



オンプレミス/ハイブリッドクラウド NetApp Solutions

NetApp
May 17, 2024

目次

オンプレミス/ハイブリッドクラウド	1
TR-4983 : 『Simplified、Automated Oracle Deployment on NetApp ASA with iSCSI』	1
NVA-1155 : 『FlexPod Datacenter with Cisco UCS and NetApp AFF A800 over FC-Design and Deployment Guide』 で、Oracle 19C RACデータベースを検証します	18
TR-4250 : 『SAP with Oracle on UNIX and NFS with NetApp Clustered Data ONTAP and SnapManager for SAP 3.4』	18
Oracle データベースを導入しています	18
解決策の概要	40
TR-4794 : 『Oracle databases on NetApp EF Series』	64

オンプレミス/ハイブリッドクラウド

TR-4983 : 『Simplified、Automated Oracle Deployment on NetApp ASA with iSCSI』

ネットアップ、Niyaz Mohamed、Allen Cao氏

この解決策では、プライマリデータベースストレージとしてNetApp ASAアレイにOracleの導入と保護を自動化するための概要と詳細を説明します。プライマリデータベースストレージはiSCSIプロトコルで、Oracleデータベースはスタンドアロンの再起動で設定されます。ASMはボリュームマネージャとして使用します。

目的

NetApp ASAシステムは、SANインフラに最新のソリューションを提供します。大規模環境を簡易化し、データベースなどのビジネスクリティカルなアプリケーションを高速化し、データの可用性を常に維持（99.9999%のアップタイム）し、TCOと二酸化炭素排出量を削減できます。NetApp ASAシステムには、パフォーマンス要件がきわめて高いアプリケーション向けに設計されたAシリーズモデルと、対費用効果の高い大容量環境向けに最適化されたCシリーズモデルがあります。ASA AシリーズとCシリーズのシステムを組み合わせることで、卓越したパフォーマンスが実現し、カスタマーエクスペリエンスの向上と成果達成までの時間の短縮、ビジネスクリティカルなデータの可用性、保護、セキュリティの維持、あらゆるワークロードの実効容量の増加を実現できます。これには、業界で最も効果的な保証が付随しています。

このドキュメントでは、ASAシステムで構築されたSAN環境に、Ansibleによる自動化を使用してOracleデータベースを簡単に導入する方法について説明します。Oracleデータベースは、データアクセス用にiSCSIプロトコルを使用し、ASAストレージアレイでのデータベースディスク管理用にOracle ASMを使用して、スタンドアロンの再起動構成で導入されます。また、NetApp SnapCenter UIツールを使用したOracleデータベースのバックアップ、リストア、およびクローニングに関する情報も提供し、NetApp ASAシステムでのストレージ効率に優れたデータベース処理を実現します。

この解決策 は、次のユースケースに対応します。

- プライマリデータベースストレージとしてのNetApp ASAシステムへのOracleデータベース導入の自動化
- NetApp SnapCenterツールを使用したNetApp ASAシステムでのOracleデータベースのバックアップとリストア
- NetApp SnapCenterツールを使用したNetApp ASAシステムでの開発/テストなどのユースケース向けのOracleデータベースのクローン

対象者

この解決策 は、次のユーザーを対象としています。

- NetApp ASAシステムにOracleを導入したいと考えているデータベース管理者。
- データベース解決策アーキテクト。NetApp ASAシステムでOracleワークロードをテストしたいと考えています。
- NetApp ASAシステムにOracleデータベースを導入して管理したいストレージ管理者。

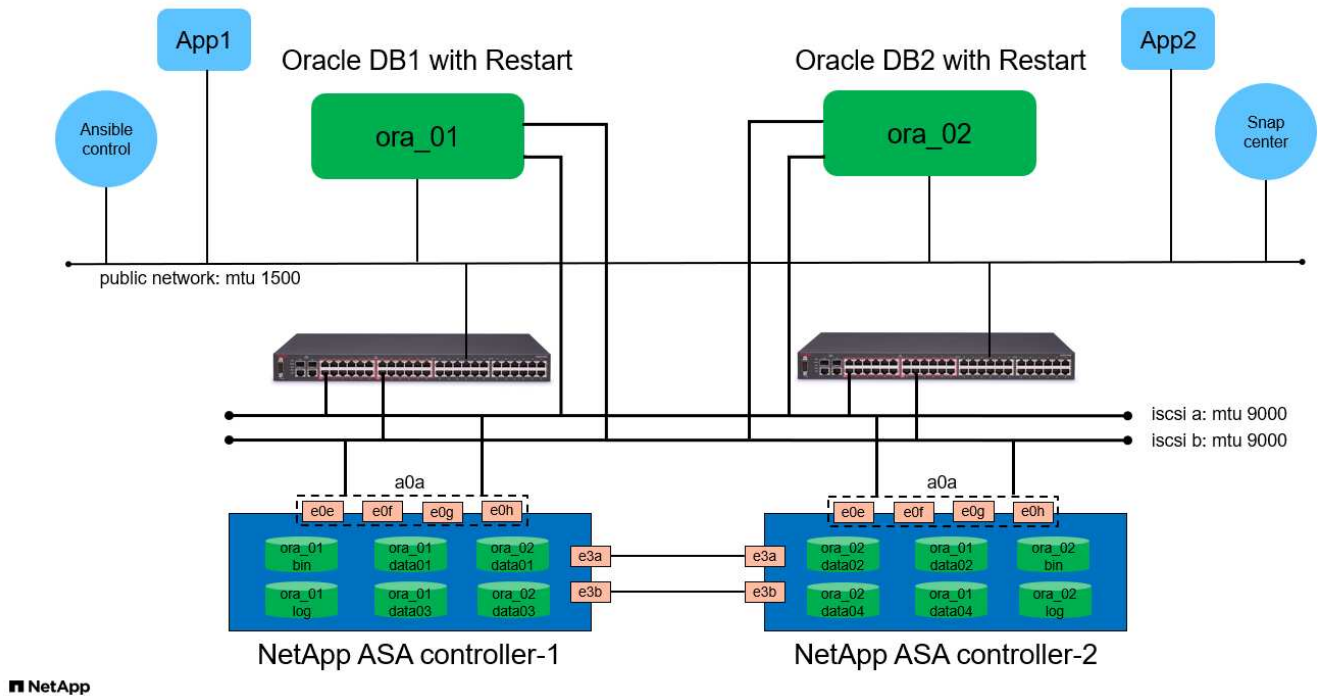
- NetApp ASAシステムでOracleデータベースを構築するアプリケーション所有者。

解決策 のテストおよび検証環境

この解決策のテストと検証は、最終的な導入環境とは一致しない可能性があるラボ環境で実行しました。を参照してください [\[導入にあたって考慮すべき主要因\]](#) を参照してください。

アーキテクチャ

Simplified, Automated Oracle Database Deployment on NetApp ASA with iSCSI



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 *		
NetApp ASA A400	バージョン9.13.1P1	NS224シェルフ×2、NVMe AFFドライブ×48、合計容量69.3TiB
UCSB-B2004-M4	Intel (R) Xeon (R) CPU E5-2690 v4 (2.60GHz)	4ノードのVMware ESXiクラスター
ソフトウェア		
Red Hat Linux	RHEL-8.6、4.18.0-372.9.1.el8.x86_64カーネル	テスト用にRedHatサブスクリプションを導入
Windows Serverの場合	2022 Standard、10.0.20348ビルド20348	SnapCenterサアハノホスト
Oracle Grid Infrastructureの略	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 データベース	バージョン19.18	RUパッチp34765931_190000_Linux-x86-64.zipを適用しました
Oracle OPatchの略	バージョン12.2.0.1.36	最新のパッチp6880880_190000_Linux-x86-64.zip
SnapCenter サーバ	バージョン4.9P1	ワークグループの導入
VMware vSphere ハイパーバイザー	バージョン6.5.0.20000	VMware Tools、バージョン：11365-Linux、12352-Windows
JDKを開く	バージョンjava-1.8.0-openjdk.x86_64	DB VMでのSnapCenterプラグインの要件

ラボ環境でのOracleデータベースの構成

* サーバ *	* データベース *	* DBストレージ*
ORA_01	NTAP1 (NTAP1_PDB1、NTAP1_PDB2、NTAP1_PDB3)	ASA A400のiSCSI LUN
ORA_02	NTAP2 (NTAP2_PDB1、NTAP2_PDB2、NTAP2_PDB3)	ASA A400のiSCSI LUN

導入にあたって考慮すべき主な要因

- * Oracleデータベースのストレージレイアウト。*このOracleの自動導入では、デフォルトで4つのデータベースボリュームをプロビジョニングして、Oracleのバイナリ、データ、ログをホストします。次に、データLUNとログLUNから2つのASMディスクグループを作成します。+data ASMディスクグループ内で、各ASA A400クラスタノードの1つのボリュームに2つのデータ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、1つのASA A400ノードのログボリュームに2つのLUNを作成します。ONTAPボリューム内に複数のLUNをレイアウトすると、一般的にパフォーマンスが向上します。
- *複数のDBサーバの導入。*自動化解決策では、1回のAnsibleプレイブック実行でOracleコンテナデータベースを複数のDBサーバに導入できます。DBサーバの数に関係なく、プレイブックの実行は変わりません。複数のDBサーバを導入する場合は、データベースLUNをASA A400のデュアルコントローラに最適に配置するアルゴリズムを使用してPlaybookを構築します。バイナリと奇数のDBサーバのLUNは、コントローラ1のサーバホストインデックスプレースに記録されます。バイナリと偶数のDBサーバのLUNは、コントローラ2のサーバホストインデックスプレースに記録されます。DBデータLUNは2台のコントローラに均等に分散されます。Oracle ASMは、2台のコントローラ上のデータLUNを1つのASMディスクグループに統合して、両方のコントローラの処理能力を最大限に活用します。
- * iSCSI構成。*データベースVMは、ストレージアクセス用にiSCSIプロトコルを使用してASAストレージに接続します。冗長性を確保するために各コントローラノードでデュアルパスを設定し、マルチパスストレージアクセス用にDBサーバでiSCSIマルチパスを設定する必要があります。パフォーマンスとスループットを最大化するには、ストレージネットワークでジャンボフレームを有効にします。
- 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。ASA A400では、クラスタディスクレベルでデータ保護のためにRAID DPでストレージが構成されるため、次を使用する必要があります。`External Redundancy`これは、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を許可しないことを意味します。
- データベースのバックアップ。NetAppは、データベースのバックアップ、リストア、クローニングを実行するためのSnapCenterソフトウェアスイートで、使いやすいUIインターフェイスを備えています。

す。NetAppでは、このような管理ツールを実装して、高速（1分未満）のSnapshotバックアップ、高速（数分）のデータベースリストア、データベースクローンを実現することを推奨しています。

解決策 の導入

以降のセクションでは、直接マウントされたデータベースLUNを使用するNetApp ASA A400で、単一ノードのiSCSI経由でDB VMに直接マウントされたOracle 19Cの導入と保護を自動化するための手順を段階的に説明します。Oracle ASMをデータベースボリュームマネージャとして使用して構成を再起動します。

導入の前提条件

導入には、次の前提条件が必要です。

1. ここでは、NetApp ASAストレージレイが設置および設定されていることを前提としています。これには、iSCSIブロードキャストドメイン、両方のコントローラード上のLACPインターフェイスグループa0a、両方のコントローラード上のiSCSI VLANポート（a0a-<iscsi-a-vlan-id>、a0a-<iscsi-b-vlan-id>）が含まれます。ヘルプが必要な場合の詳細な手順については、次のリンクを参照してください。 ["詳細ガイド-ASA A400"](#)
2. 最新バージョンのAnsibleとGitがインストールされたAnsibleコントローラードとしてLinux VMをプロビジョニングします。詳細については、次のリンクを参照してください。 ["NetApp解決策 自動化の導入"](#) セクション- Setup the Ansible Control Node for CLI deployments on RHEL / CentOS または Setup the Ansible Control Node for CLI deployments on Ubuntu / Debian。
3. iSCSI用のNetApp Oracle Deployment Automation Toolkitのコピーをクローニングします。

```
git clone https://bitbucket.ngage.netapp.com/scm/ns-bb/na_oracle_deploy_iscsi.git
```

4. NetApp SnapCenter UIツールを最新バージョンで実行するようにWindowsサーバをプロビジョニングします。詳細については、次のリンクを参照してください。 ["SnapCenter サーバをインストールします"](#)
5. ベアメタルまたは仮想VMのRHEL Oracle DBサーバを2台構築します。パスワード権限なしでsudoを使用してDBサーバに管理者ユーザを作成し、AnsibleホストとOracle DBサーバホストの間でSSHの秘密鍵/公開鍵認証を有効にします。Oracle 19CインストールファイルをDBサーバ/tmp/archiveディレクトリにステージングします。

```
installer_archives:  
- "LINUX.X64_193000_grid_home.zip"  
- "p34762026_190000_Linux-x86-64.zip"  
- "LINUX.X64_193000_db_home.zip"  
- "p34765931_190000_Linux-x86-64.zip"  
- "p6880880_190000_Linux-x86-64.zip"
```



Oracle VMのルートボリュームに少なくとも50Gが割り当てられており、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が確保されていることを確認してください。

6. 次のビデオをご覧ください。

[iSCSIを使用したNetApp ASAへのOracle導入の簡易化と自動化](#)

自動化パラメータファイル

Ansible Playbookは、事前定義されたパラメータを使用してデータベースのインストールと設定のタスクを実行します。このOracle自動化解決策では、プレイブックを実行する前にユーザ入力が必要な3つのユーザ定義パラメータファイルがあります。

- Hosts -自動化プレイブックの実行対象となるターゲットを定義します。
- vars/vars.yml -すべてのターゲットに適用される変数を定義するグローバル変数ファイル。
- host_vars/host_name.yml -ローカルターゲットにのみ適用される変数を定義するローカル変数ファイル。今回のユースケースでは、これらがOracle DBサーバです。

これらのユーザー定義変数ファイルに加えて、必要でない限り変更を必要としないデフォルトパラメータを含むデフォルトの変数ファイルがいくつかあります。次のセクションでは、ユーザー定義の変数ファイルの設定方法について説明します。

パラメータファイルの設定

1. Ansibleターゲット hosts ファイル構成：

```
# Enter NetApp ASA controller management IP address
[ontap]
172.16.9.32

# Enter Oracle servers names to be deployed one by one, follow by
each Oracle server public IP address, and ssh private key of admin
user for the server.
[oracle]
ora_01 ansible_host=10.61.180.21 ansible_ssh_private_key_file
=ora_01.pem
ora_02 ansible_host=10.61.180.23 ansible_ssh_private_key_file
=ora_02.pem
```

2. グローバル vars/vars.yml ファイル構成

```
#####
#####
#####
                                Oracle 19c deployment global user
configurable variables                                #####
#####
                                Consolidate all variables from ONTAP, linux
and oracle                                #####
#####
#####

#####
#####
#####
                                ONTAP env specific config variables
#####
#####
#####

# Enter the supported ONTAP platform: on-prem, aws-fsx.
ontap_platform: on-prem

# Enter ONTAP cluster management user credentials
username: "xxxxxxxx"
password: "xxxxxxxx"

##### on-prem platform specific user defined variables #####

# Enter Oracle SVM iSCSI lif addresses. Each controller configures
```

```

with dual paths iscsi_a, iscsi_b for redundancy
ora_iscsi_lif_mgmt:
  - {name: '{{ svm_name }}_lif_mgmt', address: 172.21.253.220, netmask:
255.255.255.0, vlan_name: ora_mgmt, vlan_id: 3509}

ora_iscsi_lifs_node1:
  - {name: '{{ svm_name }}_lif_1a', address: 172.21.234.221,
netmask: 255.255.255.0, vlan_name: ora_iscsi_a, vlan_id: 3490}
  - {name: '{{ svm_name }}_lif_1b', address: 172.21.235.221,
netmask: 255.255.255.0, vlan_name: ora_iscsi_b, vlan_id: 3491}
ora_iscsi_lifs_node2:
  - {name: '{{ svm_name }}_lif_2a', address: 172.21.234.223,
netmask: 255.255.255.0, vlan_name: ora_iscsi_a, vlan_id: 3490}
  - {name: '{{ svm_name }}_lif_2b', address: 172.21.235.223,
netmask: 255.255.255.0, vlan_name: ora_iscsi_b, vlan_id: 3491}

#####
#####
###                               Linux env specific config variables
###
#####
#####

# Enter RHEL subscription to enable repo
redhat_sub_username: xxxxxxxx
redhat_sub_password: "xxxxxxxx"

#####
#####
###                               Oracle DB env specific config variables
###
#####
#####

# Enter Database domain name
db_domain: solutions.netapp.com

# Enter initial password for all required Oracle passwords. Change
them after installation.
initial_pwd_all: xxxxxxxx

```

3. ローカルDBサーバ host_vars/host_name.yml 構成

```
# User configurable Oracle host specific parameters

# Enter container database SID. By default, a container DB is
created with 3 PDBs within the CDB
oracle_sid: NTAP1

# Enter database shared memory size or SGA. CDB is created with SGA
at 75% of memory_limit, MB. The grand total of SGA should not exceed
75% available RAM on node.
memory_limit: 8192
```

Playbookの実施

自動化ツールキットには、合計6つのプレイブックが用意されています。それぞれが異なるタスクブロックを実行し、さまざまな目的に対応します。

```
0-all_playbook.yml - execute playbooks from 1-4 in one playbook run.
1-ansible_requirements.yml - set up Ansible controller with required
libs and collections.
2-linux_config.yml - execute Linux kernel configuration on Oracle DB
servers.
3-ontap_config.yml - configure ONTAP svm/volumes/luns for Oracle
database and grant DB server access to luns.
4-oracle_config.yml - install and configure Oracle on DB servers for
grid infrastructure and create a container database.
5-destroy.yml - optional to undo the environment to dismantle all.
```

次のコマンドを使用してプレイブックを実行する方法は3つあります。

1. すべての導入プレイブックを1回の組み合わせで実行します。

```
ansible-playbook -i hosts 0-all_playbook.yml -u admin -e
@vars/vars.yml
```

2. 1～4の番号順でプレイブックを1つずつ実行します。

```
ansible-playbook -i hosts 1-ansible_requirements.yml -u admin -e
@vars/vars.yml
```

```
ansible-playbook -i hosts 2-linux_config.yml -u admin -e
@vars/vars.yml
```

```
ansible-playbook -i hosts 3-ontap_config.yml -u admin -e
@vars/vars.yml
```

```
ansible-playbook -i hosts 4-oracle_config.yml -u admin -e
@vars/vars.yml
```

3. タグを指定して0-all_playbook.ymlを実行します。

```
ansible-playbook -i hosts 0-all_playbook.yml -u admin -e  
@vars/vars.yml -t ansible_requirements
```

```
ansible-playbook -i hosts 0-all_playbook.yml -u admin -e  
@vars/vars.yml -t linux_config
```

```
ansible-playbook -i hosts 0-all_playbook.yml -u admin -e  
@vars/vars.yml -t ontap_config
```

```
ansible-playbook -i hosts 0-all_playbook.yml -u admin -e  
@vars/vars.yml -t oracle_config
```

4. 環境を元に戻す

```
ansible-playbook -i hosts 5-destroy.yml -u admin -e @vars/vars.yml
```

実行後の検証

Playbookの実行後、Oracle DBサーバにOracleユーザとしてログインし、Oracleグリッドインフラとデータベースが正常に作成されたことを確認します。次に、ホストora_01でのOracleデータベース検証の例を示します。

1. 作成したグリッドインフラとリソースを検証します。

```
[oracle@ora_01 ~]$ df -h
Filesystem                Size      Used Avail Use% Mounted on
devtmpfs                  7.7G       40K   7.7G   1% /dev
tmpfs                     7.8G      1.1G   6.7G  15% /dev/shm
tmpfs                     7.8G      312M   7.5G   4% /run
tmpfs                     7.8G         0   7.8G   0% /sys/fs/cgroup
/dev/mapper/rhel-root      44G       38G   6.8G  85% /
/dev/sda1                 1014M     258M   757M  26% /boot
tmpfs                     1.6G       12K   1.6G   1% /run/user/42
tmpfs                     1.6G       4.0K   1.6G   1% /run/user/1000
/dev/mapper/ora_01_biny_01p1 40G      21G    20G  52% /u01
[oracle@ora_01 ~]$ asm
[oracle@ora_01 ~]$ crsctl stat res -t
-----
-----
Name                Target  State          Server                State
details
-----
-----
Local Resources
-----
-----
ora.DATA.dg
                ONLINE  ONLINE          ora_01                STABLE
ora.LISTENER.lsnr
                ONLINE  INTERMEDIATE   ora_01                Not All
Endpoints Re
gistered, STABLE
ora.LOGS.dg
                ONLINE  ONLINE          ora_01                STABLE
ora.asm
                ONLINE  ONLINE          ora_01
Started, STABLE
ora.ons
                OFFLINE OFFLINE         ora_01                STABLE
-----
-----
Cluster Resources
```

```

-----
ora.cssd
      1      ONLINE  ONLINE      ora_01      STABLE
ora.diskmon
      1      OFFLINE OFFLINE      STABLE
ora.driver.afd
      1      ONLINE  ONLINE      ora_01      STABLE
ora.evmd
      1      ONLINE  ONLINE      ora_01      STABLE
ora.ntap1.db
      1      ONLINE  ONLINE      ora_01
Open,HOME=/u01/app/o

racle/product/19.0.0

/NTAP1,STABLE
-----
-----
[oracle@ora_01 ~]$

```



無視する Not All Endpoints Registered 州内の詳細。これは、リスナーとの手動および動的なデータベース登録の競合が原因で発生するため、無視しても問題ありません。

2. ASMフィルタドライバが正常に動作していることを確認します。


```

[oracle@ora_01 ~]$ asmcmd
ASMCMD> 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      4194304
327680     318644      0       318644      0
N  DATA/
MOUNTED    EXTERN    N       512      512      4096      4194304
81920      78880      0       78880      0
N  LOGS/
ASMCMD> lsdk
Path
AFD:ORA_01_DAT1_01
AFD:ORA_01_DAT1_03
AFD:ORA_01_DAT1_05
AFD:ORA_01_DAT1_07
AFD:ORA_01_DAT2_02
AFD:ORA_01_DAT2_04
AFD:ORA_01_DAT2_06
AFD:ORA_01_DAT2_08
AFD:ORA_01_LOGS_01
AFD:ORA_01_LOGS_02
ASMCMD> afd_state
ASMCMD-9526: The AFD state is 'LOADED' and filtering is 'ENABLED' on
host 'ora_01'
ASMCMD>


```

3. Oracle Enterprise Manager Expressにログインして、データベースを検証します。

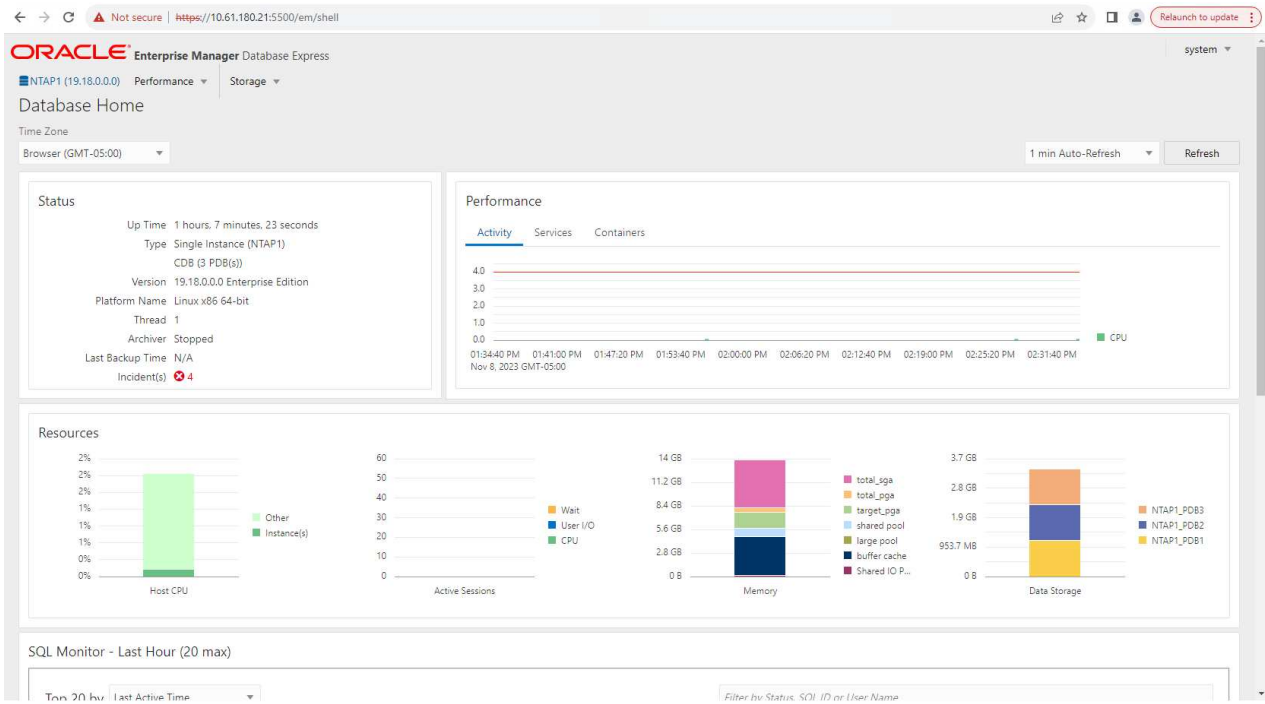
← → ↻ ⚠ Not secure | https://10.61.180.21:5500/em/login



ORACLE ENTERPRISE MANAGER DATABASE EXPRESS



Copyright 2013, 2020, Oracle and/or its affiliates. All rights reserved.



Enable additional port from sqlplus for login to individual container database or PDBs.

```
SQL> show pdbs
```

CON_ID	CON_NAME	OPEN MODE	RESTRICTED
2	PDB\$SEED	READ ONLY	NO
3	NTAP1_PDB1	READ WRITE	NO
4	NTAP1_PDB2	READ WRITE	NO
5	NTAP1_PDB3	READ WRITE	NO

```
SQL> alter session set container=NTAP1_PDB1;
```

Session altered.

```
SQL> select dbms_xdb_config.gethttpsport() from dual;
```

```
DBMS_XDB_CONFIG.GETHTTPS()
-----
0
```

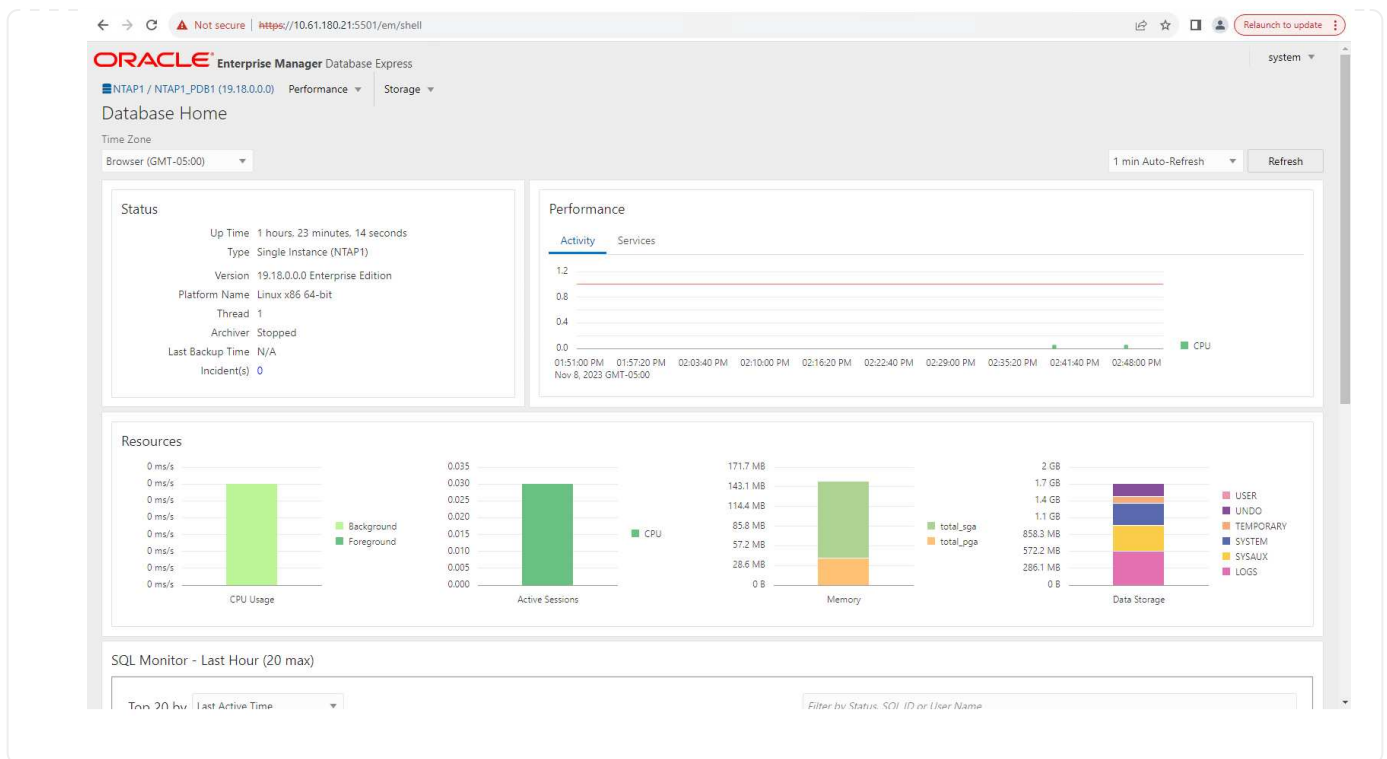
```
SQL> exec DBMS_XDB_CONFIG.SETHTTPS(5501);
```

PL/SQL procedure successfully completed.

```
SQL> select dbms_xdb_config.gethttpsport() from dual;
```

```
DBMS_XDB_CONFIG.GETHTTPS()
-----
5501
```

login to NTAP1_PDB1 from port 5501.



SnapCenterによるOracleのバックアップ、リストア、クローニング

TR-4979を参照 ["ゲストマウント型FSx ONTAPにより、VMware Cloud on AWS上のシンプルで自己管理型のOracleを実現"](#) セクション。Oracle backup, restore, and clone with SnapCenter
SnapCenterのセットアップと、データベースのバックアップ、リストア、クローニングのワークフローの実行の詳細については、[こちら](#)を参照してください。

追加情報の参照先

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、以下のドキュメントや Web サイトを参照してください。

- NetApp ASA：オールフラッシュSANアレイ

["https://www.netapp.com/data-storage/all-flash-san-storage-array/"](https://www.netapp.com/data-storage/all-flash-san-storage-array/)

- 新規データベースをインストールしたスタンドアロンサーバー用の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)

- 応答ファイルを使用したOracleデータベースのインストールと設定

["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)

- ONTAPでRed Hat Enterprise Linux 8.2を使用する

NVA-1155 : 『FlexPod Datacenter with Cisco UCS and NetApp AFF A800 over FC-Design and Deployment Guide』で、Oracle 19C RACデータベースを検証します

ネットアップ Allen Cao

この設計および導入ガイドでは、FlexPod Datacenter with Cisco UCS and NetApp AFF A800 over FCを使用したOracle 19C RACデータベースについて説明します。解決策 また、Oracle Linux 8.2を使用した最新のFlexPod データセンターインフラストラクチャでOracle RACデータベースをホストするためのステップバイステップの導入プロセスについても説明しますオペレーティングシステムとRed Hat互換カーネル。

["NVA-1155 : FlexPod DatacenterにOracle 19C RACデータベースを格納し、FC経由でCisco UCSおよびNetApp AFF A800を使用する"](#)

TR-4250 : 『SAP with Oracle on UNIX and NFS with NetApp Clustered Data ONTAP and SnapManager for SAP 3.4』

ネットアップ Nils Bauer

TR-4250は、Oracleデータベースを使用してSAP Business Suite製品をサポートするストレージソリューションを設計する際の課題に対応しています。ストレージインフラの設計、導入、運用、管理を中心に、最新世代の SAP ソリューションの利用に際して経営陣や IT 部門責任者が一般に直面する課題について、その対処方法を示します。本書で紹介する推奨事項は、特定の SAP アプリケーションに限定したものではなく、規模や範囲に関係なくさまざまな SAP 環境に応用できます。TR-4250は、ネットアップ製品およびSAP製品のテクノロジーと運用に関する基本的な知識があることを前提としています。TR-4250は、ネットアップ、SAP、Oracle、およびお客様の技術スタッフの協力のもとに開発されました。

["TR-4250 : 『SAP with Oracle on UNIX and NFS with NetApp Clustered Data ONTAP and SnapManager for SAP 3.4』"](#)

Oracle データベースを導入しています

解決策の概要

このページでは、NetApp ONTAP ストレージ上に Oracle19c を導入するための自動化方式について説明します。

Oracle19c for ONTAP の NFS への自動導入

組織は環境を自動化して、効率を高め、導入を高速化し、手動作業を削減しています。Ansible などの構成管理ツールを使用して、エンタープライズデータベースの運用を合理化しています。この解決策では、Ansible を使用して、Oracle 19C のプロビジョニングと設定を NetApp ONTAP で自動化する方法を紹介します。ストレージ管理者、システム管理者、DBA は、新しいストレージの一貫した迅速な導入、データベースサーバの構成、Oracle 19C ソフトウェアのインストールを可能にすることで、次のようなメリットを得ることができます。

- 設計の複雑さと人為的ミスを排除し、繰り返し実行可能な一貫した導入とベストプラクティスを実装する
- ストレージのプロビジョニング、DB ホストの構成、Oracle のインストールにかかる時間を短縮
- データベース管理者、システム管理者、ストレージ管理者の生産性を向上
- ストレージとデータベースを簡単に拡張できます

ネットアップは、検証済みの Ansible モジュールとロールをお客様に提供し、Oracle データベース環境の導入、構成、ライフサイクル管理を迅速化します。この解決策では、以下の作業に役立つ Ansible の Playbook コードを提供しています。

- Oracle データベース用の ONTAP NFS ストレージを作成して設定します
- Oracle 19C を Red Hat Enterprise Linux 7/8 または Oracle にインストールします Linux 7/8.
- ONTAP NFS ストレージ上に Oracle 19C を設定します

詳細または概要については、以下の概要ビデオをご覧ください。

AWX / タワー型の導入

パート 1 : はじめに、要件、自動化の詳細、AWX/Tower の初期構成

AWXの導入

パート 2 : 変数とプレイブックの実行

AWX Playbookの実行

CLI の導入

パート 1 : はじめに、要件、自動化の詳細、Ansible Control Host Setup を確認する

CLI の導入

パート 2 : 変数とプレイブックの実行

CLI Playbookの実行

はじめに

この解決策は、AWX/Tower 環境または Ansible コントロールホストの CLI で実行されるように設計されています。

AWX / タワー環境の場合は、ONTAP クラスタ管理と Oracle サーバ（IP およびホスト名）のインベントリの作成、クレデンシャルの作成、NetApp Automation Github から Ansible コードを取得するプロジェクトの設定、および自動化を開始するジョブテンプレートの設定を案内されます。

1. 環境に固有の変数を入力し、ジョブテンプレートのその他の VAR フィールドにコピーして貼り付けます。
2. ジョブテンプレートに変数を追加したら、自動化を起動できます。
3. ジョブテンプレートは、ontap/config、linux_config、および ORACLE_config のタグを指定することで、3つのフェーズで実行されます。

Ansible コントロールホストを介して **CLI** に接続します

1. Ansible 制御ホストとして使用できるように Linux ホストを設定するには、["詳細な手順については、ここをクリックしてください"](#)
2. Ansible 制御ホストが設定されたら、Ansible Automation リポジトリのクローンを Git で作成できます。
3. ONTAP クラスタ管理 IP および Oracle サーバの管理 IP の IP またはホスト名を使用して hosts ファイルを編集してください。
4. 環境に固有の変数を入力し '変数 .yml ファイルにコピーして貼り付けます
5. 各 Oracle ホストには、ホスト固有の変数を含むホスト名で識別される変数ファイルがあります。
6. すべての変数ファイルが完了したら 'ONTAP_config' 'linux_config' および 'ORACLE_config' のタグを指定することで '3つのフェーズでプレイブックを実行できます

要件

環境	要件
* Ansible 環境 *	AWX/Tower または Linux ホストを Ansible コントロールホストにします
	Ansible v.2.10 以上
	Python 3.
	Python ライブラリ - NetApp-lib-xmltodict-jmespath
* ONTAP *	ONTAP バージョン 9.3-9.7
	データアグリゲート × 2
	NFS VLAN および ifgrp が作成されました
* Oracle サーバ *	RHEL 7/8
	Oracle Linux 7/8
	NFS、パブリック、オプションの管理用のネットワークインターフェイス
	Oracle サーバ上の Oracle インストールファイル

自動化の詳細

この自動導入は、3つのロールで構成される Ansible プレイブックを使用して設計されています。ロールは ONTAP、Linux、Oracle の各構成に対応しています。次の表に、自動化されるタスクを示します。

ロール	タスク
* ONTAP_CONFIG *	ONTAP 環境の事前チェック
	Oracle 用の NFS ベースの SVM の作成
	エクスポートポリシーが作成されました
	Oracle 用のボリュームの作成
	NFS LIF の作成
linux_config	マウントポイントを作成し、NFS ボリュームをマウント
	NFS マウントを確認
	OS 固有の設定
	Oracle ディレクトリを作成します
	hugepages を設定します
	SELinux とファイアウォールデーモンを無効にする
	サービスを有効にして開始します
	ファイル記述子のハードリミットを増やします
	pam.d セッションファイルを作成します
* ORACLE_CONFIG *	Oracle ソフトウェアのインストール
	Oracle リスナーを作成します
	Oracle データベースを作成します
	Oracle 環境構成
	PDB 状態を保存します
	インスタンスアーカイブモードを有効にします
	DNFS クライアントを有効にしてください
	OS のリブート間のデータベースの自動起動とシャットダウンを有効にします

デフォルトパラメータ

自動化を簡易化するために、必要な Oracle 導入パラメータが多数デフォルト値であらかじめ設定されています。通常、ほとんどの環境でデフォルトパラメータを変更する必要はありません。上級ユーザーは 'デフォルト・パラメータを変更する際に注意してくださいデフォルトのパラメータは、各ロールフォルダの defaults ディレクトリにあります。

導入手順

開始する前に '次の Oracle インストール・ファイルとパッチ・ファイルをダウンロードし '/tmp/archive' ディレクトリに配置しますこのディレクトリには '展開する各 DB サーバ上のすべてのユーザに対する読み取り '書き込み 'および実行のアクセス権が含まれます自動化タスクは、その特定のディレクトリにある指定されたインストールファイルを検索して、Oracle のインストールと構成を行います。

```
LINUX.X64_193000_db_home.zip -- 19.3 base installer
p31281355_190000_Linux-x86-64.zip -- 19.8 RU patch
p6880880_190000_Linux-x86-64.zip -- opatch version 12.2.0.1.23
```

使用許諾

Github リポジトリに記載されているライセンス情報をお読みください。このリポジトリ内のコンテンツにアクセス、ダウンロード、インストール、または使用することにより、ライセンスの条項に同意したものとみなされます ["こちらをご覧ください"](#)。

このリポジトリ内のコンテンツの作成および / または派生著作物の共有に関しては、一定の制限事項があります。の条件を必ずお読みください ["使用許諾"](#) コンテンツを使用する前に。すべての条件に同意しない場合は、このリポジトリのコンテンツにアクセスしたり、コンテンツをダウンロードしたり、使用したりしないでください。

準備ができれば、をクリックします ["AWX/Tower の導入手順の詳細については、こちらを参照してください"](#) または ["CLI の導入については、こちらをご覧ください"](#)。

ステップバイステップの導入手順

このページでは、NetApp ONTAP ストレージ上に Oracle19c を導入するための自動化方式について説明します。

AWX/Tower の導入 Oracle 19C データベース

1. 環境のインベントリ、グループ、ホスト、およびクレデンシャルを作成します

このセクションでは、ネットアップの自動化ソリューションを使用する環境を準備するための AWX/Ansible タワーでのインベントリ、グループ、ホスト、アクセスクレデンシャルのセットアップについて説明します。

1. インベントリを設定します。

- リソース→インベントリ→追加と進み、インベントリの追加をクリックします。
- 名前と組織の詳細を入力し、[保存] をクリックします。
- インベントリページで、作成されたインベントリをクリックします。
- インベントリ変数がある場合は、その変数を変数フィールドに貼り付けます。
- [グループ] サブメニューに移動し、[追加] をクリックします。
- ONTAP のグループの名前を入力し、グループ変数（ある場合）を貼り付けて、[保存] をクリックします。
- Oracle の別のグループに対してこの手順を繰り返します。
- 作成した ONTAP グループを選択し、Hosts サブメニューに移動して、Add New Host をクリックします。
- ONTAP クラスタ管理 IP の IP アドレスを入力し、ホスト変数（存在する場合）を貼り付けて、[保存] をクリックします。
- このプロセスは、Oracle グループおよび Oracle ホストの管理 IP / ホスト名に対して繰り返す必要があります。

2. クレデンシャルタイプを作成する。ONTAP を使用するソリューションでは、ユーザ名とパスワードのエントリを照合するようにクレデンシャルタイプを設定する必要があります。
 - a. [管理] → [資格情報の種類] に移動し、[追加] をクリックします。
 - b. 名前と概要を指定します。
 - c. 入力構成に次の内容を貼り付けます。

```
fields:
  - id: username
    type: string
    label: Username
  - id: password
    type: string
    label: Password
    secret: true
  - id: vsadmin_password
    type: string
    label: vsadmin_password
    secret: true
```

1. 次の内容をインジェクター設定に貼り付けます。

```
extra_vars:
  password: '{{ password }}'
  username: '{{ username }}'
  vsadmin_password: '{{ vsadmin_password }}'
```

1. クレデンシャルを設定します。
 - a. [リソース] → [資格情報] に移動し、[追加] をクリックします。
 - b. ONTAP の名前と組織の詳細を入力します。
 - c. ONTAP 用に作成したカスタム資格情報タイプを選択します。
 - d. [タイプの詳細] で、ユーザー名、パスワード、および vsadmin-readonly を入力します。
 - e. [資格情報に戻る] をクリックし、[追加] をクリックします
 - f. Oracle の名前と組織の詳細を入力します。
 - g. マシンクレデンシャルタイプを選択します。
 - h. Type Details （タイプの詳細）に、Oracle ホストのユーザー名とパスワードを入力します。
 - i. 適切な特権昇格方式を選択し、ユーザ名とパスワードを入力します。

2. プロジェクトを作成します

1. [リソース] → [プロジェクト] に移動し、[追加] をクリックします。

- a. 名前と組織の詳細を入力します
- b. Source Control Credential Type フィールドで Git を選択します。
- c. 入力するコマンド <https://github.com/NetApp-Automation/na_oracle19c_deploy.git> をソース管理 URL として指定します。
- d. [保存] をクリックします。
- e. ソースコードが変更されたときに、プロジェクトの同期が必要になることがあります。

3. Oracle host_vars を設定します

このセクションで定義した変数は、個々の Oracle サーバとデータベースに適用されます。

1. 次の組み込み Oracle ホスト変数または host_vars フォームに、環境固有のパラメータを入力します。



青の項目は、環境に合わせて変更する必要があります。

ホスト VAR 構成

```
#####
##### Host Variables Configuration #####
#####

# Add your Oracle Host
ansible_host: "10.61.180.15"

# Oracle db log archive mode: true - ARCHIVELOG or false - NOARCHIVELOG
log_archive_mode: "true"

# Number of pluggable databases per container instance identified by sid.
Pdb_name specifies the prefix for container database naming in this case
cdb2_pdb1, cdb2_pdb2, cdb2_pdb3
oracle_sid: "cdb2"
pdb_num: "3"
pdb_name: "{{ oracle_sid }}_pdb"

# CDB listener port, use different listener port for additional CDB on
same host
listener_port: "1523"

# CDB is created with SGA at 75% of memory_limit, MB. Consider how many
databases to be hosted on the node and how much ram to be allocated to
each DB. The grand total SGA should not exceed 75% available RAM on node.
memory_limit: "5464"

# Set "em_configuration: DBEXPRESS" to install enterprise manager express
and choose a unique port from 5500 to 5599 for each sid on the host.
```

```
# Leave them black if em express is not installed.
em_configuration: "DBEXPRESS"
em_express_port: "5501"

# {{groups.oracle[0]}} represents first Oracle DB server as defined in
Oracle hosts group [oracle]. For concurrent multiple Oracle DB servers
deployment, [0] will be incremented for each additional DB server. For
example, {{groups.oracle[1]}} represents DB server 2,
"{{groups.oracle[2]}}" represents DB server 3 ... As a good practice and
the default, minimum three volumes is allocated to a DB server with
corresponding /u01, /u02, /u03 mount points, which store oracle binary,
oracle data, and oracle recovery files respectively. Additional volumes
can be added by click on "More NFS volumes" but the number of volumes
allocated to a DB server must match with what is defined in global vars
file by volumes_nfs parameter, which dictates how many volumes are to be
created for each DB server.
host_datastores_nfs:
  - {vol_name: "{{groups.oracle[0]}}_u01", aggr_name: "aggr01_node01",
lif: "172.21.94.200", size: "25"}
  - {vol_name: "{{groups.oracle[0]}}_u02", aggr_name: "aggr01_node01",
lif: "172.21.94.200", size: "25"}
  - {vol_name: "{{groups.oracle[0]}}_u03", aggr_name: "aggr01_node01",
lif: "172.21.94.200", size: "25"}
```

1. 青のフィールドにすべての変数を入力します。
2. 変数の入力完了したら、フォームの [コピー] ボタンをクリックして、AWX またはタワーに転送されるすべての変数をコピーします。
3. AWX またはタワーに戻って、Resources (リソース) → Hosts (ホスト) に移動し、Oracle サーバ設定ページを選択して開きます。
4. [詳細] タブで、編集をクリックし、コピーした変数を手順 1 から YAML タブの [変数] フィールドに貼り付けます。
5. [保存] をクリックします。
6. システム内の他の Oracle サーバについても、この手順を繰り返します。

4. グローバル変数を設定します

このセクションで定義する変数は、すべての Oracle ホスト、データベース、および ONTAP クラスタに適用されます。

1. 次の組み込みグローバル変数または変数フォームに環境固有のパラメータを入力します。



青の項目は、環境に合わせて変更する必要があります。

```
#####
```

```

##### Oracle 19c deployment global user configuration variables #####
##### Consolidate all variables from ontap, linux and oracle #####
#####

#####

### Ontap env specific config variables ###
#####

#Inventory group name
#Default inventory group name - 'ontap'
#Change only if you are changing the group name either in inventory/hosts
file or in inventory groups in case of AWX/Tower
hosts_group: "ontap"

#CA_signed_certificates (ONLY CHANGE to 'true' IF YOU ARE USING CA SIGNED
CERTIFICATES)
ca_signed_certs: "false"

#Names of the Nodes in the ONTAP Cluster
nodes:
  - "AFF-01"
  - "AFF-02"

#Storage VLANs
#Add additional rows for vlans as necessary
storage_vlans:
  - {vlan_id: "203", name: "infra_NFS", protocol: "NFS"}
More Storage VLANsEnter Storage VLANs details

#Details of the Data Aggregates that need to be created
#If Aggregate creation takes longer, subsequent tasks of creating volumes
may fail.
#There should be enough disks already zeroed in the cluster, otherwise
aggregate create will zero the disks and will take long time
data_aggregates:
  - {aggr_name: "aggr01_node01"}
  - {aggr_name: "aggr01_node02"}

#SVM name
svm_name: "ora_svm"

# SVM Management LIF Details
svm_mgmt_details:
  - {address: "172.21.91.100", netmask: "255.255.255.0", home_port: "e0M"}

# NFS storage parameters when data_protocol set to NFS. Volume named after

```

Oracle hosts name identified by mount point as follow for oracle DB server 1. Each mount point dedicates to a particular Oracle files: u01 - Oracle binary, u02 - Oracle data, u03 - Oracle redo. Add additional volumes by click on "More NFS volumes" and also add the volumes list to corresponding host_vars as host_datastores_nfs variable. For multiple DB server deployment, additional volumes sets needs to be added for additional DB server. Input variable "{{groups.oracle[1]}}_u01", "{{groups.oracle[1]}}_u02", and "{{groups.oracle[1]}}_u03" as vol_name for second DB server. Place volumes for multiple DB servers alternately between controllers for balanced IO performance, e.g. DB server 1 on controller node1, DB server 2 on controller node2 etc. Make sure match lif address with controller node.

volumes_nfs:

- {vol_name: "{{groups.oracle[0]}}_u01", aggr_name: "aggr01_node01", lif: "172.21.94.200", size: "25"}
- {vol_name: "{{groups.oracle[0]}}_u02", aggr_name: "aggr01_node01", lif: "172.21.94.200", size: "25"}
- {vol_name: "{{groups.oracle[0]}}_u03", aggr_name: "aggr01_node01", lif: "172.21.94.200", size: "25"}

#NFS LIFs IP address and netmask

nfs_lifs_details:

- address: "172.21.94.200" #for node-1
netmask: "255.255.255.0"
- address: "172.21.94.201" #for node-2
netmask: "255.255.255.0"

#NFS client match

client_match: "172.21.94.0/24"

Linux env specific config variables ###
#####

#NFS Mount points for Oracle DB volumes

mount_points:

- "/u01"
- "/u02"
- "/u03"

Up to 75% of node memory size divided by 2mb. Consider how many databases to be hosted on the node and how much ram to be allocated to each DB.

```
# Leave it blank if hugepage is not configured on the host.

hugepages_nr: "1234"

# RedHat subscription username and password

redhat_sub_username: "xxx"
redhat_sub_password: "xxx"

#####
### DB env specific install and config variables ###
#####

db_domain: "your.domain.com"

# Set initial password for all required Oracle passwords. Change them
after installation.

initial_pwd_all: "netapp123"
```

1. すべての変数を青のフィールドに入力します。

2. 変数の入力完了したら、フォームの [コピー] ボタンをクリックして、AWX またはタワーに転送されるすべての変数を次のジョブテンプレートにコピーします。

5. ジョブテンプレートを設定して起動します。

1. ジョブテンプレートを作成します。

- a. [リソース] → [テンプレート] → [追加] に移動し、[ジョブテンプレートの追加] をクリックします。
- b. 名前と概要を入力します
- c. ジョブタイプを選択します。Run は、プレイブックに基づいてシステムを設定します。Check は、実際にシステムを設定することなく、プレイブックの事前チェックを実行します。
- d. 対応するインベントリ、プロジェクト、プレイブック、およびクレデンシャルを選択します。
- e. 実行するデフォルトのプレイブックとして、all_cplaybook.yml を選択します。
- f. 手順 4 からコピーしたグローバル変数を YAML タブの Template Variables フィールドに貼り付けます。
- g. [ジョブタグ] フィールドの [起動時にプロンプトを表示する] チェックボックスをオンにします。
- h. [保存] をクリックします。

2. ジョブテンプレートを起動します。

- a. [リソース] → [テンプレート] に移動します。
- b. 目的のテンプレートをクリックし、[起動] をクリックします。
- c. ジョブタグの起動時にプロンプトが表示されたら、requires_config と入力します。requires_config の下にある Create Job Tag 行をクリックして、ジョブタグを入力する必要がある場合があります。



requirement_config により、他のロールを実行するための正しいライブラリが確保されます。

1. [次へ] をクリックし、[起動] をクリックしてジョブを開始します。
2. ジョブの出力と進行状況を監視するには、表示→ジョブをクリックします。
3. ジョブタグの起動を求めるプロンプトが表示されたら、「ONTAP_config」と入力します。ジョブタグを入力するには、ONTAP_config の下にある「ジョブタグの作成」行をクリックする必要があります。
4. [次へ] をクリックし、[起動] をクリックしてジョブを開始します。
5. ジョブ出力および進行状況を監視するには、表示→ジョブをクリックします 進捗状況
6. ONTAP_CONFIG ロールの完了後、linux_config のプロセスを再度実行します。
7. [リソース] → [テンプレート] に移動します。
8. 目的のテンプレートを選択し、[起動] をクリックします。
9. linux_config でジョブタグタイプの起動時にプロンプトが表示されたら、linux_config のすぐ下にある「ジョブタグの作成」行を選択して、ジョブタグを入力する必要があります。
10. [次へ] をクリックし、[起動] をクリックしてジョブを開始します。
11. ジョブの出力と進行状況を監視するには、表示→ジョブを選択します。
12. linux_config ロールが完了したら、ORACLE_config のプロセスを再度実行します。
13. [リソース] → [テンプレート] に移動します。
14. 目的のテンプレートを選択し、[起動] をクリックします。
15. ジョブタグの起動時にプロンプトが表示されたら、ORACLE_config と入力します。ORACLE_config の直下にある「ジョブタグの作成」行を選択して、ジョブタグを入力する必要がある場合があります。
16. [次へ] をクリックし、[起動] をクリックしてジョブを開始します。
17. ジョブの出力と進行状況を監視するには、表示→ジョブを選択します。

6. 同じ Oracle ホストに追加のデータベースを配置します

このプレイブックの Oracle 部分では、1 回の実行につき Oracle サーバ上に Oracle コンテナデータベースが 1 つ作成されます。同じサーバ上に追加のコンテナデータベースを作成するには、次の手順を実行します。

1. host_vars 変数を改訂。
 - a. 手順 2 - Oracle host_vars の設定に戻ります。
 - b. Oracle SID を別の名前文字列に変更します。
 - c. リスナーポートを別の番号に変更します。
 - d. EM Express をインストールする場合は、EM Express ポートを別の番号に変更します。
 - e. 改訂されたホスト変数を Host Configuration Detail タブの Oracle Host Variables フィールドにコピーして貼り付けます。
2. ORACLE_config タグのみを使用して、導入ジョブテンプレートを起動します。
3. Oracle ユーザとして Oracle サーバにログインし、次のコマンドを実行します。

```
ps -ef | grep ora
```



インストールが正常に完了した場合は、Oracle プロセスが一覧表示されます Oracle DB のサポートを開始しました

4. データベースにログインして、次のコマンドセットを使用して作成されたDB設定およびPDBを確認します。


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

SQL*Plus: Release 19.0.0.0.0 - Production on Thu May 6 12:52:51 2021
Version 19.8.0.0.0

Copyright (c) 1982, 2019, Oracle. All rights reserved.

Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.8.0.0.0

SQL>

SQL> select name, log_mode from v$database;
NAME          LOG_MODE
-----
CDB2          ARCHIVELOG

SQL> show pdbs

      CON_ID CON_NAME                                OPEN MODE  RESTRICTED
-----
          2 PDB$SEED                                READ ONLY  NO
          3 CDB2_PDB1                                READ WRITE NO
          4 CDB2_PDB2                                READ WRITE NO
          5 CDB2_PDB3                                READ WRITE NO

col svrname form a30
col dirname form a30
select svrname, dirname, nfsversion from v$dnfs_servers;

SQL> col svrname form a30
SQL> col dirname form a30
SQL> select svrname, dirname, nfsversion from v$dnfs_servers;

SVRNAME                                DIRNAME                                NFSVERSION
-----
172.21.126.200                        /rhelora03_u02                        NFSv3.0
172.21.126.200                        /rhelora03_u03                        NFSv3.0
172.21.126.200                        /rhelora03_u01                        NFSv3.0
```

これにより、dNFSが正常に動作していることが確認されます。

5. 次のコマンドを使用して'リスナー経由でデータベースに接続し'Oracleリスナーの構成を確認します適切なリスナーポートとデータベースサービス名に変更します。

```
[oracle@localhost ~]$ sqlplus
system@//localhost:1523/cdb2_pdb1.cie.netapp.com

SQL*Plus: Release 19.0.0.0.0 - Production on Thu May 6 13:19:57 2021
Version 19.8.0.0.0

Copyright (c) 1982, 2019, Oracle. All rights reserved.

Enter password:
Last Successful login time: Wed May 05 2021 17:11:11 -04:00

Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.8.0.0.0

SQL> show user
USER is "SYSTEM"
SQL> show con_name
CON_NAME
CDB2_PDB1
```

これにより、Oracleリスナーが正常に動作していることが確認されます。

サポートが必要な場所

ツールキットに関するサポートが必要な場合は、にご参加ください ["ネットアップの解決策自動化コミュニティでは、余裕期間のチャンネルがサポートさ"](#) また、ソリューション自動化チャンネルを検索して、質問や問い合わせを投稿しましょう。

ステップバイステップの導入手順

このドキュメントでは、自動コマンドラインインターフェイス（CLI）を使用したOracle 19Cの導入について詳しく説明します。

CLI による Oracle 19C データベースの導入

このセクションでは、CLI を使用して Oracle19c データベースを準備および導入するために必要な手順について説明します。を確認しておきます ["「はじめに」および「要件」セクション"](#) それに応じて環境の準備を整えます。

Oracle19c repo をダウンロードします

1. Ansibleコントローラで、次のコマンドを実行します。

```
git clone https://github.com/NetApp-Automation/na_oracle19c_deploy.git
```

2. リポジトリをダウンロードしたら、ディレクトリをna_oracle19c_deploy <cd na_oracle19c_deploy>に変更します。

hosts ファイルを編集します

導入前に、次の手順を実行します。

1. hosts ファイル na_oracle19c_deploy ディレクトリを編集します。
2. ONTAP で、IP アドレスをクラス管理 IP に変更します。
3. [Oracle] グループの下に、Oracle ホスト名を追加します。DNS または hosts ファイルを使用してホスト名を IP アドレスに解決しておくか、ホストで指定する必要があります。
4. これらの手順を完了したら、変更を保存します。

次の例は、ホストファイルを示しています。

```
#ONTAP Host

[ontap]

"10.61.184.183"

#Oracle hosts

[oracle]

"rtpora01"

"rtpora02"
```

この例では、Playbook を実行し、Oracle 19C を 2 台の Oracle DB サーバに同時に導入しています。1 つの DB サーバでテストすることもできます。この場合、設定が必要なホスト変数ファイルは 1 つだけです。



このプレイブックの内容は、導入する Oracle ホストとデータベースの数に関係なく同じです。

host_vars で host_name.yml ファイルを編集します

各 Oracle ホストには、ホスト固有の変数を含むホスト名で識別されるホスト変数ファイルがあります。ホストには任意の名前を指定できます。Host VAR Config セクションから「host_vars」を編集してコピーし、目的の「host_name.yml」ファイルに貼り付けます。



青の項目は、環境に合わせて変更する必要があります。

ホスト VAR 構成

```
#####
```

```

##### Host Variables Configuration #####
#####

# Add your Oracle Host
ansible_host: "10.61.180.15"

# Oracle db log archive mode: true - ARCHIVELOG or false - NOARCHIVELOG
log_archive_mode: "true"

# Number of pluggable databases per container instance identified by sid.
Pdb_name specifies the prefix for container database naming in this case
cdb2_pdb1, cdb2_pdb2, cdb2_pdb3
oracle_sid: "cdb2"
pdb_num: "3"
pdb_name: "{{ oracle_sid }}_pdb"

# CDB listener port, use different listener port for additional CDB on
same host
listener_port: "1523"

# CDB is created with SGA at 75% of memory_limit, MB. Consider how many
databases to be hosted on the node and how much ram to be allocated to
each DB. The grand total SGA should not exceed 75% available RAM on node.
memory_limit: "5464"

# Set "em_configuration: DBEXPRESS" to install enterprise manager express
and choose a unique port from 5500 to 5599 for each sid on the host.
# Leave them blank if em express is not installed.
em_configuration: "DBEXPRESS"
em_express_port: "5501"

# {{groups.oracle[0]}} represents first Oracle DB server as defined in
Oracle hosts group [oracle]. For concurrent multiple Oracle DB servers
deployment, [0] will be incremented for each additional DB server. For
example, {{groups.oracle[1]}} represents DB server 2,
"{{groups.oracle[2]}}" represents DB server 3 ... As a good practice and
the default, minimum three volumes is allocated to a DB server with
corresponding /u01, /u02, /u03 mount points, which store oracle binary,
oracle data, and oracle recovery files respectively. Additional volumes
can be added by click on "More NFS volumes" but the number of volumes
allocated to a DB server must match with what is defined in global vars
file by volumes_nfs parameter, which dictates how many volumes are to be
created for each DB server.
host_datastores_nfs:
  - {vol_name: "{{groups.oracle[0]}}_u01", aggr_name: "aggr01_node01",
    lif: "172.21.94.200", size: "25"}

```

```
- {vol_name: "{{groups.oracle[0]}}_u02", aggr_name: "aggr01_node01",
  lif: "172.21.94.200", size: "25"}
- {vol_name: "{{groups.oracle[0]}}_u03", aggr_name: "aggr01_node01",
  lif: "172.21.94.200", size: "25"}
```

vars.yml ファイルを編集します

変数 .yml` ファイルは 'Oracle の導入に向けて' 環境固有のすべての変数（ ONTAP 'Linux'Oracle ）を統合します

1. 変数を VAR セクションから編集してコピーし、変数を自分の「 vars.yml 」ファイルに貼り付けます。

```
#####
##### Oracle 19c deployment global user configuration variables #####
##### Consolidate all variables from ontap, linux and oracle #####
#####

#####
### Ontap env specific config variables ###
#####

#Inventory group name
#Default inventory group name - 'ontap'
#Change only if you are changing the group name either in inventory/hosts
file or in inventory groups in case of AWX/Tower
hosts_group: "ontap"

#CA_signed_certificates (ONLY CHANGE to 'true' IF YOU ARE USING CA SIGNED
CERTIFICATES)
ca_signed_certs: "false"

#Names of the Nodes in the ONTAP Cluster
nodes:
  - "AFF-01"
  - "AFF-02"

#Storage VLANs
#Add additional rows for vlans as necessary
storage_vlans:
  - {vlan_id: "203", name: "infra_NFS", protocol: "NFS"}
More Storage VLANsEnter Storage VLANs details

#Details of the Data Aggregates that need to be created
#If Aggregate creation takes longer, subsequent tasks of creating volumes
may fail.
#There should be enough disks already zeroed in the cluster, otherwise
```

```

aggregate create will zero the disks and will take long time
data_aggregates:
  - {aggr_name: "aggr01_node01"}
  - {aggr_name: "aggr01_node02"}

#SVM name
svm_name: "ora_svm"

# SVM Management LIF Details
svm_mgmt_details:
  - {address: "172.21.91.100", netmask: "255.255.255.0", home_port: "e0M"}

# NFS storage parameters when data_protocol set to NFS. Volume named after
Oracle hosts name identified by mount point as follow for oracle DB server
1. Each mount point dedicates to a particular Oracle files: u01 - Oracle
binary, u02 - Oracle data, u03 - Oracle redo. Add additional volumes by
click on "More NFS volumes" and also add the volumes list to corresponding
host_vars as host_datastores_nfs variable. For multiple DB server
deployment, additional volumes sets needs to be added for additional DB
server. Input variable "{{groups.oracle[1]}}_u01",
 "{{groups.oracle[1]}}_u02", and "{{groups.oracle[1]}}_u03" as vol_name for
second DB server. Place volumes for multiple DB servers alternately
between controllers for balanced IO performance, e.g. DB server 1 on
controller node1, DB server 2 on controller node2 etc. Make sure match lif
address with controller node.

volumes_nfs:
  - {vol_name: "{{groups.oracle[0]}}_u01", aggr_name: "aggr01_node01",
    lif: "172.21.94.200", size: "25"}
  - {vol_name: "{{groups.oracle[0]}}_u02", aggr_name: "aggr01_node01",
    lif: "172.21.94.200", size: "25"}
  - {vol_name: "{{groups.oracle[0]}}_u03", aggr_name: "aggr01_node01",
    lif: "172.21.94.200", size: "25"}

#NFS LIFs IP address and netmask

nfs_lifs_details:
  - address: "172.21.94.200" #for node-1
    netmask: "255.255.255.0"
  - address: "172.21.94.201" #for node-2
    netmask: "255.255.255.0"

#NFS client match

client_match: "172.21.94.0/24"

```

```
#####
### Linux env specific config variables ###
#####

#NFS Mount points for Oracle DB volumes

mount_points:
  - "/u01"
  - "/u02"
  - "/u03"

# Up to 75% of node memory size divided by 2mb. Consider how many
databases to be hosted on the node and how much ram to be allocated to
each DB.
# Leave it blank if hugepage is not configured on the host.

hugepages_nr: "1234"

# RedHat subscription username and password

redhat_sub_username: "xxx"
redhat_sub_password: "xxx"

#####
### DB env specific install and config variables ###
#####

db_domain: "your.domain.com"

# Set initial password for all required Oracle passwords. Change them
after installation.

initial_pwd_all: "netapp123"
```

プレイブックを実行します

必要な環境の前提条件を完了し '変数を vars.yml' および 'Your_host.yml' にコピーした後 'プレイブックを導入する準備が整いました



<username> は、環境に合わせて変更する必要があります。

1. 正しいタグとONTAPクラスタユーザ名を渡してONTAP Playbookを実行します。プロンプトが表示されたら、ONTAPクラスタのパスワードとvsadminを入力します。

```
ansible-playbook -i hosts all_playbook.yml -u username -k -K -t  
ontap_config -e @vars/vars.yml
```

2. Linux Playbookを実行して、導入のLinux部分を実行します。admin sshパスワードとsudoパスワードを入力します。

```
ansible-playbook -i hosts all_playbook.yml -u username -k -K -t  
linux_config -e @vars/vars.yml
```

3. Oracle Playbookを実行して、導入のOracle部分を実行します。admin sshパスワードとsudoパスワードを入力します。

```
ansible-playbook -i hosts all_playbook.yml -u username -k -K -t  
oracle_config -e @vars/vars.yml
```

同じ **Oracle** ホストに追加のデータベースを導入します

このプレイブックの Oracle 部分では、1 回の実行につき Oracle サーバ上に Oracle コンテナデータベースが 1 つ作成されます。同じサーバ上に追加のコンテナデータベースを作成するには、次の手順を実行します。

1. host_vars 変数を改訂します。
 - a. ステップ 3 に戻ります - 'host_vars' の下の 'host_name.yml' ファイルを編集します
 - b. Oracle SID を別の名前文字列に変更します。
 - c. リスナーポートを別の番号に変更します。
 - d. EM Express をインストールしている場合は、EM Express ポートを別の番号に変更します。
 - e. 変更したホスト変数を 'host_vars' の下の Oracle ホスト変数ファイルにコピーして貼り付けます
2. 上記のように 'ORACLE_CONFIG' タグを使用してプレイブックを実行します インチ [\[プレイブックを実行します\]](#)。

Oracle のインストールを検証します

1. OracleユーザとしてOracleサーバにログインし、次のコマンドを実行します。

```
ps -ef | grep ora
```



インストールが正常に完了した場合は、Oracle プロセスが一覧表示されます Oracle DB のサポートを開始しました

2. データベースにログインして、次のコマンドセットを使用して作成されたDB設定およびPDBを確認します。


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

SQL*Plus: Release 19.0.0.0.0 - Production on Thu May 6 12:52:51 2021
Version 19.8.0.0.0

Copyright (c) 1982, 2019, Oracle. All rights reserved.

Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.8.0.0.0

SQL>

SQL> select name, log_mode from v$database;
NAME          LOG_MODE
-----
CDB2          ARCHIVELOG

SQL> show pdbs

      CON_ID CON_NAME                                OPEN MODE  RESTRICTED
-----
          2 PDB$SEED                                READ ONLY   NO
          3 CDB2_PDB1                                READ WRITE  NO
          4 CDB2_PDB2                                READ WRITE  NO
          5 CDB2_PDB3                                READ WRITE  NO

col svrname form a30
col dirname form a30
select svrname, dirname, nfsversion from v$dnfs_servers;

SQL> col svrname form a30
SQL> col dirname form a30
SQL> select svrname, dirname, nfsversion from v$dnfs_servers;

SVRNAME                                DIRNAME                                NFSVERSION
-----
172.21.126.200                        /rhelora03_u02                        NFSv3.0
172.21.126.200                        /rhelora03_u03                        NFSv3.0
172.21.126.200                        /rhelora03_u01                        NFSv3.0
```

これにより、dNFSが正常に動作していることが確認されます。

3. 次のコマンドを使用して'リスナー経由でデータベースに接続し'Oracleリスナーの構成を確認します適切なリスナーポートとデータベースサービス名に変更します。

```
[oracle@localhost ~]$ sqlplus
system@//localhost:1523/cdb2_pdb1.cie.netapp.com

SQL*Plus: Release 19.0.0.0.0 - Production on Thu May 6 13:19:57 2021
Version 19.8.0.0.0

Copyright (c) 1982, 2019, Oracle. All rights reserved.

Enter password:
Last Successful login time: Wed May 05 2021 17:11:11 -04:00

Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.8.0.0.0

SQL> show user
USER is "SYSTEM"
SQL> show con_name
CON_NAME
CDB2_PDB1
```

これにより、Oracleリスナーが正常に動作していることが確認されます。

サポートが必要な場所

ツールキットに関するサポートが必要な場合は、[にご参加ください](#) ["ネットアップの解決策自動化コミュニティでは、余裕期間のチャンネルがサポートさ"](#) また、ソリューション自動化チャンネルを検索して、質問や問い合わせを投稿しましょう。

解決策の概要

このページでは、NetApp ONTAP ストレージ上に Oracle19c を導入するための自動化方式について説明します。

Oracle データベースのデータ保護を自動化

組織は環境を自動化して、効率を高め、導入を高速化し、手動作業を削減しています。Ansible などの構成管理ツールを使用して、エンタープライズデータベースの運用を合理化しています。この解決策では、Ansible を使用して NetApp ONTAP による Oracle のデータ保護を自動化する方法を紹介します。ストレージ管理者、システム管理者、DBA は、オフサイトのデータセンターやパブリッククラウドへのデータレプリケーションを一貫して迅速にセットアップできるため、次のようなメリットがあります。

- 設計の複雑さと人為的ミスを排除し、繰り返し実行可能な一貫した導入とベストプラクティスを実装する
- クラスタ間レプリケーション、CVO のインスタンス化、Oracle データベースのリカバリの構成にかかる時間を短縮できます

- データベース管理者、システム管理者、ストレージ管理者の生産性を向上
- データベースリカバリワークフローを使用して、DR シナリオを簡単にテストできます。

ネットアップは、検証済みの Ansible モジュールとロールをお客様に提供し、Oracle データベース環境の導入、構成、ライフサイクル管理を迅速化します。この解決策では、以下の作業に役立つ Ansible の Playbook コードを提供しています。

オンプレミスからオンプレミスへのレプリケーション

- ソースとデスティネーションにクラスタ間 LIF を作成
- クラスタと SVM のピア関係を確立
- Oracle ボリュームの SnapMirror を作成して初期化
- AWX/Tower を使用して、Oracle バイナリ、データベース、ログ用のレプリケーションスケジュールを作成します
- デスティネーションで Oracle DB のリストアを行い、データベースをオンラインにします

オンプレミスから **AWS** の **CVO** へ

- AWS コネクタを作成します
- AWS で CVO インスタンスを作成
- オンプレミスのクラスタを Cloud Manager に追加
- ソースにクラスタ間 LIF を作成
- クラスタと SVM のピア関係を確立
- Oracle ボリュームの SnapMirror を作成して初期化
- AWX/Tower を使用して、Oracle バイナリ、データベース、ログ用のレプリケーションスケジュールを作成します
- デスティネーションで Oracle DB のリストアを行い、データベースをオンラインにします

準備ができれば、をクリックします ["解決策の使用を開始するには、こちらをクリックしてください"](#)。

はじめに

この解決策は、AWX/Tower 環境で動作するように設計されています。

AWX / タワー

AWX / タワー環境の場合は、ONTAP クラスタ管理と Oracle サーバ（IP およびホスト名）のインベントリの作成、クレデンシャルの作成、NetApp Automation Github から Ansible コードを取得するプロジェクトの設定、および自動化を開始するジョブテンプレートの設定を案内されます。

1. 解決策は、プライベートクラウドのシナリオ（オンプレミスからオンプレミス）およびハイブリッドクラウド（オンプレミスからパブリッククラウドへの Cloud Volumes ONTAP [CVO]）で実行するように設計されています。
2. 環境に固有の変数を入力し、ジョブテンプレートのその他の VAR フィールドにコピーして貼り付けます。

3. ジョブテンプレートに変数を追加したら、自動化を起動できます。
4. 自動化は、Oracle バイナリのセットアップ、データベース、ログ、ログのレプリケーションスケジュール、ログのみのレプリケーションスケジュールの 3 つのフェーズと、DR サイトでのデータベースリカバリのための 4 つのフェーズで実行されます。
5. CVO Data Protection に必要なキーとトークンの取得方法の詳細については、を参照してください "[CVO の導入と Connector の導入の前提条件を収集](#)"

要件

<strong class="big"> オンプレミス |

環境	要件
* Ansible 環境 *	AWX / タワー
	Ansible v.2.10 以上
	Python 3.
	Python ライブラリ - NetApp-lib-xmltodict-jmespath
* ONTAP *	ONTAP バージョン 9.8+
	データアグリゲート × 2
	NFS VLAN および ifgrp が作成されました
* Oracle サーバ *	RHEL 7/8
	Oracle Linux 7/8
	NFS、パブリック、オプションの管理用のネットワークインターフェイス
	ソース上の既存の Oracle 環境と、デスティネーション（DR サイトまたはパブリッククラウド）上の同等の Linux オペレーティングシステム

 「ビッグ」 >CVO

環境	要件
* Ansible 環境 *	AWX / タワー
	Ansible v.2.10 以上
	Python 3.
	Python ライブラリ - NetApp-lib-xmltodict-jmespath
* ONTAP *	ONTAP バージョン 9.8+
	データアグリゲート × 2
	NFS VLAN および ifgrp が作成されました
* Oracle サーバ *	RHEL 7/8
	Oracle Linux 7/8
	NFS、パブリック、オプションの管理用のネットワークインターフェイス
	ソース上の既存の Oracle 環境と、デスティネーション（DR サイトまたはパブリッククラウド）上の同等の Linux オペレーティングシステム
	Oracle EC2 インスタンスに適切なスワップスペースを設定します。デフォルトでは、一部の EC2 インスタンスは 0 スワップで導入されます
* Cloud Manager / AWS *	AWS のアクセス / シークレットキー
	NetApp Cloud Manager アカウント
	NetApp Cloud Manager Refresh Token

<strong class="big"> オンプレミス |

この自動導入は、3つのロールで構成される Ansible プレイブックを使用して設計されています。ロールは ONTAP、Linux、Oracle の各構成に対応しています。次の表に、自動化されるタスクを示します。

Playbook	タスク
* ONTAP_setup*	ONTAP 環境の事前チェック
	ソースクラスタでのクラスタ間 LIF の作成（オプション）
	デスティネーションクラスタでのクラスタ間 LIF の作成（オプション）
	クラスタ / SVM ピアリングの作成
	SnapMirror デスティネーションの作成と、指定された Oracle ボリュームの初期化
* ora_replication_cg *	/etc/oratab 内の各データベースのバックアップモードを有効にします
	Oracle バイナリボリュームとデータベースボリュームの Snapshot
	SnapMirror を更新しました
	/etc/oratab 内の各データベースのバックアップモードをオフにします
* ora_replication_log *	/etc/oratab 内の各データベースの現在のログを切り替えます
	Oracle ログボリュームの Snapshot
	SnapMirror を更新しました
* ora_recovery*	SnapMirror を解除します
	デスティネーションで NFS を有効にし、Oracle ボリュームのジャンクションパスを作成します
	DR Oracle ホストを設定
	Oracle ボリュームをマウントして確認
	Oracle データベースをリカバリして起動します

 「ビッグ」 >CVO

この自動導入は、3つのロールで構成される Ansible プレイブックを使用して設計されています。ロールは ONTAP、Linux、Oracle の各構成に対応しています。次の表に、自動化されるタスクを示します。

Playbook	タスク
* CVF_setup*	環境の事前チェック
	AWS Configure / AWS Access Key ID / Secret Key / Default Region
	AWS ロールの作成
	AWS での NetApp Cloud Manager Connector インスタンスの作成
	AWS での Cloud Volumes ONTAP （CVO） インスタンスの作成
	オンプレミスのソース ONTAP クラスタを NetApp Cloud Manager に追加
	SnapMirror デスティネーションの作成と、指定された Oracle ボリュームの初期化
* ora_replication_cg *	/etc/oratab 内の各データベースのバックアップモードを有効にします
	Oracle バイナリボリュームとデータベースボリュームの Snapshot
	SnapMirror を更新しました
	/etc/oratab 内の各データベースのバックアップモードをオフにします
* ora_replication_log *	/etc/oratab 内の各データベースの現在のログを切り替えます
	Oracle ログボリュームの Snapshot
	SnapMirror を更新しました
* ora_recovery*	SnapMirror を解除します
	デスティネーション CVO で NFS を有効にし、Oracle ボリュームのジャンクションパスを作成してください
	DR Oracle ホストを設定
	Oracle ボリュームをマウントして確認
	Oracle データベースをリカバリして起動します

デフォルトパラメータ

自動化を簡易化するために、必要な Oracle パラメータがデフォルト値で多数設定されています。通常、ほとんどの環境でデフォルトパラメータを変更する必要はありません。上級ユーザーは 'デフォルト・パラメータ' を変更する際に注意してくださいデフォルトのパラメータは、各ロールフォルダの defaults ディレクトリにあります。

使用許諾

Github リポジトリに記載されているライセンス情報をお読みください。このリポジトリ内のコンテンツにアクセス、ダウンロード、インストール、または使用することにより、ライセンスの条項に同意したものとみなされます ["こちらをご覧ください"](#)。

このリポジトリ内のコンテンツの作成および / または派生著作物の共有に関しては、一定の制限事項があります。の条件を必ずお読みください ["使用許諾"](#) コンテンツを使用する前に。すべての条件に同意しない場合は、このリポジトリのコンテンツにアクセスしたり、コンテンツをダウンロードしたり、使用したりしないでください。

準備ができれば、をクリックします ["AWX/Tower の詳細な手順については、こちらを参照してください"](#)。

ステップバイステップの導入手順

このページでは、NetApp ONTAP ストレージ上の Oracle19c の自動データ保護について説明します。

AWX/Tower Oracle データ保護

環境のインベントリ、グループ、ホスト、クレデンシャルを作成します

このセクションでは、ネットアップの自動化ソリューションを使用する環境を準備するための AWX/Ansible タワーでのインベントリ、グループ、ホスト、アクセスクレデンシャルのセットアップについて説明します。

1. インベントリを設定します。
 - a. リソース→インベントリ→追加と進み、インベントリの追加をクリックします。
 - b. 名前と組織の詳細を入力し、[保存] をクリックします。
 - c. インベントリページで、作成されたインベントリをクリックします。
 - d. [グループ] サブメニューに移動し、[追加] をクリックします。
 - e. 最初のグループの Oracle という名前を入力し、[保存] をクリックします。
 - f. DR_Oracle という名前の 2 つ目のグループに対してこの手順を繰り返します。
 - g. 作成した Oracle グループを選択し、Hosts サブメニューに移動して、Add New Host をクリックします。
 - h. ソース Oracle ホストの管理 IP の IP アドレスを入力し、[保存] をクリックします。
 - i. DR_Oracle グループに対してこの手順を繰り返し、DR/Destination Oracle ホストの管理 IP / ホスト名を追加する必要があります。



以下は、オンプレミスと ONTAP、または AWS 上の CVO のクレデンシャルタイプとクレデンシャルを作成する手順です。

オンプレミス

1. クレデンシャルを設定します。
2. クレデンシャルタイプの作成ONTAP を使用するソリューションでは、ユーザ名とパスワードのエントリを照合するようにクレデンシャルタイプを設定する必要があります。
 - a. [管理] → [資格情報の種類] に移動し、[追加] をクリックします。
 - b. 名前と概要を指定します。
 - c. 入力構成に次の内容を貼り付けます。

```
fields:
  - id: dst_cluster_username
    type: string
    label: Destination Cluster Username
  - id: dst_cluster_password
    type: string
    label: Destination Cluster Password
    secret: true
  - id: src_cluster_username
    type: string
    label: Source Cluster Username
  - id: src_cluster_password
    type: string
    label: Source Cluster Password
    secret: true
```

- d. 次の内容をインジェクタ設定に貼り付け、[保存] をクリックします。

```
extra_vars:
  dst_cluster_username: '{{ dst_cluster_username }}'
  dst_cluster_password: '{{ dst_cluster_password }}'
  src_cluster_username: '{{ src_cluster_username }}'
  src_cluster_password: '{{ src_cluster_password }}'
```

3. ONTAP のクレデンシャルを作成します
 - a. [リソース] → [資格情報] に移動し、[追加] をクリックします。
 - b. ONTAP クレデンシャルの名前と組織の詳細を入力します
 - c. 前の手順で作成したクレデンシャルタイプを選択します。
 - d. タイプの詳細で、ソースクラスタとデスティネーションクラスタのユーザ名とパスワードを入力します。
 - e. [保存] をクリックします .
4. Oracle のクレデンシャルを作成します

- a. [リソース] → [資格情報] に移動し、[追加] をクリックします。
- b. Oracle の名前と組織の詳細を入力します。
- c. マシントラディショナルタイプを選択します。
- d. Type Details （タイプの詳細）に、Oracle ホストのユーザー名とパスワードを入力します。
- e. 適切な特権昇格方式を選択し、ユーザ名とパスワードを入力します。
- f. [保存] をクリックします。
- g. 必要に応じて、DR_Oracle ホストの別のトラディショナルに対して同じ手順を繰り返します。

CVO を確認して

1. クレデンシャルを設定します。
2. クレデンシャルタイプを作成する。ONTAP が関連するソリューションでは、ユーザ名とパスワードのエントリに一致するトラディショナルタイプを設定する必要があります。また、Cloud Central と AWS のエントリも追加します。
 - a. [管理] → [資格情報の種類] に移動し、[追加] をクリックします。
 - b. 名前と概要を指定します。
 - c. 入力構成に次の内容を貼り付けます。

```
fields:
  - id: dst_cluster_username
    type: string
    label: CVO Username
  - id: dst_cluster_password
    type: string
    label: CVO Password
    secret: true
  - id: cvo_svm_password
    type: string
    label: CVO SVM Password
    secret: true
  - id: src_cluster_username
    type: string
    label: Source Cluster Username
  - id: src_cluster_password
    type: string
    label: Source Cluster Password
    secret: true
  - id: regular_id
    type: string
    label: Cloud Central ID
    secret: true
  - id: email_id
    type: string
    label: Cloud Manager Email
    secret: true
  - id: cm_password
    type: string
    label: Cloud Manager Password
    secret: true
  - id: access_key
    type: string
    label: AWS Access Key
    secret: true
  - id: secret_key
    type: string
    label: AWS Secret Key
    secret: true
  - id: token
    type: string
    label: Cloud Central Refresh Token
    secret: true
```

d. 次の内容をインジェクタ構成に貼り付け、[保存 (Save)] をクリックする。

```
extra_vars:
  dst_cluster_username: '{{ dst_cluster_username }}'
  dst_cluster_password: '{{ dst_cluster_password }}'
  cvo_svm_password: '{{ cvo_svm_password }}'
  src_cluster_username: '{{ src_cluster_username }}'
  src_cluster_password: '{{ src_cluster_password }}'
  regular_id: '{{ regular_id }}'
  email_id: '{{ email_id }}'
  cm_password: '{{ cm_password }}'
  access_key: '{{ access_key }}'
  secret_key: '{{ secret_key }}'
  token: '{{ token }}'
```

3. ONTAP / CVO / AWS のクレデンシャルを作成

- a. [リソース] → [資格情報] に移動し、[追加] をクリックします。
- b. ONTAP クレデンシャルの名前と組織の詳細を入力します
- c. 前の手順で作成したクレデンシャルタイプを選択します。
- d. Type Details に、ソースクラスタと CVO クラスタ、Cloud Central / Manager 、AWS Access / Secret Key 、Cloud Central Refresh Token のユーザ名とパスワードを入力します。
- e. [保存] をクリックします。

4. Oracle のクレデンシャルの作成（ソース）

- a. [リソース] → [資格情報] に移動し、[追加] をクリックします。
- b. Oracle ホストの名前と組織の詳細を入力します
- c. マシンのクレデンシャルタイプを選択します。
- d. Type Details （タイプの詳細）に、Oracle ホストのユーザー名とパスワードを入力します。
- e. 適切な特権昇格方式を選択し、ユーザ名とパスワードを入力します。
- f. [保存] をクリックします。

5. Oracle 保存先のクレデンシャルを作成します

- a. [リソース] → [資格情報] に移動し、[追加] をクリックします。
- b. DR Oracle ホストの名前と組織の詳細を入力します
- c. マシンのクレデンシャルタイプを選択します。
- d. Type Details に、ユーザ名（ec2-user またはデフォルトの入力から変更した場合は、そのユーザ名）と SSH 秘密鍵を入力します
- e. 適切な特権昇格方式（sudo）を選択し、必要に応じてユーザ名とパスワードを入力します。
- f. [保存] をクリックします。

プロジェクトを作成します

1. [リソース] → [プロジェクト] に移動し、[追加] をクリックします。
 - a. 名前と組織の詳細を入力します
 - b. Source Control Credential Type フィールドで Git を選択します。
 - c. 入力するコマンド <https://github.com/NetApp-Automation/na_oracle19c_data_protection.git> をソース管理 URL として指定します。
 - d. [保存] をクリックします。
 - e. ソースコードが変更されたときに、プロジェクトの同期が必要になることがあります。

グローバル変数を設定します

このセクションで定義する変数は、すべての Oracle ホスト、データベース、および ONTAP クラスタに適用されます。

1. 次の組み込みグローバル変数または変数フォームに環境固有のパラメータを入力します。



青の項目は、環境に合わせて変更する必要があります。

オンプレミス

```
# Oracle Data Protection global user configuration variables
# Ontap env specific config variables
hosts_group: "ontap"
ca_signed_certs: "false"

# Inter-cluster LIF details
src_nodes:
  - "AFF-01"
  - "AFF-02"

dst_nodes:
  - "DR-AFF-01"
  - "DR-AFF-02"

create_source_intercluster_lifs: "yes"

source_intercluster_network_port_details:
  using_dedicated_ports: "yes"
  using_ifgrp: "yes"
  using_vlans: "yes"
  failover_for_shared_individual_ports: "yes"
  ifgrp_name: "a0a"
  vlan_id: "10"
  ports:
    - "e0b"
    - "e0g"
  broadcast_domain: "NFS"
  ipspace: "Default"
  failover_group_name: "iclifs"

source_intercluster_lif_details:
  - name: "icl_1"
    address: "10.0.0.1"
    netmask: "255.255.255.0"
    home_port: "a0a-10"
    node: "AFF-01"
  - name: "icl_2"
    address: "10.0.0.2"
    netmask: "255.255.255.0"
    home_port: "a0a-10"
    node: "AFF-02"

create_destination_intercluster_lifs: "yes"
```

```

destination_intercluster_network_port_details:
  using_dedicated_ports: "yes"
  using_ifgrp: "yes"
  using_vlans: "yes"
  failover_for_shared_individual_ports: "yes"
  ifgrp_name: "a0a"
  vlan_id: "10"
  ports:
    - "e0b"
    - "e0g"
  broadcast_domain: "NFS"
  ipspace: "Default"
  failover_group_name: "iclifs"

destination_intercluster_lif_details:
  - name: "icl_1"
    address: "10.0.0.3"
    netmask: "255.255.255.0"
    home_port: "a0a-10"
    node: "DR-AFF-01"
  - name: "icl_2"
    address: "10.0.0.4"
    netmask: "255.255.255.0"
    home_port: "a0a-10"
    node: "DR-AFF-02"

# Variables for SnapMirror Peering
passphrase: "your-passphrase"

# Source & Destination List
dst_cluster_name: "dst-cluster-name"
dst_cluster_ip: "dst-cluster-ip"
dst_vserver: "dst-vserver"
dst_nfs_lif: "dst-nfs-lif"
src_cluster_name: "src-cluster-name"
src_cluster_ip: "src-cluster-ip"
src_vserver: "src-vserver"

# Variable for Oracle Volumes and SnapMirror Details
cg_snapshot_name_prefix: "oracle"
src_orabinary_vols:
  - "binary_vol"
src_db_vols:
  - "db_vol"
src_archivelog_vols:
  - "log_vol"

```



```

snapmirror_policy: "async_policy_oracle"

# Export Policy Details
export_policy_details:
  name: "nfs_export_policy"
  client_match: "0.0.0.0/0"
  ro_rule: "sys"
  rw_rule: "sys"

# Linux env specific config variables
mount_points:
  - "/u01"
  - "/u02"
  - "/u03"
hugepages_nr: "1234"
redhat_sub_username: "xxx"
redhat_sub_password: "xxx"

# DB env specific install and config variables
recovery_type: "scn"
control_files:
  - "/u02/oradata/CDB2/control01.ctl"
  - "/u03/orareco/CDB2/control02.ctl"

```

CVOを確認して

```

#####
### Ontap env specific config variables ###
#####

#Inventory group name
#Default inventory group name - "ontap"
#Change only if you are changing the group name either in
inventory/hosts file or in inventory groups in case of AWX/Tower
hosts_group: "ontap"

#CA signed certificates (ONLY CHANGE to "true" IF YOU ARE USING CA
SIGNED CERTIFICATES)
ca_signed_certs: "false"

#Names of the Nodes in the Source ONTAP Cluster
src_nodes:
  - "AFF-01"
  - "AFF-02"

#Names of the Nodes in the Destination CVO Cluster

```

```

dst_nodes:
  - "DR-AFF-01"
  - "DR-AFF-02"

#Define whether or not to create intercluster lifs on source cluster
(ONLY CHANGE to "No" IF YOU HAVE ALREADY CREATED THE INTERCLUSTER LIFS)
create_source_intercluster_lifs: "yes"

source_intercluster_network_port_details:
  using_dedicated_ports: "yes"
  using_ifgrp: "yes"
  using_vlans: "yes"
  failover_for_shared_individual_ports: "yes"
  ifgrp_name: "a0a"
  vlan_id: "10"
  ports:
    - "e0b"
    - "e0g"
  broadcast_domain: "NFS"
  ipspace: "Default"
  failover_group_name: "iclifs"

source_intercluster_lif_details:
  - name: "icl_1"
    address: "10.0.0.1"
    netmask: "255.255.255.0"
    home_port: "a0a-10"
    node: "AFF-01"
  - name: "icl_2"
    address: "10.0.0.2"
    netmask: "255.255.255.0"
    home_port: "a0a-10"
    node: "AFF-02"

#####
### CVO Deployment Variables ###
#####

##### Access Keys Variables #####

# Region where your CVO will be deployed.
region_deploy: "us-east-1"

##### CVO and Connector Vars #####

# AWS Managed Policy required to give permission for IAM role creation.

```

```

aws_policy: "arn:aws:iam::1234567:policy/OCCM"

# Specify your aws role name, a new role is created if one already does
not exist.
aws_role_name: "arn:aws:iam::1234567:policy/OCCM"

# Name your connector.
connector_name: "awx_connector"

# Name of the key pair generated in AWS.
key_pair: "key_pair"

# Name of the Subnet that has the range of IP addresses in your VPC.
subnet: "subnet-12345"

# ID of your AWS security group that allows access to on-prem
resources.
security_group: "sg-123123123"

# Your Cloud Manager Account ID.
account: "account-A23123A"

# Name of the your CVO instance
cvo_name: "test_cvo"

# ID of the VPC in AWS.
vpc: "vpc-123123123"

#####
#####
# Variables for - Add on-prem ONTAP to Connector in Cloud Manager
#####
#####

# For Federated users, Client ID from API Authentication Section of
Cloud Central to generate access token.
sso_id: "123123123123123123123"

# For regular access with username and password, please specify "pass"
as the connector_access. For SSO users, use "refresh_token" as the
variable.
connector_access: "pass"

#####
#####
# Variables for SnapMirror Peering
#####

```

```
#####
passphrase: "your-passphrase"

#####
#####
# Source & Destination List
#####
#####
#Please Enter Destination Cluster Name
dst_cluster_name: "dst-cluster-name"

#Please Enter Destination Cluster (Once CVO is Created Add this
Variable to all templates)
dst_cluster_ip: "dst-cluster-ip"

#Please Enter Destination SVM to create mirror relationship
dst_vserver: "dst-vserver"

#Please Enter NFS Lif for dst vserver (Once CVO is Created Add this
Variable to all templates)
dst_nfs_lif: "dst-nfs-lif"

#Please Enter Source Cluster Name
src_cluster_name: "src-cluster-name"

#Please Enter Source Cluster
src_cluster_ip: "src-cluster-ip"

#Please Enter Source SVM
src_vserver: "src-vserver"

#####
#####
# Variable for Oracle Volumes and SnapMirror Details
#####
#####
#Please Enter Source Snapshot Prefix Name
cg_snapshot_name_prefix: "oracle"

#Please Enter Source Oracle Binary Volume(s)
src_orabinary_vols:
- "binary_vol"
#Please Enter Source Database Volume(s)
src_db_vols:
- "db_vol"
#Please Enter Source Archive Volume(s)
```

```

src_archivelog_vols:
  - "log_vol"
#Please Enter Destination Snapmirror Policy
snapmirror_policy: "async_policy_oracle"

#####
#####
# Export Policy Details
#####
#####
#Enter the destination export policy details (Once CVO is Created Add
this Variable to all templates)
export_policy_details:
  name: "nfs_export_policy"
  client_match: "0.0.0.0/0"
  ro_rule: "sys"
  rw_rule: "sys"

#####
#####
### Linux env specific config variables ###
#####
#####

#NFS Mount points for Oracle DB volumes
mount_points:
  - "/u01"
  - "/u02"
  - "/u03"

# Up to 75% of node memory size divided by 2mb. Consider how many
databases to be hosted on the node and how much ram to be allocated to
each DB.
# Leave it blank if hugepage is not configured on the host.
hugepages_nr: "1234"

# RedHat subscription username and password
redhat_sub_username: "xxx"
redhat_sub_password: "xxx"

#####
### DB env specific install and config variables ###
#####
#Recovery Type (leave as scn)
recovery_type: "scn"

```

```
#Oracle Control Files
control_files:
  - "/u02/oradata/CDB2/control01.ctl"
  - "/u03/orareco/CDB2/control02.ctl"
```

自動化ハンドブック

実行する必要があるプレイブックは4つあります。

1. 環境のセットアップに関するプレイブック：オンプレミス、CVO
2. Oracle バイナリとデータベースをスケジュールどおりにレプリケートする Playbook
3. Oracle ログをスケジュールどおりにレプリケートするためのプレイブック
4. デスティネーションホストでのデータベースのリカバリに関するプレイブック

ONTAP/CVO セットアップ

ONTAP と CVO のセットアップ

ジョブテンプレートを設定して起動します。

1. ジョブテンプレートを作成します。
 - a. [リソース]→[テンプレート]→[追加]に移動し、[ジョブテンプレートの追加]をクリックします。
 - b. 「ONTAP/CVO Setup」という名前を入力します
 - c. ジョブタイプを選択します。Run は、プレイブックに基づいてシステムを設定します。
 - d. 対応するインベントリ、プロジェクト、プレイブック、およびクレデンシャルを選択します。
 - e. オンプレミス環境用の ONTAP_setup.yml プレイブックを選択するか、CVO-setup.yml を選択して CVO インスタンスにレプリケーションします。
 - f. 手順 4 からコピーしたグローバル変数を YAML タブの Template Variables フィールドに貼り付けます。
 - g. [保存]をクリックします。
2. ジョブテンプレートを起動します。
 - a. [リソース]→[テンプレート]に移動します。
 - b. 目的のテンプレートをクリックし、[起動]をクリックします。



このテンプレートを使用して、他のプレイブック用にコピーします。

バイナリおよびデータベースボリュームのレプリケーション

バイナリおよびデータベースのレプリケーションマニュアルのスケジュール

ジョブテンプレートを設定して起動します。

1. 以前に作成したジョブテンプレートをコピーします。
 - a. [リソース]→[テンプレート]に移動します。
 - b. 「ONTAP/CVO Setup Template」を探して、右端で「Copy Template」をクリックします
 - c. コピーしたテンプレートで[テンプレートの編集]をクリックし、名前を[バイナリおよびデータベースのレプリケーションのマニュアル]に変更します。
 - d. テンプレートの同じインベントリ、プロジェクト、資格情報を保持します。
 - e. 実行するプレイブックとして ora_replication_cg.yml を選択します。
 - f. 変数は変更されませんが、CVO クラスターの IP は変数 dst_cluster_ip に設定する必要があります。
 - g. [保存]をクリックします。
2. ジョブテンプレートをスケジュールします。
 - a. [リソース]→[テンプレート]に移動します。

- b. バイナリおよびデータベースのレプリケーション用プレイブックテンプレートをクリックし、一番上のオプションセットにあるスケジュールをクリックします。
- c. [追加] をクリックし、[バイナリおよびデータベースレプリケーションの名前スケジュールの追加] をクリックし、時間の開始時に [開始日時] を選択し、[ローカルタイムゾーン] を選択して、[実行頻度] をクリックします。実行頻度は、多くの場合、SnapMirror レプリケーションが更新されます。



ログボリュームのレプリケーション用に別のスケジュールが作成されるため、より頻繁にレプリケートできます。

ログボリュームのレプリケーション

ログ・レプリケーション・プレイブックのスケジュール

ジョブテンプレートを設定して起動します。

1. 以前に作成したジョブテンプレートをコピーします。
 - a. [リソース] → [テンプレート] に移動します。
 - b. 「ONTAP/CVO Setup Template」を探して、右端で「Copy Template」をクリックします
 - c. コピーしたテンプレートで [テンプレートの編集] をクリックし、名前を [ログレプリケーションのプレイブック] に変更します。
 - d. テンプレートの同じインベントリ、プロジェクト、資格情報を保持します。
 - e. 実行するプレイブックとして ora_replication_loges.yml を選択します。
 - f. 変数は変更されませんが、CVO クラスターの IP は変数 dst_cluster_ip に設定する必要があります。
 - g. [保存] をクリックします。
2. ジョブテンプレートをスケジュールします。
 - a. [リソース] → [テンプレート] に移動します。
 - b. Log Replication Playbook テンプレートをクリックし、一番上のオプションセットにある Schedules (スケジュール) をクリックします。
 - c. [追加] をクリックし、[ログ複製の名前スケジュールの追加] をクリックし、時間の開始時に開始日時を選択し、[ローカルタイムゾーン] と [実行頻度] を選択します。実行頻度は、多くの場合、SnapMirror レプリケーションが更新されます。



1 時間ごとの最新の更新に確実にリカバリできるように、ログスケジュールを 1 時間ごとに更新するように設定することを推奨します。

データベースのリストアとリカバリ

ログ・レプリケーション・プレイブックのスケジュール

ジョブテンプレートを設定して起動します。

1. 以前に作成したジョブテンプレートをコピーします。
 - a. [リソース] → [テンプレート] に移動します。

- b. 「ONTAP/CVO Setup Template」を探して、右端で「Copy Template」をクリックします
- c. コピーしたテンプレートで [テンプレートの編集] をクリックし、名前を [リストアとリカバリプレイブック] に変更します。
- d. テンプレートの同じインベントリ、プロジェクト、資格情報を保持します。
- e. 実行するプレイブックとして ora_recovery.yml を選択します。
- f. 変数は変更されませんが、CVO クラスターの IP は変数 dst_cluster_ip に設定する必要があります。
- g. [保存] をクリックします。



このプレイブックは、リモートサイトでデータベースをリストアする準備ができるまでは実行されません。

Oracle データベースをリカバリしています

1. オンプレミスの本番 Oracle データベースのデータボリュームは、NetApp SnapMirror レプリケーションを使用して、セカンダリデータセンターの冗長 ONTAP クラスターまたはパブリッククラウドの Cloud Volume ONTAP に保護されます。完全に構成されたディザスタリカバリ環境では、セカンダリデータセンターまたはパブリッククラウドのリカバリコンピューティングインスタンスがスタンバイ状態になり、災害発生時に本番データベースをリカバリできます。スタンバイコンピューティングインスタンスは、OS カーネルパッチで parallel アップデートを実行するか、ロックステップでアップグレードすることで、オンプレミスインスタンスと同期したままになります。
2. この解決策で実証されている Oracle バイナリ・ボリュームは、ターゲット・インスタンスに複製され、ターゲット・インスタンスにマウントされて、Oracle ソフトウェア・スタックが起動されます。この Oracle リカバリアプローチには、災害発生時に Oracle を新規にインストールした場合よりも優れています。Oracle のインストールは、現在のオンプレミスの本番ソフトウェアのインストールレベルやパッチレベルと完全に同期されていることが保証されます。ただし、Oracle でのソフトウェアライセンスの構成によっては、リカバリサイトで複製された Oracle バイナリボリュームにソフトウェアライセンスが影響する場合とそうでない場合があります。ユーザは、Oracle のライセンス要件を評価するために、ソフトウェアライセンス担当者に確認してから、同じ方法を使用することを推奨します。
3. デスティネーションのスタンバイ Oracle ホストには、Oracle の前提条件となる構成が設定されています。
4. SnapMirror が切断され、ボリュームが書き込み可能になり、スタンバイ Oracle ホストにマウントされます。
5. すべての DB ボリュームがスタンバイコンピューティングインスタンスにマウントされたあと、Oracle リカバリモジュールは以下のタスクを実行して、リカバリサイトで Oracle をリカバリおよび起動します。
 - a. 制御ファイルを同期します。重要なデータベース制御ファイルを保護するために、異なるデータベースボリュームに Oracle 制御ファイルを重複して配置しました。1 つはデータボリューム上にあり、もう 1 つはログボリューム上にあります。データボリュームとログボリュームは異なる頻度でレプリケートされるため、リカバリ時に同期されません。
 - b. Oracle バイナリの再リンク：Oracle バイナリは新しいホストに再配置されるため、再リンクが必要です。
 - c. Oracle データベースのリカバリ：リカバリ・メカニズムは、Oracle ログ・ボリューム内の最後に使用可能なアーカイブ・ログのシステム変更番号を制御ファイルから取得し、Oracle データベースをリカバリして、障害発生時に DR サイトにレプリケートされたすべてのビジネス・トランザクションをリカバリします。次に、データベースが新しいインカンセーションで起動され、リカバリサイトでユー

ザ接続とビジネストランザクションが実行されます。



Recovering Playbook を実行する前に、次の情報を確認してください。 /etc/oratab および /etc/orainst.loc を介して、ソース Oracle ホストからデスティネーションホストにコピーしてください

TR-4794 : 『Oracle databases on NetApp EF Series』

Mitch Blackburn、Ebin Kadavy、ネットアップ

TR-4794は、ストレージ管理者とデータベース管理者がOracleをNetApp EFシリーズのストレージに正常に導入できるようにするためのものです。

"TR-4794 : 『Oracle databases on NetApp EF Series』 "

著作権に関する情報

Copyright © 2024 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。