



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 NetApp database solutions

NetApp
August 18, 2025

目次

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	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
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
EC2インスタンスのカーネル構成	4
データベースボリュームとLUNをEC2インスタンスホストにプロビジョニングしてマッピングする	8
データベースストレージ構成	12
Oracle グリッド インフラストラクチャのインストール	20
Oracleデータベースのインストール	25
自動展開オプション	32
SnapCenterサービスによる Oracle データベースのバックアップ、リストア、クローン作成	33
詳細情報の入手方法	33

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

アレン・カオ、ニヤズ・モハメド、NetApp

このソリューションでは、iSCSI プロトコルを使用した AWS FSx ONTAP ストレージおよび EC2 コンピューティングインスタンスでの Oracle データベースのデプロイメントと保護、およびボリューム マネージャーとして asm を使用してスタンドアロン ReStart で構成された Oracle データベースの概要と詳細を示します。

目的

ASM (Automatic Storage Management) は、多くの Oracle インストールで採用されている一般的な Oracle ストレージ ボリューム マネージャです。これは、Oracle が推奨するストレージ管理ソリューションでもあります。従来のボリューム マネージャーおよびファイル システムに代わる手段を提供します。Oracle バージョン 11g 以降、ASM はデータベースではなくグリッド インフラストラクチャにパッケージ化されています。そのため、RAC を使用せずに Oracle ASM をストレージ管理に利用するには、Oracle Restart と呼ばれるスタンドアロン サーバーに Oracle Grid Infrastructure をインストールする必要があります。そうすることで、Oracle データベースの導入がさらに複雑になります。ただし、名前が示すように、Oracle が再起動モードでデプロイされると、障害が発生した Oracle サービスはグリッド インフラストラクチャによって自動的に再起動されるか、またはユーザーの介入なしにホストの再起動後に再起動されるため、ある程度の高可用性または HA 機能が提供されます。

このドキュメントでは、EC2 コンピューティングインスタンスを備えた Amazon FSx ONTAP ストレージ環境に、iSCSI プロトコルと Oracle ASM を使用して Oracle データベースをデプロイする方法を説明します。また、NetApp BlueXP コンソールを介して NetApp SnapCenter サービスを使用し、AWS パブリック クラウドでのストレージ効率の高いデータベース操作の開発/テストやその他のユースケースのために Oracle データベースをバックアップ、復元、およびクローン化する方法も説明します。

このソリューションは、次のユースケースに対応します。

- Amazon FSx ONTAP ストレージと iSCSI/ASM を使用した EC2 コンピューティングインスタンスでの Oracle データベースのデプロイメント
- iSCSI/ASM を使用したパブリック AWS クラウドでの Oracle ワークロードのテストと検証
- AWS にデプロイされた Oracle データベースの再起動機能のテストと検証

観客

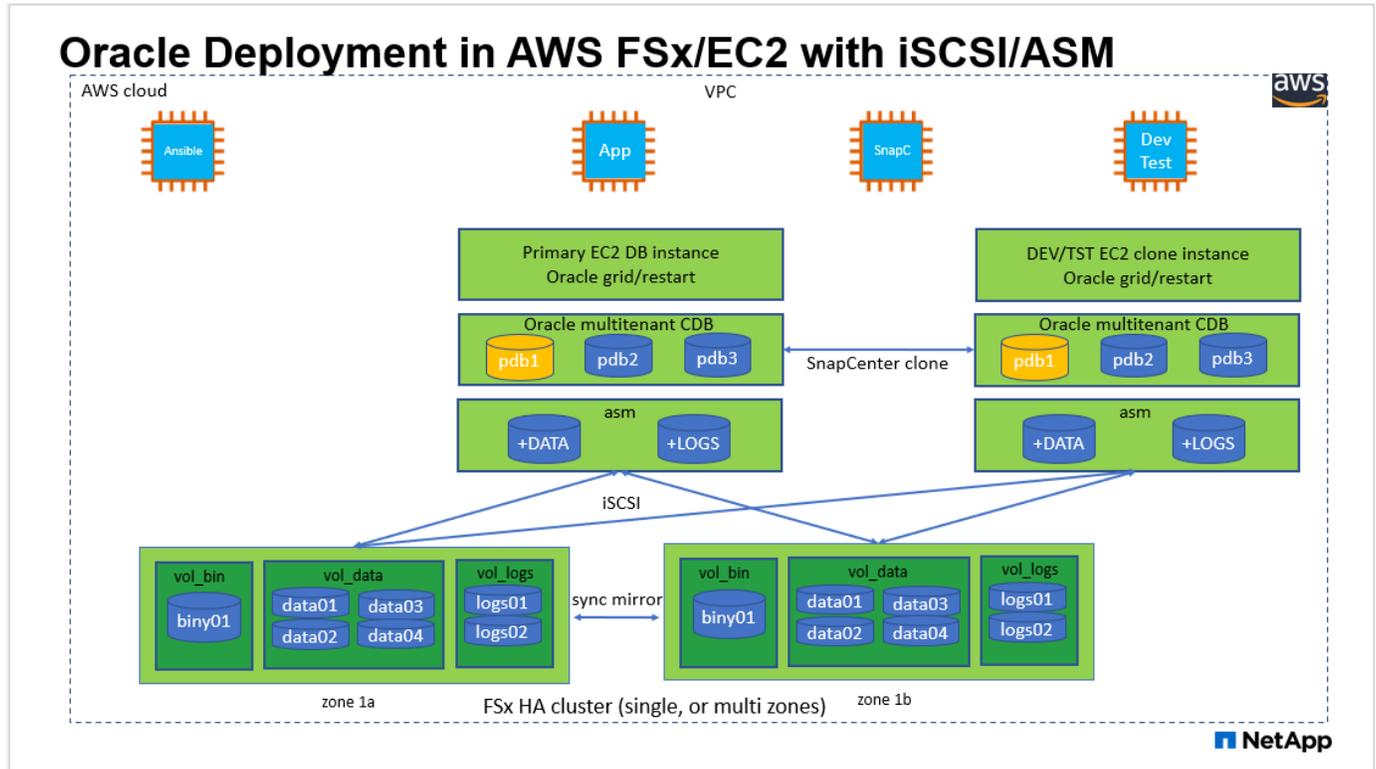
このソリューションは次の人々を対象としています。

- iSCSI/ASM を使用して AWS パブリッククラウドに Oracle を導入したい DBA。
- AWS パブリッククラウドで Oracle ワークロードをテストしたいデータベース ソリューション アーキテクト。
- AWS FSx ストレージにデプロイされた Oracle データベースをデプロイおよび管理したいストレージ管理者。
- AWS FSx/EC2 で Oracle データベース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 および EC2 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



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	2 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、もう 1 つはクローン 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
SnapCenterサービス	version	v2.3.1.2324

導入検討の重要な要素

- *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 インスタンスに AWS EC2 t2.xlarge インスタンス タイプを使用しました。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 クラスターの単一ゾーン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 HA 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。
- *Oracle データとログのレイアウト。*テストと検証では、データ用とログ用にそれぞれ 2 つの ASM ディスク グループを展開しました。+DATA asm ディスク グループ内で、データ ボリュームに 4 つの LUN をプロビジョニングしました。+LOGS asm ディスク グループ内で、ログ ボリュームに 2 つの LUN をプロビジョニングしました。一般的に、Amazon FSx ONTAP ボリューム内に複数の LUN を配置すると、パフォーマンスが向上します。
- **iSCSI** 構成 EC2 インスタンス データベース サーバーは、iSCSI プロトコルを使用して FSx ストレージに接続します。EC2 インスタンスは通常、単一のネットワーク インターフェイスまたは ENI を使用してデプロイされます。単一の NIC インターフェイスは、iSCSI トラフィックとアプリケーショントラフィックの両方を伝送します。アプリケーションと iSCSI トラフィックのスループット要件の両方を満たす適切な EC2 コンピューティングインスタンスを選択するには、Oracle AWR レポートを慎重に分析して、Oracle データベースのピーク I/O スループット要件を評価することが重要です。NetApp、マルチパスが適切に構成された両方の FSx iSCSI エンドポイントに 4 つの iSCSI 接続を割り当てることも推奨しています。
- 作成する各 **Oracle ASM** ディスク グループに使用する **Oracle ASM** 冗長性レベル。FSx はすでに FSx クラスター レベルでストレージをミラーリングしているため、外部冗長性を使用する必要があります。つまり、このオプションでは Oracle ASM がディスク グループの内容をミラーリングすることはできません。
- データベースのバックアップ。NetApp は、NetApp BlueXP コンソール UI を通じて利用できるクラウドでのデータベースのバックアップ、リストア、クローン作成用の SnapCenter software サービスの SaaS バージョンを提供しています。NetApp、高速 (1 分未満) なスナップショット バックアップ、迅速な (数分) データベース リストア、およびデータベースのクローン作成を実現するために、このようなサービスを実装することを推奨しています。

ソリューションの展開

次のセクションでは、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. 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、プライマリ Oracle DB サーバーとして 1 つ、オプションの代替クローンターゲット DB サーバーとして 1 つの EC2 Linux インスタンスをデプロイする必要があります。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、"[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 データベース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。

```
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。

EC2インスタンスのカーネル構成

前提条件がプロビジョニングされたら、ec2-userとしてEC2インスタンスにログインし、rootユーザーにsudoして、Oracleインストール用のLinuxカーネルを構成します。

1. ステージングディレクトリを作成する`/tmp/archive`フォルダを設定し、`777`許可。

```
mkdir /tmp/archive  
  
chmod 777 /tmp/archive
```

2. Oracleバイナリインストールファイルとその他の必要なrpmファイルをダウンロードしてステージングします。`/tmp/archive`ディレクトリ。

記載するインストールファイルのリストについては以下を参照してください。/tmp/archive EC2インスタンス上。

```
[ec2-user@ip-172-30-15-58 ~]$ ls -l /tmp/archive  
total 10537316  
-rw-rw-r--. 1 ec2-user ec2-user      19112 Mar 21 15:57 compat-  
libcap1-1.10-7.el7.x86_64.rpm  
-rw-rw-r--  1 ec2-user ec2-user 3059705302 Mar 21 22:01  
LINUX.X64_193000_db_home.zip  
-rw-rw-r--  1 ec2-user ec2-user 2889184573 Mar 21 21:09  
LINUX.X64_193000_grid_home.zip  
-rw-rw-r--. 1 ec2-user ec2-user      589145 Mar 21 15:56  
netapp_linux_unified_host_utilities-7-1.x86_64.rpm  
-rw-rw-r--. 1 ec2-user ec2-user      31828 Mar 21 15:55 oracle-  
database-preinstall-19c-1.0-2.el8.x86_64.rpm  
-rw-rw-r--  1 ec2-user ec2-user 2872741741 Mar 21 22:31  
p34762026_190000_Linux-x86-64.zip  
-rw-rw-r--  1 ec2-user ec2-user 1843577895 Mar 21 22:32  
p34765931_190000_Linux-x86-64.zip  
-rw-rw-r--  1 ec2-user ec2-user  124347218 Mar 21 22:33  
p6880880_190000_Linux-x86-64.zip  
-rw-r--r--  1 ec2-user ec2-user    257136 Mar 22 16:25  
policycoreutils-python-utils-2.9-9.el8.noarch.rpm
```

3. ほとんどのカーネル構成要件を満たす Oracle 19c プレインストール RPM をインストールします。

```
yum install /tmp/archive/oracle-database-preinstall-19c-1.0-  
2.el8.x86_64.rpm
```

4. 不足しているものをダウンロードしてインストールする`compat-libcap1`Linux 8では。

```
yum install /tmp/archive/compat-libcap1-1.10-7.el7.x86_64.rpm
```

5. NetAppから、NetAppホスト ユーティリティをダウンロードしてインストールします。

```
yum install /tmp/archive/netapp_linux_unified_host_utilities-7-1.x86_64.rpm
```

6. インストール `policycoreutils-python-utils` EC2 インスタンスでは使用できません。

```
yum install /tmp/archive/policycoreutils-python-utils-2.9-9.el8.noarch.rpm
```

7. オープン JDK バージョン 1.8 をインストールします。

```
yum install java-1.8.0-openjdk.x86_64
```

8. iSCSI イニシエーター ユーティリティをインストールします。

```
yum install iscsi-initiator-utils
```

9. インストール sg3_utils。

```
yum install sg3_utils
```

10. インストール device-mapper-multipath。

```
yum install device-mapper-multipath
```

11. 現在のシステムで透過的な巨大ページを無効にします。

```
echo never > /sys/kernel/mm/transparent_hugepage/enabled  
echo never > /sys/kernel/mm/transparent_hugepage/defrag
```

次の行を追加します `etc/rc.local` 無効にする `transparent_hugepage` 再起動後:

```
# Disable transparent hugepages
    if test -f /sys/kernel/mm/transparent_hugepage/enabled;
then
    echo never > /sys/kernel/mm/transparent_hugepage/enabled
fi
    if test -f /sys/kernel/mm/transparent_hugepage/defrag;
then
    echo never > /sys/kernel/mm/transparent_hugepage/defrag
fi
```

12. selinuxを無効にするには SELINUX=enforcing`に `SELINUX=disabled。変更を有効にするには、ホストを再起動する必要があります。

```
vi /etc/sysconfig/selinux
```

13. 次の行を追加します limit.conf`ファイル記述子の制限とスタックサイズを引用符なしで設定する`" "`。

```
vi /etc/security/limits.conf
**                hard    nofile            65536"
**                soft    stack             10240"
```

14. 次の手順に従って、EC2 インスタンスにスワップ領域を追加します。["スワップファイルを使用して、Amazon EC2 インスタンスでスワップ領域として機能するメモリを割り当てるにはどうすればよいですか?"](#)追加するスペースの正確な量は、最大 16G の RAM のサイズによって異なります。
15. 変化 `node.session.timeo.replacement_timeout`の中で `iscsi.conf`設定ファイルを 120 秒から 5 秒に変更します。

```
vi /etc/iscsi/iscsid.conf
```

16. EC2 インスタンスで iSCSI サービスを有効にして開始します。

```
systemctl enable iscsid
systemctl start iscsid
```

17. データベース LUN マッピングに使用する iSCSI イニシエーター アドレスを取得します。

```
cat /etc/iscsi/initiatorname.iscsi
```

18. asm sysasm グループに使用する ASM グループを追加します。

```
groupadd asm
```

19. oracle ユーザーを変更して、ASM をセカンダリ グループとして追加します (oracle ユーザーは、Oracle プレインストール RPM のインストール後に作成されている必要があります)。

```
usermod -a -G asm oracle
```

20. Linux ファイアウォールがアクティブになっている場合は停止して無効にします。

```
systemctl stop firewalld  
systemctl disable firewalld
```

21. EC2 インスタンスを再起動します。

データベースボリュームとLUNをEC2インスタンスホストにプロビジョニングしてマッピングする

Oracle データベース バイナリ、データ、およびログ ファイルをホストするために、FSx クラスター管理 IP を持つ fsxadmin ユーザーとして ssh 経由で FSx クラスターにログインし、コマンド ラインから 3 つのボリュームをプロビジョニングします。

1. fsxadmin ユーザーとして SSH 経由で FSx クラスターにログインします。

```
ssh fsxadmin@172.30.15.53
```

2. 次のコマンドを実行して、Oracle バイナリのボリュームを作成します。

```
vol create -volume ora_01_biny -aggregate aggr1 -size 50G -state  
online -type RW -snapshot-policy none -tiering-policy snapshot-only
```

3. 次のコマンドを実行して、Oracle データ用のボリュームを作成します。

```
vol create -volume ora_01_data -aggregate aggr1 -size 100G -state  
online -type RW -snapshot-policy none -tiering-policy snapshot-only
```

4. 次のコマンドを実行して、Oracle ログ用のボリュームを作成します。

```
vol create -volume ora_01_logs -aggregate aggr1 -size 100G -state  
online -type RW -snapshot-policy none -tiering-policy snapshot-only
```

5. データベース バイナリ ボリューム内にバイナリ LUN を作成します。

```
lun create -path /vol/ora_01_biny/ora_01_biny_01 -size 40G -ostype  
linux
```

6. データベース データ ボリューム内にデータ LUN を作成します。

```
lun create -path /vol/ora_01_data/ora_01_data_01 -size 20G -ostype linux
```

```
lun create -path /vol/ora_01_data/ora_01_data_02 -size 20G -ostype linux
```

```
lun create -path /vol/ora_01_data/ora_01_data_03 -size 20G -ostype linux
```

```
lun create -path /vol/ora_01_data/ora_01_data_04 -size 20G -ostype linux
```

7. データベース ログ ボリューム内にログ LUN を作成します。

```
lun create -path /vol/ora_01_logs/ora_01_logs_01 -size 40G -ostype linux
```

```
lun create -path /vol/ora_01_logs/ora_01_logs_02 -size 40G -ostype linux
```

8. 上記の EC2 カーネル構成のステップ 14 から取得したイニシエーターを使用して、EC2 インスタンスの igroup を作成します。

```
igroup create -igroup ora_01 -protocol iscsi -ostype linux  
-initiator iqn.1994-05.com.redhat:f65fed7641c2
```

9. 上記で作成した igroup に LUN をマップします。ボリューム内の追加 LUN ごとに、LUN ID を順番に増加します。

```

lun map -path /vol/ora_01_biny/ora_01_biny_01 -igroup ora_01
-vserver svm_ora -lun-id 0
lun map -path /vol/ora_01_data/ora_01_data_01 -igroup ora_01
-vserver svm_ora -lun-id 1
lun map -path /vol/ora_01_data/ora_01_data_02 -igroup ora_01
-vserver svm_ora -lun-id 2
lun map -path /vol/ora_01_data/ora_01_data_03 -igroup ora_01
-vserver svm_ora -lun-id 3
lun map -path /vol/ora_01_data/ora_01_data_04 -igroup ora_01
-vserver svm_ora -lun-id 4
lun map -path /vol/ora_01_logs/ora_01_logs_01 -igroup ora_01
-vserver svm_ora -lun-id 5
lun map -path /vol/ora_01_logs/ora_01_logs_02 -igroup ora_01
-vserver svm_ora -lun-id 6

```

10. LUN マッピングを検証します。

```
mapping show
```

次のような結果が返されると予想されます。

```

FsxId02ad7bf3476b741df::> mapping show
(lun mapping show)
Vserver      Path                                          Igroup   LUN ID
Protocol
-----
svm_ora      /vol/ora_01_biny/ora_01_biny_01          ora_01    0
iscsi
svm_ora      /vol/ora_01_data/ora_01_data_01          ora_01    1
iscsi
svm_ora      /vol/ora_01_data/ora_01_data_02          ora_01    2
iscsi
svm_ora      /vol/ora_01_data/ora_01_data_03          ora_01    3
iscsi
svm_ora      /vol/ora_01_data/ora_01_data_04          ora_01    4
iscsi
svm_ora      /vol/ora_01_logs/ora_01_logs_01          ora_01    5
iscsi
svm_ora      /vol/ora_01_logs/ora_01_logs_02          ora_01    6
iscsi

```

データベースストレージ構成

次に、EC2 インスタンス ホストに Oracle グリッド インフラストラクチャとデータベース インストール用の FSx ストレージをインポートしてセットアップします。

1. SSH キーと EC2 インスタンスの IP アドレスを使用して、ec2-user として SSH 経由で EC2 インスタンスにログインします。

```
ssh -i ora_01.pem ec2-user@172.30.15.58
```

2. いずれかの SVM iSCSI IP アドレスを使用して FSx iSCSI エンドポイントを検出します。次に、環境固有のポータル アドレスに変更します。

```
sudo iscsiadm iscsiadm --mode discovery --op update --type sendtargets --portal 172.30.15.51
```

3. 各ターゲットにログインして iSCSI セッションを確立します。

```
sudo iscsiadm --mode node -l all
```

コマンドからの予想される出力は次のとおりです。

```
[ec2-user@ip-172-30-15-58 ~]$ sudo iscsiadm --mode node -l all
Logging in to [iface: default, target: iqn.1992-08.com.netapp:sn.1f795e65c74911edb785affbf0a2b26e:vs.3, portal: 172.30.15.51,3260]
Logging in to [iface: default, target: iqn.1992-08.com.netapp:sn.1f795e65c74911edb785affbf0a2b26e:vs.3, portal: 172.30.15.13,3260]
Login to [iface: default, target: iqn.1992-08.com.netapp:sn.1f795e65c74911edb785affbf0a2b26e:vs.3, portal: 172.30.15.51,3260] successful.
Login to [iface: default, target: iqn.1992-08.com.netapp:sn.1f795e65c74911edb785affbf0a2b26e:vs.3, portal: 172.30.15.13,3260] successful.
```

4. アクティブな iSCSI セッションのリストを表示および検証します。

```
sudo iscsiadm --mode session
```

iSCSI セッションを返します。

```
[ec2-user@ip-172-30-15-58 ~]$ sudo iscsiadm --mode session
tcp: [1] 172.30.15.51:3260,1028 iqn.1992-
08.com.netapp:sn.1f795e65c74911edb785affbf0a2b26e:vs.3 (non-flash)
tcp: [2] 172.30.15.13:3260,1029 iqn.1992-
08.com.netapp:sn.1f795e65c74911edb785affbf0a2b26e:vs.3 (non-flash)
```

5. LUN がホストにインポートされたことを確認します。

```
sudo sanlun lun show
```

これにより、FSx から Oracle LUN のリストが返されます。

```

[ec2-user@ip-172-30-15-58 ~]$ sudo sanlun lun show
controller(7mode/E-Series)/                               device
host                lun
vservers(cDOT/FlashRay)  lun-pathname
filename             adapter  protocol  size  product

svm_ora              /vol/ora_01_logs/ora_01_logs_02
/dev/sdn             host3    iSCSI    40g   cDOT
svm_ora              /vol/ora_01_logs/ora_01_logs_01
/dev/sdm             host3    iSCSI    40g   cDOT
svm_ora              /vol/ora_01_data/ora_01_data_03
/dev/sdk             host3    iSCSI    20g   cDOT
svm_ora              /vol/ora_01_data/ora_01_data_04
/dev/sdl             host3    iSCSI    20g   cDOT
svm_ora              /vol/ora_01_data/ora_01_data_01
/dev/sdi             host3    iSCSI    20g   cDOT
svm_ora              /vol/ora_01_data/ora_01_data_02
/dev/sdj             host3    iSCSI    20g   cDOT
svm_ora              /vol/ora_01_biny/ora_01_biny_01
/dev/sdh             host3    iSCSI    40g   cDOT
svm_ora              /vol/ora_01_logs/ora_01_logs_02
/dev/sdg             host2    iSCSI    40g   cDOT
svm_ora              /vol/ora_01_logs/ora_01_logs_01
/dev/sdf             host2    iSCSI    40g   cDOT
svm_ora              /vol/ora_01_data/ora_01_data_04
/dev/sde             host2    iSCSI    20g   cDOT
svm_ora              /vol/ora_01_data/ora_01_data_02
/dev/sdc             host2    iSCSI    20g   cDOT
svm_ora              /vol/ora_01_data/ora_01_data_03
/dev/sdd             host2    iSCSI    20g   cDOT
svm_ora              /vol/ora_01_data/ora_01_data_01
/dev/sdb             host2    iSCSI    20g   cDOT
svm_ora              /vol/ora_01_biny/ora_01_biny_01
/dev/sda             host2    iSCSI    40g   cDOT

```

6. 設定する `multipath.conf` 次のデフォルトおよびブラックリスト エントリを含むファイル。

```
sudo vi /etc/multipath.conf

defaults {
    find_multipaths yes
    user_friendly_names yes
}

blacklist {
    devnode "^(ram|raw|loop|fd|md|dm-|sr|scd|st) [0-9]*"
    devnode "^hd[a-z]"
    devnode "^cciss.*"
}
```

7. マルチパス サービスを開始します。

```
sudo systemctl start multipathd
```

マルチパスデバイスが `/dev/mapper` ディレクトリ。

```
[ec2-user@ip-172-30-15-58 ~]$ ls -l /dev/mapper
total 0
lrwxrwxrwx 1 root root          7 Mar 21 20:13
3600a09806c574235472455534e68512d -> ../dm-0
lrwxrwxrwx 1 root root          7 Mar 21 20:13
3600a09806c574235472455534e685141 -> ../dm-1
lrwxrwxrwx 1 root root          7 Mar 21 20:13
3600a09806c574235472455534e685142 -> ../dm-2
lrwxrwxrwx 1 root root          7 Mar 21 20:13
3600a09806c574235472455534e685143 -> ../dm-3
lrwxrwxrwx 1 root root          7 Mar 21 20:13
3600a09806c574235472455534e685144 -> ../dm-4
lrwxrwxrwx 1 root root          7 Mar 21 20:13
3600a09806c574235472455534e685145 -> ../dm-5
lrwxrwxrwx 1 root root          7 Mar 21 20:13
3600a09806c574235472455534e685146 -> ../dm-6
crw----- 1 root root 10, 236 Mar 21 18:19 control
```

8. SSH 経由で fsxadmin ユーザーとして FSx クラスターにログインし、6c574xxx... で始まる各 LUN のシリアル 16 進数を取得します。この 16 進数は 3600a0980 で始まり、これは AWS ベンダー ID です。

```
lun show -fields serial-hex
```

次のように返されます。

```
FsxId02ad7bf3476b741df::> lun show -fields serial-hex
vserver path                               serial-hex
-----
svm_ora /vol/ora_01_biny/ora_01_biny_01 6c574235472455534e68512d
svm_ora /vol/ora_01_data/ora_01_data_01 6c574235472455534e685141
svm_ora /vol/ora_01_data/ora_01_data_02 6c574235472455534e685142
svm_ora /vol/ora_01_data/ora_01_data_03 6c574235472455534e685143
svm_ora /vol/ora_01_data/ora_01_data_04 6c574235472455534e685144
svm_ora /vol/ora_01_logs/ora_01_logs_01 6c574235472455534e685145
svm_ora /vol/ora_01_logs/ora_01_logs_02 6c574235472455534e685146
7 entries were displayed.
```

9. 更新する `/dev/multipath.conf` マルチパスデバイスにユーザーフレンドリな名前を追加するファイルです。

```
sudo vi /etc/multipath.conf
```

次のエントリが含まれます:

```
multipaths {
    multipath {
        wwid          3600a09806c574235472455534e68512d
        alias         ora_01_biny_01
    }
    multipath {
        wwid          3600a09806c574235472455534e685141
        alias         ora_01_data_01
    }
    multipath {
        wwid          3600a09806c574235472455534e685142
        alias         ora_01_data_02
    }
    multipath {
        wwid          3600a09806c574235472455534e685143
        alias         ora_01_data_03
    }
    multipath {
        wwid          3600a09806c574235472455534e685144
        alias         ora_01_data_04
    }
    multipath {
        wwid          3600a09806c574235472455534e685145
        alias         ora_01_logs_01
    }
    multipath {
        wwid          3600a09806c574235472455534e685146
        alias         ora_01_logs_02
    }
}
```

10. マルチパスサービスを再起動して、以下のデバイスが `/dev/mapper` シリアル16進IDではなくLUN名に変更されました。

```
sudo systemctl restart multipathd
```

チェック `/dev/mapper` 次のように返されます。

```
[ec2-user@ip-172-30-15-58 ~]$ ls -l /dev/mapper
total 0
crw----- 1 root root 10, 236 Mar 21 18:19 control
lrwxrwxrwx 1 root root      7 Mar 21 20:41 ora_01_biny_01 -> ../dm-
0
lrwxrwxrwx 1 root root      7 Mar 21 20:41 ora_01_data_01 -> ../dm-
1
lrwxrwxrwx 1 root root      7 Mar 21 20:41 ora_01_data_02 -> ../dm-
2
lrwxrwxrwx 1 root root      7 Mar 21 20:41 ora_01_data_03 -> ../dm-
3
lrwxrwxrwx 1 root root      7 Mar 21 20:41 ora_01_data_04 -> ../dm-
4
lrwxrwxrwx 1 root root      7 Mar 21 20:41 ora_01_logs_01 -> ../dm-
5
lrwxrwxrwx 1 root root      7 Mar 21 20:41 ora_01_logs_02 -> ../dm-
6
```

11. バイナリ LUN を単一のプライマリパーティションに分割します。

```
sudo fdisk /dev/mapper/ora_01_biny_01
```

12. パーティション化されたバイナリ LUN を XFS ファイルシステムでフォーマットします。

```
sudo mkfs.xfs /dev/mapper/ora_01_biny_01p1
```

13. バイナリLUNをマウントする /u01。

```
sudo mount -t xfs /dev/mapper/ora_01_biny_01p1 /u01
```

14. 変化`/u01`マウントポイントの所有権を Oracle ユーザーとそれに関連付けられたプライマリグループに付与します。

```
sudo chown oracle:oinstall /u01
```

15. バイナリ LUN の UUI を見つけます。

```
sudo blkid /dev/mapper/ora_01_biny_01p1
```

16. マウントポイントを追加する /etc/fstab。

```
sudo vi /etc/fstab
```

次の行を追加します。

```
UUID=d89fb1c9-4f89-4de4-b4d9-17754036d11d    /u01    xfs
defaults,nofail 0                2
```



EC2 インスタンスの再起動中に発生する可能性のあるルートロックの問題を回避するには、UUID のみと nofail オプションを使用してバイナリをマウントすることが重要です。

17. root ユーザーとして、Oracle デバイスの udev ルールを追加します。

```
vi /etc/udev/rules.d/99-oracle-asmdevices.rules
```

次のエントリを含めます:

```
ENV{DM_NAME}=="ora*", GROUP:="oinstall", OWNER:="oracle",
MODE:="660"
```

18. root ユーザーとして、udev ルールを再ロードします。

```
udevadm control --reload-rules
```

19. root ユーザーとして、udev ルールをトリガーします。

```
udevadm trigger
```

20. root ユーザーとして、multipathd をリロードします。

```
systemctl restart multipathd
```

21. EC2 インスタンス ホストを再起動します。

Oracle グリッド インフラストラクチャのインストール

1. SSH経由でec2-userとしてEC2インスタンスにログインし、コメントアウトを解除してパスワード認証を有効にします。 PasswordAuthentication yes`そしてコメントアウトする
`PasswordAuthentication no。

```
sudo vi /etc/ssh/sshd_config
```

2. sshd サービスを再起動します。

```
sudo systemctl restart sshd
```

3. Oracle ユーザーのパスワードをリセットします。

```
sudo passwd oracle
```

4. Oracle Restart ソフトウェア所有者ユーザー (oracle) としてログインします。次のように Oracle ディレクトリを作成します。

```
mkdir -p /u01/app/oracle  
mkdir -p /u01/app/oraInventory
```

5. ディレクトリの権限設定を変更します。

```
chmod -R 775 /u01/app
```

6. グリッド ホーム ディレクトリを作成し、そこに移動します。

```
mkdir -p /u01/app/oracle/product/19.0.0/grid  
cd /u01/app/oracle/product/19.0.0/grid
```

7. グリッドインストールファイルを解凍します。

```
unzip -q /tmp/archive/LINUX.X64_193000_grid_home.zip
```

8. グリッドホームから、`OPatch`ディレクトリ。

```
rm -rf OPatch
```

9. グリッドホームから解凍 p6880880_190000_Linux-x86-64.zip。

```
unzip -q /tmp/archive/p6880880_190000_Linux-x86-64.zip
```

10. グリッドホームから修正 `cv/admin/cvu_config`` コメントを外して置き換えます
`CV_ASSUME_DISTID=OEL5`と `CV_ASSUME_DISTID=OL7`。

```
vi cv/admin/cvu_config
```

11. 更新 `gridsetup.rsp` サイレントインストール用のファイルを作成し、rsp ファイルを `tmp/archive` ディレクトリ。rsp ファイルは、セクション A、B、および G をカバーし、次の情報を含める必要があります。

```
INVENTORY_LOCATION=/u01/app/oraInventory
oracle.install.option=HA_CONFIG
ORACLE_BASE=/u01/app/oracle
oracle.install.asm.OSDBA=dba
oracle.install.asm.OSOPER=oper
oracle.install.asm.OSASM=asm
oracle.install.asm.SYSASMPassword="SetPWD"
oracle.install.asm.diskGroup.name=DATA
oracle.install.asm.diskGroup.redundancy=EXTERNAL
oracle.install.asm.diskGroup.AUSize=4
oracle.install.asm.diskGroup.disks=/dev/mapper/ora_01_data_01,/dev/mapper/ora_01_data_02,/dev/mapper/ora_01_data_03,/dev/mapper/ora_01_data_04
oracle.install.asm.diskGroup.diskDiscoveryString=/dev/mapper/*
oracle.install.asm.monitorPassword="SetPWD"
oracle.install.asm.configureAFD=true
```

12. EC2 インスタンスにルートユーザーとしてログインし、`ORACLE_HOME``そして `ORACLE_BASE`。

```
export ORACLE_HOME=/u01/app/oracle/product/19.0.0/grid
export ORACLE_BASE=/tmp
cd /u01/app/oracle/product/19.0.0/grid/bin
```

13. Oracle ASM フィルタ ドライバで使用するディスク デバイスをプロビジョニングします。

```
./asmcmd afd_label DATA01 /dev/mapper/ora_01_data_01 --init  
./asmcmd afd_label DATA02 /dev/mapper/ora_01_data_02 --init  
./asmcmd afd_label DATA03 /dev/mapper/ora_01_data_03 --init  
./asmcmd afd_label DATA04 /dev/mapper/ora_01_data_04 --init  
./asmcmd afd_label LOGS01 /dev/mapper/ora_01_logs_01 --init  
./asmcmd afd_label LOGS02 /dev/mapper/ora_01_logs_02 --init
```

14. インストール cvuqdisk-1.0.10-1.rpm。

```
rpm -ivh /u01/app/oracle/product/19.0.0/grid/cv/rpm/cvuqdisk-1.0.10-1.rpm
```

15. 設定解除 \$ORACLE_BASE。

```
unset ORACLE_BASE
```

16. OracleユーザーとしてEC2インスタンスにログインし、`/tmp/archive`フォルダ。

```
unzip /tmp/archive/p34762026_190000_Linux-x86-64.zip -d /tmp/archive
```

17. グリッドホーム/u01/app/oracle/product/19.0.0/gridからoracleユーザーとして起動します。
`gridSetup.sh`グリッド インフラストラクチャのインストール用。

```
./gridSetup.sh -applyRU /tmp/archive/34762026/ -silent  
-responseFile /tmp/archive/gridsetup.rsp
```

グリッド インフラストラクチャの間違ったグループに関する警告は無視してください。 Oracle Restart を管理するために単一の Oracle ユーザーを使用しているため、これは予想された動作です。

18. root ユーザーとして、次のスクリプトを実行します。

```
/u01/app/oraInventory/orainstRoot.sh  
  
/u01/app/oracle/product/19.0.0/grid/root.sh
```

19. root ユーザーとして、multipathd をリロードします。

```
systemctl restart multipathd
```

20. Oracle ユーザーとして、次のコマンドを実行して構成を完了します。

```
/u01/app/oracle/product/19.0.0/grid/gridSetup.sh -executeConfigTools  
-responseFile /tmp/archive/gridsetup.rsp -silent
```

21. Oracle ユーザーとして、\$GRID_HOME から LOGS ディスク グループを作成します。

```
bin/asmca -silent -sysAsmPassword 'yourPWD' -asmsnmpPassword  
'yourPWD' -createDiskGroup -diskGroupName LOGS -disk 'AFD:LOGS*'  
-redundancy EXTERNAL -au_size 4
```

22. Oracle ユーザーとして、インストール構成後にグリッド サービスを検証します。

```
bin/crsctl stat res -t  
+  
Name                          Target  State          Server  
State details  
Local Resources  
ora.DATA.dg                    ONLINE  ONLINE         ip-172-30-15-58  
STABLE  
ora.LISTENER.lsnr              ONLINE  ONLINE         ip-172-30-15-58  
STABLE  
ora.LOGS.dg                     ONLINE  ONLINE         ip-172-30-15-58  
STABLE  
ora.asm                         ONLINE  ONLINE         ip-172-30-15-58  
Started,STABLE  
ora.ons                         OFFLINE  OFFLINE        ip-172-30-15-58  
STABLE  
Cluster Resources  
ora.cssd                       ONLINE  ONLINE         ip-172-30-15-58  
STABLE  
ora.diskmon                    OFFLINE  OFFLINE  
STABLE  
ora.driver.afd                 ONLINE  ONLINE         ip-172-30-15-58  
STABLE  
ora.evmd                       ONLINE  ONLINE         ip-172-30-15-58  
STABLE
```

23. ASM フィルター ドライバーのステータスを検証します。

```
[oracle@ip-172-30-15-58 grid]$ export
ORACLE_HOME=/u01/app/oracle/product/19.0.0/grid
[oracle@ip-172-30-15-58 grid]$ export ORACLE_SID=+ASM
[oracle@ip-172-30-15-58 grid]$ export PATH=$PATH:$ORACLE_HOME/bin
[oracle@ip-172-30-15-58 grid]$ asmcmd
ASMCMDS> lsdg
State      Type      Rebal  Sector  Logical_Sector  Block      AU
Total_MB  Free_MB  Req_mir_free_MB  Usable_file_MB  Offline_disks
Voting_files  Name
MOUNTED  EXTERN  N      512     512     4096    1048576
81920    81847      0      81847      0
N  DATA/
MOUNTED  EXTERN  N      512     512     4096    1048576
81920    81853      0      81853      0
N  LOGS/
ASMCMDS> afd_state
ASMCMDS-9526: The AFD state is 'LOADED' and filtering is 'ENABLED' on
host 'ip-172-30-15-58.ec2.internal'
```

Oracleデータベースのインストール

1. Oracleユーザーとしてログインし、`\$ORACLE_HOME`そして`\$ORACLE_SID`設定されている場合。

```
unset ORACLE_HOME
unset ORACLE_SID
```

2. Oracle DB ホーム ディレクトリを作成し、そこに移動します。

```
mkdir /u01/app/oracle/product/19.0.0/db1
cd /u01/app/oracle/product/19.0.0/db1
```

3. Oracle DB インストール ファイルを解凍します。

```
unzip -q /tmp/archive/LINUX.X64_193000_db_home.zip
```

4. DBホームから、`OPatch`ディレクトリ。

```
rm -rf OPatch
```

5. DBホームから解凍 p6880880_190000_Linux-x86-64.zip。

```
unzip -q /tmp/archive/p6880880_190000_Linux-x86-64.zip
```

6. DBホームから修正 cv/admin/cvu_config`をコメント解除して置き換えます
`CV_ASSUME_DISTID=OEL5`と `CV_ASSUME_DISTID=OL7`。

```
vi cv/admin/cvu_config
```

7. から`/tmp/archive`ディレクトリで、DB 19.18 RU パッチを解凍します。

```
unzip p34765931_190000_Linux-x86-64.zip
```

8. DBサイレントインストールの標準rspファイルを更新します`/tmp/archive/dbinstall.rsp`関連するセクションに次の値を持つディレクトリ:

```
oracle.install.option=INSTALL_DB_SWONLY
UNIX_GROUP_NAME=oinstall
INVENTORY_LOCATION=/u01/app/oraInventory
ORACLE_HOME=/u01/app/oracle/product/19.0.0/db1
ORACLE_BASE=/u01/app/oracle
oracle.install.db.InstallEdition=EE
oracle.install.db.OSDBA_GROUP=dba
oracle.install.db.OSOPER_GROUP=oper
oracle.install.db.OSBACKUPDBA_GROUP=oper
oracle.install.db.OSDGDBA_GROUP=dba
oracle.install.db.OSKMDBA_GROUP=dba
oracle.install.db.OSRACDBA_GROUP=dba
oracle.install.db.rootconfig.executeRootScript=false
```

9. db1 ホーム /u01/app/oracle/product/19.0.0/db1 から、サイレント ソフトウェアのみの DB インストールを実行します。

```
./runInstaller -applyRU /tmp/archive/34765931/ -silent
-ignorePrereqFailure -responseFile /tmp/archive/dbinstall.rsp
```

10. ルートユーザーとして、`root.sh`ソフトウェアのみのインストール後のスクリプト。

```
/u01/app/oracle/product/19.0.0/db1/root.sh
```

11. Oracleユーザーとして標準を更新 `dbca.rsp` 関連するセクションに次のエントリを含むファイルを作成します。

```
gdbName=db1.demo.netapp.com
sid=db1
createAsContainerDatabase=true
numberOfPDBs=3
pdbName=db1_pdb
useLocalUndoForPDBs=true
pdbAdminPassword="yourPWD"
templateName=General_Purpose.dbc
sysPassword="yourPWD"
systemPassword="yourPWD"
dbsnmpPassword="yourPWD"
datafileDestination=+DATA
recoveryAreaDestination=+LOGS
storageType=ASM
diskGroupName=DATA
characterSet=AL32UTF8
nationalCharacterSet=AL16UTF16
listeners=LISTENER
databaseType=MULTIPURPOSE
automaticMemoryManagement=false
totalMemory=8192
```

12. Oracle ユーザーとして、\$ORACLE_HOME ディレクトリから dbca を使用して DB 作成を開始します。

```
bin/dbca -silent -createDatabase -responseFile /tmp/archive/dbca.rsp
```

output:

Prepare for db operation

7% complete

Registering database with Oracle Restart

11% complete

Copying database files

33% complete

Creating and starting Oracle instance

35% complete

38% complete

42% complete

45% complete

48% complete

Completing Database Creation

53% complete

55% complete

56% complete

Creating Pluggable Databases

60% complete

64% complete

69% complete

78% complete

Executing Post Configuration Actions

100% complete

Database creation complete. For details check the logfiles at:

/u01/app/oracle/cfgtoollogs/dbca/db1.

Database Information:

Global Database Name:db1.demo.netapp.com

System Identifier(SID):db1

Look at the log file "/u01/app/oracle/cfgtoollogs/dbca/db1/db1.log"
for further details.

13. Oracle ユーザーとして、DB 作成後に Oracle Restart HA サービスを検証します。

```
[oracle@ip-172-30-15-58 db1]$ ../grid/bin/crsctl stat res -t
```

Name	Target	State	Server	State
Local Resources				
ora.DATA.dg	ONLINE	ONLINE	ip-172-30-15-58	STABLE
ora.LISTENER.lsnr	ONLINE	ONLINE	ip-172-30-15-58	STABLE
ora.LOGS.dg	ONLINE	ONLINE	ip-172-30-15-58	STABLE
ora.asm	ONLINE	ONLINE	ip-172-30-15-58	Started,STABLE
ora.ons	OFFLINE	OFFLINE	ip-172-30-15-58	STABLE
Cluster Resources				
ora.cssd	ONLINE	ONLINE	ip-172-30-15-58	STABLE
ora.dbf.db	ONLINE	ONLINE	ip-172-30-15-58	Open,HOME=/u01/app/oracle/product/19.0.0/db1,STABLE
ora.diskmon	OFFLINE	OFFLINE		STABLE
ora.driver.afd	ONLINE	ONLINE	ip-172-30-15-58	STABLE
ora.evmd	ONLINE	ONLINE	ip-172-30-15-58	STABLE

14. Oracleユーザーを設定する .bash_profile。

```
vi ~/.bash_profile
```

15. 次のエントリを追加します。

```
export ORACLE_HOME=/u01/app/oracle/product/19.0.0/db1
export ORACLE_SID=db1
export PATH=$PATH:$ORACLE_HOME/bin
alias asm='export
ORACLE_HOME=/u01/app/oracle/product/19.0.0/grid;export
ORACLE_SID=+ASM;export PATH=$PATH:$ORACLE_HOME/bin'
```

16. 作成された CDB/PDB を検証します。

```
source /home/oracle/.bash_profile

sqlplus / as sysdba
```

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

```
NAME          OPEN_MODE
```

```
DB1           READ WRITE
```

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

```
NAME
```

```
+DATA/DB1/DATAFILE/system.256.1132176177
```

```
+DATA/DB1/DATAFILE/sysaux.257.1132176221
```

```
+DATA/DB1/DATAFILE/undotbs1.258.1132176247
```

```
+DATA/DB1/86B637B62FE07A65E053F706E80A27CA/DATAFILE/system.265.1132177009
```

```
+DATA/DB1/86B637B62FE07A65E053F706E80A27CA/DATAFILE/sysaux.266.1132177009
```

```
+DATA/DB1/DATAFILE/users.259.1132176247
```

```
+DATA/DB1/86B637B62FE07A65E053F706E80A27CA/DATAFILE/undotbs1.267.1132177009
```

```
+DATA/DB1/F7852758DCD6B800E0533A0F1EAC1DC6/DATAFILE/system.271.1132177853
```

```
+DATA/DB1/F7852758DCD6B800E0533A0F1EAC1DC6/DATAFILE/sysaux.272.1132177853
```

```
+DATA/DB1/F7852758DCD6B800E0533A0F1EAC1DC6/DATAFILE/undotbs1.270.1132177853
```

```
+DATA/DB1/F7852758DCD6B800E0533A0F1EAC1DC6/DATAFILE/users.274.1132177871
```

```
NAME
```

```
+DATA/DB1/F785288BBCD1BA78E0533A0F1EACCD6F/DATAFILE/system.276.1132177871
```

```
+DATA/DB1/F785288BBCD1BA78E0533A0F1EACCD6F/DATAFILE/sysaux.277.1132177871
```

```
+DATA/DB1/F785288BBCD1BA78E0533A0F1EACCD6F/DATAFILE/undotbs1.275.1132177871
```

```
+DATA/DB1/F785288BBCD1BA78E0533A0F1EACCD6F/DATAFILE/users.279.1132177889
```

```
+DATA/DB1/F78529A14DD8BB18E0533A0F1EACB8ED/DATAFILE/system.281.1132177889
```

```
+DATA/DB1/F78529A14DD8BB18E0533A0F1EACB8ED/DATAFILE/sysaux.282.1132177889
```

```
+DATA/DB1/F78529A14DD8BB18E0533A0F1EACB8ED/DATAFILE/undotbs1.280.1132177889
```

```
+DATA/DB1/F78529A14DD8BB18E0533A0F1EACB8ED/DATAFILE/users.284.1132177907
```

```
19 rows selected.
```

```
SQL> show pdbs
```

CON_ID	CON_NAME	OPEN MODE	RESTRICTED
2	PDB\$SEED	READ ONLY	NO
3	DB1_PDB1	READ WRITE	NO
4	DB1_PDB2	READ WRITE	NO
5	DB1_PDB3	READ WRITE	NO

```
SQL>
```

17. DB リカバリ先のサイズを +LOGS ディスク グループのサイズに設定します。

```
alter system set db_recovery_file_dest_size = 80G scope=both;
```

18. sqlplus を使用してデータベースにログインし、アーカイブ ログ モードを有効にします。

```
sqlplus /as sysdba.  
  
shutdown immediate;  
  
startup mount;  
  
alter database archivelog;  
  
alter database open;
```

これで、Amazon FSx ONTAPおよび EC2 コンピューティングインスタンスでの Oracle 19c バージョン 19.18 の再起動のデプロイメントが完了します。必要に応じて、NetAppOracle 制御ファイルとオンライン ログ ファイルを +LOGS ディスク グループに再配置することを推奨します。

自動展開オプション

参照"[TR-4986: iSCSI を使用したAmazon FSx ONTAPでの簡素化された自動化された Oracle デプロイメント](#)" 詳細については。

SnapCenterサービスによる Oracle データベースのバックアップ、リストア、クローン作成

見る "[Oracle向けSnapCenterサービス](#)" NetApp BlueXPコンソールを使用した Oracle データベースのバックアップ、リストア、およびクローン作成の詳細については、こちらをご覧ください。

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。

- 新しいデータベースのインストールによるスタンドアロン・サーバーへのOracle Grid Infrastructureのインストール

["https://docs.oracle.com/en/database/oracle/oracle-database/19/ladbi/installing-oracle-grid-infrastructure-for-a-standalone-server-with-a-new-database-installation.html#GUID-0B1CEE8C-C893-46AA-8A6A-7B5FAAEC72B3"](https://docs.oracle.com/en/database/oracle/oracle-database/19/ladbi/installing-oracle-grid-infrastructure-for-a-standalone-server-with-a-new-database-installation.html#GUID-0B1CEE8C-C893-46AA-8A6A-7B5FAAEC72B3)

- レスポンスファイルを使用したOracle Databaseのインストールと構成

["https://docs.oracle.com/en/database/oracle/oracle-database/19/ladbi/installing-and-configuring-oracle-database-using-response-files.html#GUID-D53355E9-E901-4224-9A2A-B882070EDDF7"](https://docs.oracle.com/en/database/oracle/oracle-database/19/ladbi/installing-and-configuring-oracle-database-using-response-files.html#GUID-D53355E9-E901-4224-9A2A-B882070EDDF7)

- 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=Cj0KCQiA54KfBhCKARIsAJzSrdqwQrghn6l71jiWzSeaT9Uh1-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 © 2026 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。