データベースの構成を確認します。このデモでは、ANF ストレージ上の 3 つの NFS マウントを使用して、プライマリ DB サーバーに NTAP と呼ばれるプライマリ Oracle データベースをセットアップしました。
2. NetApp のドキュメント TR-4987 に従って Oracle スタンバイ DB サーバーをセットアップする場合 "[TR-4987: NFS を使用した Azure NetApp Files への簡素化された自動 Oracle デプロイメント](#)" タグを使用する `-t software_only_install` ステップ 2 の `Playbook execution` 自動 Oracle インストールを実行します。修正されたコマンド構文を以下に示します。このタグにより、Oracle ソフトウェア スタックのインストールと構成は可能になりますが、データベースの作成はできません。

```
ansible-playbook -i hosts 4-oracle_config.yml -u azureuser -e
@vars/vars.yml -t software_only_install
```

3. デモ ラボのスタンバイ サイトのスタンバイ Oracle DB サーバー構成。

```
oras.internal.cloudapp.net:
resource group: ANFAVSRG
Location: West US 2
size: Standard B4ms (4 vcpus, 16 GiB memory)
OS: Linux (redhat 8.6)
pub_ip: 172.179.119.75
pri_ip: 10.0.1.4
```

```
[oracle@oras ~]$ df -h
Filesystem                Size      Used Avail Use% Mounted on
devtmpfs                  7.7G         0  7.7G   0% /dev
tmpfs                     7.8G         0  7.8G   0% /dev/shm
tmpfs                     7.8G    265M  7.5G   4% /run
tmpfs                     7.8G         0  7.8G   0% /sys/fs/cgroup
/dev/mapper/rootvg-rootlv 22G    413M   22G   2% /
/dev/mapper/rootvg-usrlv  10G    2.1G   8.0G  21% /usr
/dev/sda1                 496M    181M  315M  37% /boot
/dev/mapper/rootvg-varlv  8.0G    985M   7.1G  13% /var
/dev/mapper/rootvg-homelv 2.0G     52M   2.0G   3% /home
/dev/mapper/rootvg-tmplv  12G    120M   12G   1% /tmp
/dev/sda15                495M    5.8M  489M   2% /boot/efi
/dev/sdb1                 32G     49M   30G   1% /mnt
10.0.3.36:/oras-u01       100G    9.5G   91G  10% /u01
10.0.3.36:/oras-u02       500G    8.1G  492G   2% /u02
10.0.3.36:/oras-u03       450G    4.8G  446G   2% /u03
```

4. Oracle ソフトウェアをインストールして構成したら、Oracle ホームとパスを設定します。また、スタンバイ \$ORACLE_HOME dbs ディレクトリから、プライマリ データベースから Oracle パスワードをコピーします (まだコピーしていない場合)。

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

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

```
scp oracle@10.0.0.4:$ORACLE_HOME/dbs/orapwNTAP .
```

5. 次のエントリで tnsnames.ora ファイルを更新します。

```
vi $ORACLE_HOME/network/admin/tnsnames.ora
```

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

```
NTAP_NY =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP) (HOST =
orap.internal.cloudapp.net) (PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SID = NTAP)
    )
  )
```

```
NTAP_LA =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP) (HOST =
oras.internal.cloudapp.net) (PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SID = NTAP)
    )
  )
```

6. 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。

```
vi $ORACLE_HOME/network/admin/listener.ora
```

```

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

LISTENER.NTAP =
  (DESCRIPTION_LIST =
    (DESCRIPTION =
      (ADDRESS = (PROTOCOL = TCP) (HOST =
oras.internal.cloudapp.net) (PORT = 1521))
      (ADDRESS = (PROTOCOL = IPC) (KEY = EXTPROC1521))
    )
  )

SID_LIST_LISTENER =
  (SID_LIST =
    (SID_DESC =
      (SID_NAME = NTAP)
    )
  )

SID_LIST_LISTENER.NTAP =
  (SID_LIST =
    (SID_DESC =
      (GLOBAL_DBNAME = NTAP_LA_DGMGRL.internal.cloudapp.net)
      (ORACLE_HOME = /u01/app/oracle/product/19.0.0/NTAP)
      (SID_NAME = NTAP)
    )
  )

LISTENER =
  (ADDRESS_LIST =
    (ADDRESS = (PROTOCOL = TCP) (HOST =
oras.internal.cloudapp.net) (PORT = 1521))
  )

```

7. dbca を起動して、プライマリ データベース NTAP からスタンバイ データベースをインスタンス化します。

```

dbca -silent -createDuplicateDB -gdbName NTAP
-primaryDBConnectionString
oras.internal.cloudapp.net:1521/NTAP_NY.internal.cloudapp.net -sid
NTAP -initParams fal_server=NTAP_NY -createAsStandby -dbUniqueName
NTAP_LA

```

```
[oracle@oras admin]$ dbca -silent -createDuplicateDB -gdbName NTAP
-primaryDBConnectionString
orap.internal.cloudapp.net:1521/NTAP_NY.internal.cloudapp.net -sid
NTAP -initParams fal_server=NTAP_NY -createAsStandby -dbUniqueName
NTAP_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/NTAP_LA/NTAP_LA.log" for further
details.
```

- 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@oras admin]$ 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 $ORACLE_SID are not allowed.
#
#
NTAP:/u01/app/oracle/product/19.0.0/NTAP:N
[oracle@oras admin]$ export ORACLE_SID=NTAP
[oracle@oras admin]$ sqlplus / as sysdba

```

```

SQL*Plus: Release 19.0.0.0.0 - Production on Tue Nov 26 23:04:07
2024
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
NTAP	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	NTAP
db_unique_name	string	NTAP_LA
global_names	boolean	FALSE
instance_name	string	NTAP
lock_name_space	string	
log_file_name_convert	string	
pdb_file_name_convert	string	
processor_group_name	string	

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

```
-----  
-----  
service_names                                string  
NTAP_LA.internal.cloudapp.net  
SQL> show parameter log_archive_config
```

```
NAME                                           TYPE      VALUE  
-----  
-----  
log_archive_config                            string  
DG_CONFIG=(NTAP_NY,NTAP_LA)  
SQL> show parameter fal_server
```

```
NAME                                           TYPE      VALUE  
-----  
-----  
fal_server                                    string    NTAP_NY  
SQL> select name from v$datafile;
```

```
NAME  
-----
```

```
-----  
/u02/oradata/NTAP/system01.dbf  
/u02/oradata/NTAP/sysaux01.dbf  
/u02/oradata/NTAP/undotbs01.dbf  
/u02/oradata/NTAP/pdbseed/system01.dbf  
/u02/oradata/NTAP/pdbseed/sysaux01.dbf  
/u02/oradata/NTAP/users01.dbf  
/u02/oradata/NTAP/pdbseed/undotbs01.dbf  
/u02/oradata/NTAP/NTAP_pdb1/system01.dbf  
/u02/oradata/NTAP/NTAP_pdb1/sysaux01.dbf  
/u02/oradata/NTAP/NTAP_pdb1/undotbs01.dbf  
/u02/oradata/NTAP/NTAP_pdb1/users01.dbf
```

```
NAME  
-----
```

```
-----  
/u02/oradata/NTAP/NTAP_pdb2/system01.dbf  
/u02/oradata/NTAP/NTAP_pdb2/sysaux01.dbf  
/u02/oradata/NTAP/NTAP_pdb2/undotbs01.dbf  
/u02/oradata/NTAP/NTAP_pdb2/users01.dbf  
/u02/oradata/NTAP/NTAP_pdb3/system01.dbf  
/u02/oradata/NTAP/NTAP_pdb3/sysaux01.dbf  
/u02/oradata/NTAP/NTAP_pdb3/undotbs01.dbf  
/u02/oradata/NTAP/NTAP_pdb3/users01.dbf
```

```
19 rows selected.
```

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

```
NAME
```

```
-----  
-----  
/u02/oradata/NTAP/control01.ctl  
/u03/orareco/NTAP_LA/control02.ctl
```

```
SQL> col member form a80
```

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

```
GROUP# TYPE MEMBER  
-----  
-----  
1 ONLINE  
/u03/orareco/NTAP_LA/onlinelog/o1_mf_1_mndl6mxh_.log  
2 ONLINE  
/u03/orareco/NTAP_LA/onlinelog/o1_mf_2_mndl7jdb_.log  
3 ONLINE  
/u03/orareco/NTAP_LA/onlinelog/o1_mf_3_mndl8f03_.log  
4 STANDBY  
/u03/orareco/NTAP_LA/onlinelog/o1_mf_4_mndl99m7_.log  
5 STANDBY  
/u03/orareco/NTAP_LA/onlinelog/o1_mf_5_mndlb67d_.log  
6 STANDBY  
/u03/orareco/NTAP_LA/onlinelog/o1_mf_6_mndlc2tw_.log  
7 STANDBY  
/u03/orareco/NTAP_LA/onlinelog/o1_mf_7_mndlczhb_.log
```

```
7 rows selected.
```

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 6442449688 bytes
Fixed Size                  9177880 bytes
Variable Size              1090519040 bytes
Database Buffers          5335154688 bytes
Redo Buffers                7598080 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。

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

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

ROLE	THREAD#	SEQUENCE#	ACTION
post role transition	0	0	IDLE
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	18	APPLYING_LOG
managed recovery	0	0	IDLE
RFS async	1	18	IDLE
RFS ping	1	18	IDLE
archive redo	0	0	IDLE
redo transport timer	0	0	IDLE

ROLE	THREAD#	SEQUENCE#	ACTION
gap manager	0	0	IDLE
archive redo	0	0	IDLE
archive redo	0	0	IDLE
redo transport monitor	0	0	IDLE
log writer	0	0	IDLE
archive local	0	0	IDLE

```
17 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ているプライマリからスタンバイへの NTAP 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@orap ~]$ dgmgrl sys@NTAP_NY
DGMGRL for Linux: Release 19.0.0.0.0 - Production on Wed Dec 11
20:53:20 2024
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 "NTAP_NY"
Connected as SYSDBA.
DGMGRL>
```

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
NTAP_NY connect identifier is NTAP_NY;
Configuration "dg_config" created with primary database "ntap_ny"
DGMGRL> add database NTAP_LA as connect identifier is NTAP_LA;
Database "ntap_la" added
DGMGRL> enable configuration;
Enabled.
DGMGRL> show configuration;

Configuration - dg_config

Protection Mode: MaxPerformance
Members:
  ntap_ny - Primary database
  ntap_la - Physical standby database

Fast-Start Failover: Disabled

Configuration Status:
SUCCESS (status updated 3 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。もし `Fast-Start Failover` 有効になっている場合、Data Guard Broker は、障害が検出されると、ユーザーの介入なしにプライマリ データベースをスタンバイ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

Oracle Data Guard セットアップで Oracle スタンバイ データベースを ANF 上にホストする主な利点は、シンクローンが有効になっている場合、最小限の追加ストレージ投資で、他の多くのユースケースに対応するためにすぐにクローンを作成できることです。NetApp、Oracle DataGuard データベースを管理するために SnapCenter UI ツールを使用することを推奨しています。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、ANF 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 Oracle Data Guard 内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 データベースを複製するための高レベルの手順です。ANF 上での SnapCenter for Oracle のセットアップと構成方法の詳細については、TR-4988 を参照してください。["SnapCenter を使用した ANF 上の Oracle データベースのバックアップ、リカバリ、およびクローン"](#) 詳細については。

1. ユースケースの検証は、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することから開始します。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ことを検証します。

```
[oracle@orap ~]$ sqlplus / as sysdba

SQL*Plus: Release 19.0.0.0.0 - Production on Wed Dec 11 16:33:17
2024
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=ntap_pdb1;

Session altered.

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

Table created.

SQL> insert into test values(1, sysdate, 'a test transaction at
primary database NTAP on DB server orap.internal.cloudapp.net');

1 row created.

SQL> commit;
```

```
Commit complete.
```

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

```
          ID
```

```
-----
```

```
DT
```

```
-----
```

```
-----
```

```
EVENT
```

```
-----
```

```
-----
```

```
1
```

```
11-DEC-24 04.38.44.000000 PM
```

```
a test transaction at primary database NTAP on DB server
```

```
orap.internal.cloudapp.
```

```
net
```

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

```
INSTANCE_NAME
```

```
-----
```

```
HOST_NAME
```

```
-----
```

```
NTAP
```

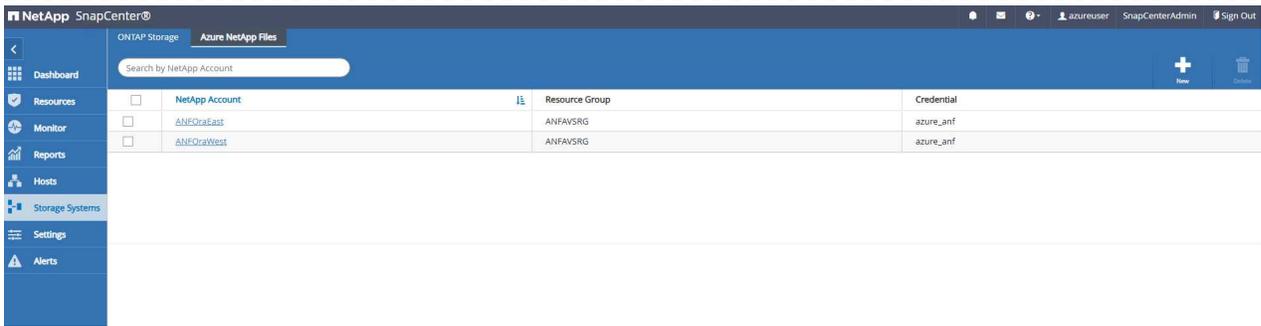
```
orap
```

```
SQL>
```

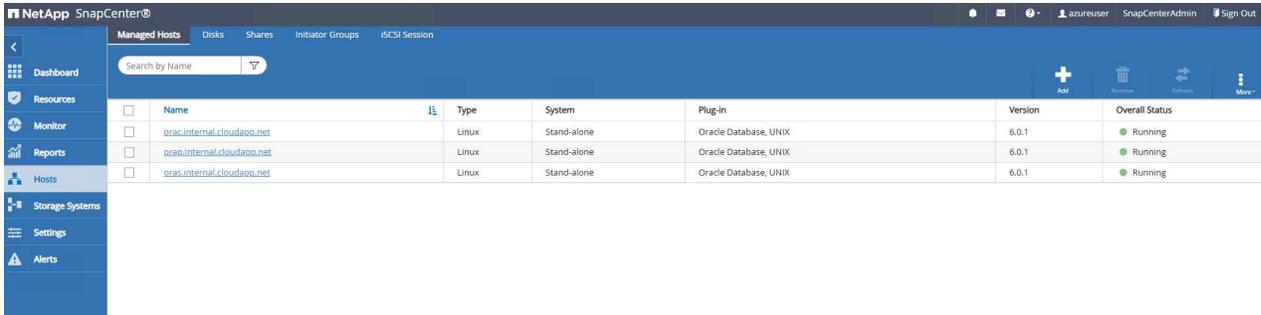
2. SnapCenter構成では、Unixユーザー（デモの場合はazureuser）とAzure資格情報（デモの場合にはazure_anf）が追加されました。Credential`で`Settings。

Credential Name	Authentication Mode	Details
azure_anf	AzureCredential	
azureuser	Linux	UserId:azureuser

3. azure_anf認証情報を使用してANFストレージを追加します Storage Systems。 Azure サブスクリプションに複数の ANF ストレージ アカウントがある場合は、ドロップダウン リストをクリックして適切なストレージ アカウントを選択してください。このデモ用に 2 つの専用の Oracle ストレージ アカウントを作成しました。

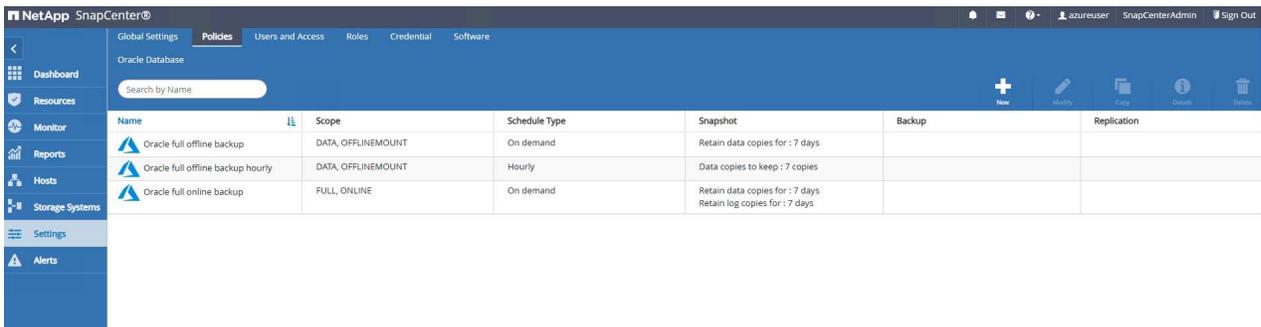


4. すべてのOracle DBサーバーがSnapCenterに追加されました Hosts。

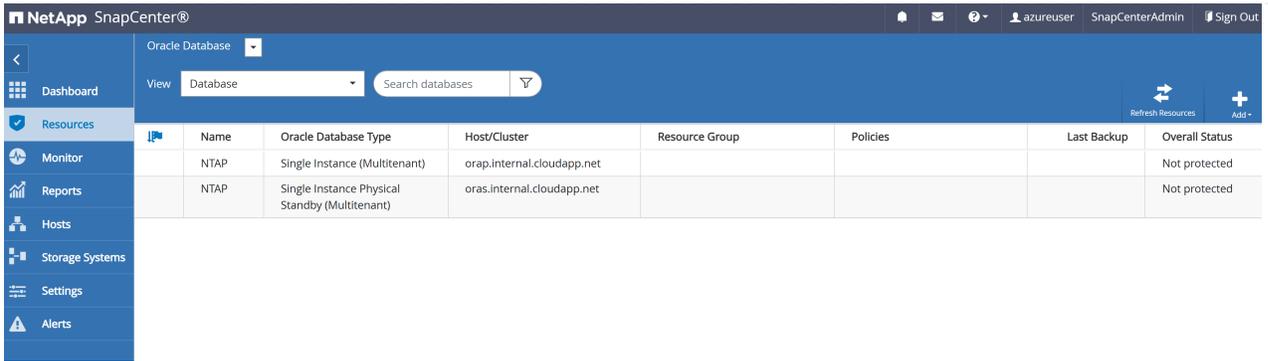


クローン DB サーバーには、同一の Oracle ソフトウェア スタックがインストールおよび構成されている必要があります。私たちのテストケースでは、Oracle 19C ソフトウェアはインストールおよび構成されていますが、データベースは作成されていません。

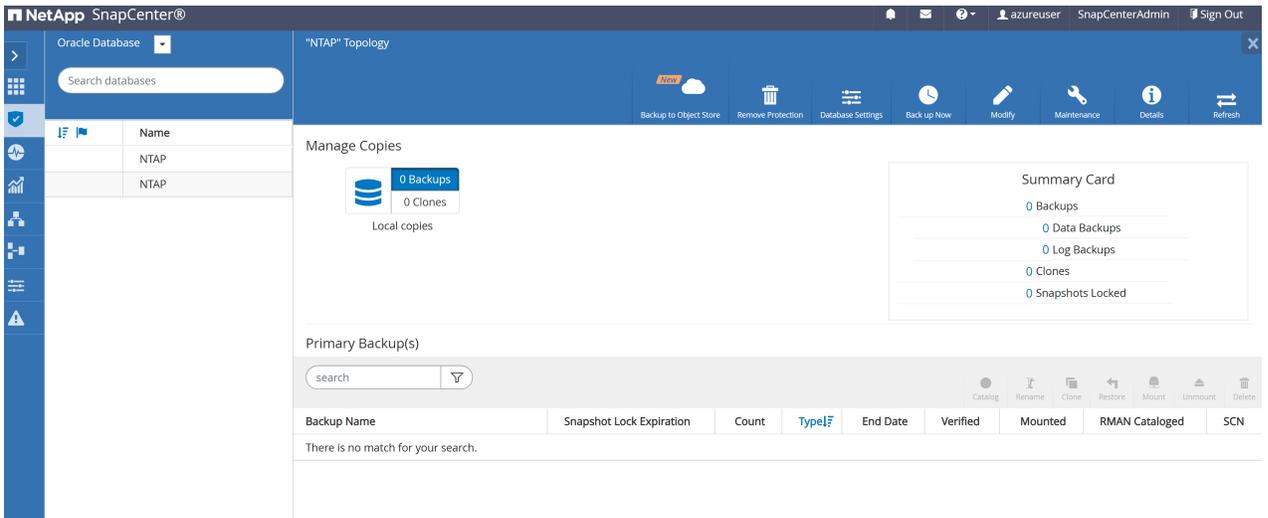
5. オフライン/マウントの完全データベース バックアップに合わせてカスタマイズされたバックアップポリシーを作成します。



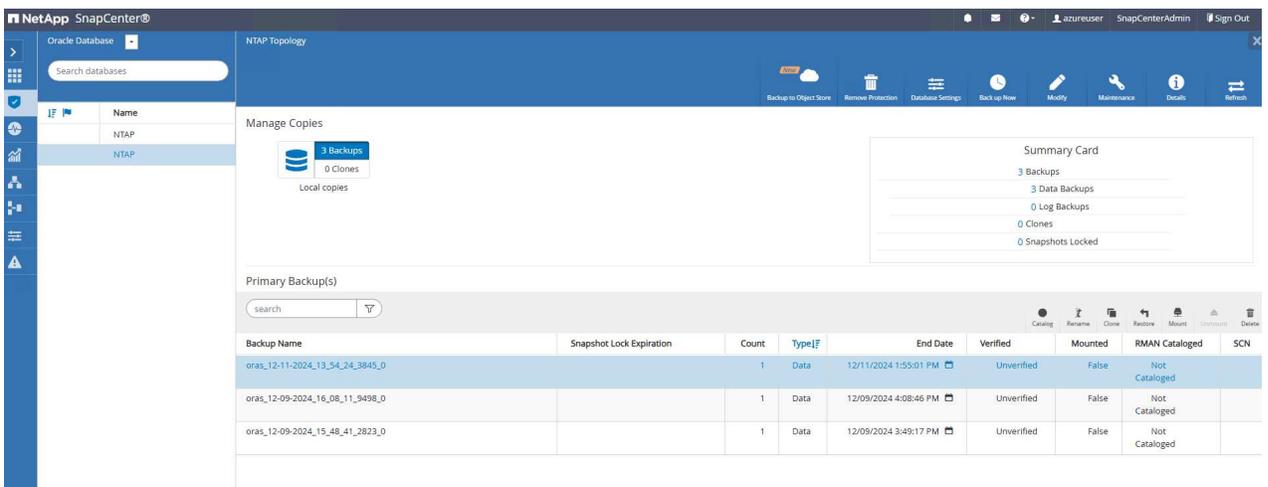
6. スタンバイデータベースを保護するためにバックアップポリシーを適用する Resources `タブ。最初に検出された場合、データベースのステータスは次のように表示されます。 `Not protected`。



7. バックアップを手動でトリガーするか、バックアップポリシーの適用後に設定された時間にスケジュールに従ってバックアップを実行するかを選択できます。



8. バックアップ後、データベース名をクリックしてデータベースバックアップページを開きます。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。



9. 選択してください `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 NTAP



- 1 Name
- 2 Locations
- 3 Credentials
- 4 PreOps
- 5 PostOps
- 6 Notification
- 7 Summary

Capacity Pool Max. Throughput (MiB/s)



Complete Database Clone

Clone SID

Exclude PDBs

PDB Clone

Previous

Next

10. 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 DB サーバーを選択します。データ ファイル、REDO ログについてはデフォルトを受け入れます。制御ファイルを /u03 マウントポイントに置きます。

Clone from NTAP

1 Name

2 Locations

3 Credentials

4 PreOps

5 PostOps

6 Notification

7 Summary

Select the host to create a clone

Clone host

Datafile locations ?

Reset

Control files ?

Redo logs ?

Group	Size	Unit	Number of files	
▶ RedoGroup 1	<input type="button" value="X"/>	200	MB	<u>1</u> <input type="button" value="+"/>
▶ RedoGroup 2	<input type="button" value="X"/>	200	MB	<u>1</u> <input type="button" value="+"/>
▶ RedoGroup 3	<input type="button" value="X"/>	200	MB	<u>1</u> <input type="button" value="+"/>

Previous

Next

- 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 DB サーバーで構成されているものと一致させます。

1 Name

Database Credentials for the clone

2 Locations

Credential name for sys user

None



3 Credentials

Database port

1521

4 PreOps

5 PostOps

Oracle Home Settings i

6 Notification

Oracle Home

/u01/app/oracle/product/19.0.0/NTAP

7 Summary

Oracle OS User

oracle

Oracle OS Group

oinstall

Previous

Next

12. 必要に応じて、クローン DB の PGA または SGA サイズを小さくするなど、クローン データベース パラメータを変更します。クローン前に実行するスクリプトがあれば指定します。

1 Name

2 Locations

3 Credentials

4 PreOps

5 PostOps

6 Notification

7 Summary

Specify scripts to run before clone operation ⓘ

Prescript full path Arguments Script timeout

Database Parameter settings

pga_aggregate_target	500M	✕	▲
processes	320	✕	
remote_login_passwordfile	EXCLUSIVE	✕	
sga_target	2G	✕	▼

+
Reset

Previous

Next

- 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 NTAP

- 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

14. 必要に応じて電子メール通知を設定します。

1 Name

2 Locations

3 Credentials

4 PreOps

5 PostOps

6 Notification

7 Summary

Provide email settings ⓘ

Email preference From To Subject Attach job report

Previous

Next

15. 概要を確認し、クリック `Finish` クローンを開始します。

Clone from NTAP

1 Name	Summary	
2 Locations	Clone from backup	oras_12-11-2024_13_54_24_3845_0
3 Credentials	Clone SID	NTAPDEV
4 PreOps	Capacity Pool Max. Throughput (MiB/s)	none
5 PostOps	Clone server	orac.internal.cloudapp.net
6 Notification	Exclude PDBs	none
7 Summary	Oracle home	/u01/app/oracle/product/19.0.0/NTAP
	Oracle OS user	oracle
	Oracle OS group	oinstall
	Datafile mountpaths	/u02_NTAPDEV /u03_NTAPDEV
	Control files	/u02_NTAPDEV/NTAPDEV/control/control01.ctl /u03_NTAPDEV/NTAPDEV/control/control02.ctl
	Redo groups	RedoGroup =1 TotalSize =200 Path =/u03_NTAPDEV/NTAPDEV/redolog/redo01_01.log RedoGroup =2 TotalSize =200 Path =/u03_NTAPDEV/NTAPDEV/redolog/redo02_01.log RedoGroup =3 TotalSize =200 Path =/u03_NTAPDEV/NTAPDEV/redolog/redo03_01.log RedoGroup =4 TotalSize =200 Path =/u03_NTAPDEV/NTAPDEV/redolog/redo04_01.log RedoGroup =5 TotalSize =200 Path =/u03_NTAPDEV/NTAPDEV/redolog/redo05_01.log RedoGroup =6 TotalSize =200 Path =/u03_NTAPDEV/NTAPDEV/redolog/redo06_01.log RedoGroup =7 TotalSize =200 Path =/u03_NTAPDEV/NTAPDEV/redolog/redo07_01.log
	Recovery scope	Until Cancel
	Prescript full path	none
	Prescript arguments	
	Postscript full path	none

16. 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 950 GB のデータベースのクローン作成には約 14 分かかりました。

Job Details



Clone from backup 'oras_12-11-2024_13_54_24_3845_0'

- ✓ ▾ Clone from backup 'oras_12-11-2024_13_54_24_3845_0'
- ✓ ▾ orac.internal.cloudapp.net
 - ✓ ▶ Prescripts
 - ✓ ▶ Query Host Information
 - ✓ ▶ Prepare for Cloning
 - ✓ ▶ Cloning Resources
 - ✓ ▶ FileSystem Clone
 - ✓ ▶ Application Clone
 - ✓ ▶ Postscripts
 - ✓ ▶ Register Clone
 - ✓ ▶ Data Collection

Task Name: orac.internal.cloudapp.net Start Time: 12/11/2024 2:53:11 PM End Time: 12/11/2024 3:07:33 PM

View Logs

Cancel Job

Close

17. 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。

Name	Oracle Database Type	Host/Cluster	Resource Group	Policies	Last Backup	Overall Status
NTAP	Single Instance (Multitenant)	orac.internal.cloudapp.net		Oracle full online backup	12/06/2024 11:45:35 AM	Backup succeeded
NTAP	Single Instance Physical Standby (Multitenant)	oras.internal.cloudapp.net		Oracle full offline backup	12/11/2024 1:55:01 PM	Backup succeeded
NTAPDB	Single Instance Physical Standby (Multitenant)	oras.internal.cloudapp.net				Not protected

18. クローン DB サーバーからクローン 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@orac ~]$ sqlplus / as sysdba

SQL*Plus: Release 19.0.0.0.0 - Production on Wed Dec 11 20:16:09
2024
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$databases;

NAME          OPEN_MODE          LOG_MODE
-----
NTAPDEV       READ WRITE         NOARCHIVELOG

SQL> select instance_name, host_name from v$instance;

INSTANCE_NAME
-----
HOST_NAME
-----
NTAPDEV
orac

SQL> alter pluggable database all open;

Pluggable database altered.

SQL> alter pluggable database all save state;

Pluggable database altered.

SQL> alter session set container=ntap_pdb1;

Session altered.
```

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

```
          ID
-----
DT
-----
EVENT
-----
          1
11-DEC-24 04.38.44.000000 PM
a test transaction at primary database NTAP on DB server
orap.internal.cloudapp.
net
```

これで、DEV、TEST、REPORT、またはその他のユース ケース向けの Azure ANF ストレージ上の Oracle Data Guard での Oracle スタンバイ データベース クローンのデモンストレーションが完了します。ANF 上の Oracle Data Guard 内の同じスタンバイ データベースから複数の Oracle データベースをクローンできます。

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。

- Azure NetApp Files

["https://azure.microsoft.com/en-us/products/netapp"](https://azure.microsoft.com/en-us/products/netapp)

- TR-4988: SnapCenterを使用した ANF 上の Oracle データベースのバックアップ、リカバリ、およびクローン

["TR-4988: SnapCenterを使用した ANF 上の Oracle データベースのバックアップ、リカバリ、およびクローン"](#)

- TR-4987: NFS を使用した Azure NetApp Files への簡素化された自動 Oracle デプロイメント

["TR-4987: NFS を使用した Azure NetApp Files への簡素化された自動 Oracle デプロイメント"](#)

- Oracle 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)

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。