NFS上のONTAP向けOracle 19cの自動導入

NetApp database solutions

NetApp
August 18, 2025

This PDF was generated from https://docs.netapp.com/ja-jp/netapp-solutions-databases/automation/marketing-overview.html on August 18, 2025. Always check docs.netapp.com for the latest.

目次

NFS上のONTAP向けOracle 19cの自動導入 . . . 1
ソリューションの概要 . . . 1
NFS上のONTAP向けOracle19cの自動導入 . . . 1
開始 . . . 2
AWX/タワー . . . 2
Ansible コントロールホスト経由の CLI . . . 2
要件 . . . 2
自動化の詳細 . . . 3
デフォルトパラメータ . . . 4
展開手順 . . . 4
ライセンス . . . 4
ステップバイステップの展開手順 . . . 5
AWX/Tower 導入 Oracle 19c データベース . . . 5
ステップバイステップの展開手順 . . . 14
CLI デプロイメント Oracle 19c データベース . . . 14

NFS上のONTAP向けOracle 19cの自動導入

ソリューションの概要

このページでは、 NetApp ONTAPストレージに Oracle19c をデプロイするための自動化された方法について説明します。

NFS上のONTAP向けOracle19cの自動導入

組織は、効率性を高め、導入を加速し、手作業の労力を削減するために環境を自動化しています。 Ansible などの構成管理ツールは、エンタープライズ データベースの運用を効率化するために使用されています。このソリューションでは、Ansible を使用して、 NetApp ONTAPを使用した Oracle 19c のプロビジョニングと構成を自動化する方法を示します。ストレージ管理者、システム管理者、DBA が新しいストレージを一貫して迅速に導入し、データベース サーバーを構成し、Oracle 19c ソフトウェアをインストールできるようにすることで、次の利点が得られます。

- 設計の複雑さと人的エラーを排除し、繰り返し可能な一貫した展開とベストプラクティスを実装します。
- 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の拡張を容易に実現

NetApp は、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するために、検証済みの Ansible モジュールとロールを顧客に提供します。このソリューションは、次のことを支援する手順と Ansible プレイブック コードを提供します。

- Oracle データベース用のONTAP NFS ストレージの作成と構成
- RedHat Enterprise Linux 7/8 または Oracle Linux 7/8 に Oracle 19c をインストールする
- ONTAP NFSストレージでOracle 19cを構成する

詳細または開始するには、以下の概要ビデオをご覧ください。

AWX/タワーの展開

パート1：はじめに、要件、自動化の詳細、AWX/Towerの初期構成

AWX 展開

パート2: 変数とプレイブックの実行

AWX プレイブック実行

CLI デプロイメント

パート 1: はじめに、要件、自動化の詳細、Ansible コントロール ホストのセットアップ

CLI デプロイメント

CLIプレイブック実行

開始

このソリューションは、AWX/Tower 環境または Ansible 制御ホスト上の CLI で実行されるように設計されています。

AWX/タワー

AWX/Tower 環境では、 ONTAPクラスタ管理と Oracle サーバ (IP とホスト名) のインベントリの作成、資格情報の作成、 NetApp Automation Github から Ansible コードを取得するプロジェクトの構成、自動化を起動するジョブ テンプレートの設定についてガイドされます。

1. 環境に固有の変数を入力し、それをコピーしてジョブ テンプレートの「Extra Vars」フィールドに貼り付けます。
2. 追加の変数がジョブ テンプレートに追加されたら、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、oracle_config のタグを指定して 3 つのフェーズで実行されます。

Ansible コントロールホスト経由の CLI

1. LinuxホストをAnsible制御ホストとして使用できるように構成する"詳しい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 リポジトリを git clone できます。
3. ONTAPクラスタ管理および Oracle サーバーの管理 IP の IP および/またはホスト名を使用して、hosts ファイルを編集します。
4. 環境固有の変数を入力し、コピーして貼り付けます。 `vars.yml`ファイル。
5. 各 Oracle ホストには、ホスト名で識別される変数ファイルがあり、その中にホスト固有の変数が含まれています。
6. すべての変数ファイルが完成したら、タグを指定してプレイブックを3つのフェーズで実行できます。 `ontap_config` 、 `linux_config` 、 そして `oracle_config`。

要件

環境	要件
Ansible環境	Ansible 制御ホストとなる AWX/Tower または Linux ホスト
	Ansible v.2.10以降
	Python 3
	Python ライブラリ - netapp-lib - xmltodict - jmespath

環境	要件
ONTAP	ONTAPバージョン 9.3 - 9.7
	2つのデータ集約
	NFS VLANとifgrpが作成されました
Oracle サーバー	RHEL 7/8
	Oracle Linux 7/8
	NFS、パブリック、およびオプションの管理用のネットワーク インターフェース
	Oracle サーバー上の Oracle インストール ファイル

自動化の詳細

この自動デプロイメントは、3 つの個別のロールで構成される単一の 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、および Oracle 構成用です。次の表は、自動化されるタスクを示しています。

ロール	タスク
オンタップ構成	ONTAP環境の事前チェック
	Oracle用NFSベースのSVMの作成
	輸出政策の策定
	Oracle用ボリュームの作成
	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する
	NFSマウントを確認する
	OS固有の構成
	Oracleディレクトリを作成する
	ヒュージページを設定する
	SELinuxとファイアウォールデーモンを無効にする
	chronyd サービスを有効にして起動する
	ファイル記述子のハード制限を増やす
	pam.dセッションファイルを作成する

ロール	タスク
oracle_config	Oracleソフトウェアのインストール
	Oracleリスナーを作成する
	Oracleデータベースを作成する
	Oracle環境の構成
	PDBの状態を保存
	インスタンスアーカイブモードを有効にする
	DNFSクライアントを有効にする
	OS の再起動間でデータベースの自動起動とシャットダウンを有効にする

デフォルトパラメータ

自動化を簡素化するために、多くの必須 Oracle デプロイメント パラメータをデフォルト値で事前に設定してあります。通常、ほとんどの展開ではデフォルトのパラメータを変更する必要はありません。より上級のユーザーは、注意しながらデフォルトのパラメータを変更できます。デフォルトのパラメータは、defaults ディレクトリの下の各ロール フォルダにあります。

展開手順

始める前に、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`デプロイされる各 DB サーバー上のすべてのユーザーが読み取り、書き込み、実行にアクセスできるディレクトリ。自動化タスクは、Oracle のインストールと構成のために、特定のディレクトリ内で指定されたインストール ファイルを検索します。

```
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
```

ライセンス

Github リポジトリに記載されているライセンス情報を読む必要があります。このリポジトリ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、あなたはここに記載されたライセンスの条件に同意したことになります。"ここをクリックしてください。" 。

このリポジトリのコンテンツを使用した派生作品の作成および/または共有には一定の制限があることに注意してください。利用規約を必ずお読みください"ライセンス"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、ダウンロードしたり、使用したりしないでください。

準備ができたらクリックします"AWX/Towerの詳細な展開手順については、こちらをご覧ください。"または"CLIデプロイメントはこちら"。

ステップバイステップの展開手順

このページでは、 NetApp ONTAPストレージに Oracle19c をデプロイするための自動化された方法について説明します。

AWX/Tower 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、資格情報を作成します

このセクションでは、 NetApp自動化ソリューションを使用するための環境を準備する、AWX/Ansible Tower でのインベントリ、グループ、ホスト、およびアクセス資格情報の設定について説明します。

1. インベントリを構成します。
    a. 「リソース」→「インベントリ」→「追加」に移動し、「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、作成したインベントリをクリックします。
    d. 在庫変数がある場合は、変数フィールドに貼り付けます。
    e. [グループ] サブメニューに移動し、[追加] をクリックします。
    f. 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。
    g. Oracle の別のグループに対してこのプロセスを繰り返します。
    h. 作成したONTAPグループを選択し、[ホスト] サブメニューに移動して [新しいホストの追加] をクリックします。
    i. ONTAPクラスタ管理 IP の IP アドレスを入力し、ホスト変数（ある場合）を貼り付けて、[保存] をクリックします。
    j. 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. 資格情報の種類を作成します。ONTAPに関連するソリューションの場合、ユーザー名とパスワードのエントリと一致するように認証情報の種類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. 名前と説明を入力します。
    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 Injector Configuration に貼り付けます。

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

1. 資格情報を設定します。
   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認証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、vsadmin_password 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. マシン資格情報タイプを選択します。
   h. 「タイプの詳細」で、Oracle ホストのユーザー名とパスワードを入力します。
   i. 正しい権限昇格方法を選択し、ユーザー名とパスワードを入力します。

2.プロジェクトを作成する

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. ソース管理資格情報タイプ フィールドで Git を選択します。
   c. 入力 `\https://github.com/NetApp-Automation/na_oracle19c_deploy.git`ソース管理 URL として。
   d. [Save]をクリックします。
   e. ソース コードが変更された場合、プロジェクトを時々同期する必要がある場合があります。

3.Oracle の host_vars を構成する

このセクションで定義された変数は、個々の Oracle サーバーおよびデータベースに適用されます。

1. 次の埋め込み 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。

青色の項目は、環境に合わせて変更する必要があります。

ホスト**VARS**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. 青いフィールドにすべての変数を入力します。
2. 変数の入力が完了したら、フォームの「コピー」ボタンをクリックして、AWX または Tower に転送するすべての変数をコピーします。
3. AWX または Tower に戻り、「リソース」→「ホスト」に移動して、Oracle サーバー構成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、手順 1 からコピーした変数を [YAML] タブの [変数] フィールドに貼り付けます。
5. [Save]をクリックします。
6. システム内の追加の Oracle サーバーに対してこのプロセスを繰り返します。

4.グローバル変数を設定する

このセクションで定義されている変数は、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. 次の埋め込みグローバル変数または vars フォームに環境固有のパラメータを入力します。

> ⓘ 青色の項目は、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 alternatingly
```

```
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. 青いフィールドにすべての変数を入力します。
2. 変数の入力が完了したら、フォームの [コピー] ボタンをクリックして、AWX または Tower に転送す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 テンプレートを構成して起動します。

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. 名前と説明を入力してください
   c. ジョブ タイプを選択します。実行ではプレイブックに基づいてシステムが構成され、チェックでは実際にシステムを構成せずにプレイブックのドライ ランが実行されます。
   d. プレイブックに対応するインベントリ、プロジェクト、プレイブック、および資格情報を選択します。
   e. 実行するデフォルトのプレイブックとして all_playbook.yml を選択します。
   f. 手順 4 でコピーしたグローバル変数を、YAML タブのテンプレート変数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] ボックスをオンにします。
   h. [Save]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. 起動時にジョブ タグの入力を求められた場合は、requirements_config と入力します。ジョブ タグを入力するには、requirements_config の下の [ジョブ タグの作成] 行をクリックする必要がある場合があります。

> requirements_config は、他のロールを実行するための正しいライブラリがあることを確認します。

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. 起動時にジョブ タグの入力を求められた場合は、ontap_config と入力します。ジョブ タグを入力するに

は、ontap_config のすぐ下にある「ジョブ タグの作成」行をクリックする必要がある場合があります。

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。
6. ontap_config ロールが完了したら、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. 希望するテンプレートを選択し、「起動」をクリックします。
9. 起動時に linux_config でジョブ タグを入力するように求められたら、ジョブ タグを入力するために linux_config のすぐ下の「ジョブ タグの作成」行を選択する必要がある場合があります。
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. 希望するテンプレートを選択し、「起動」をクリックします。
15. 起動時にジョブ タグの入力を求められた場合は、「oracle_config」と入力します。ジョブ タグを入力するには、oracle_config のすぐ下の「ジョブ タグの作成」行を選択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導入する

プレイブックの Oracle 部分では、実行ごとに Oracle サーバー上に単一の Oracle コンテナ データベースが作成されます。同じサーバー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 変数を修正します。
   a. 手順 2 に戻り、Oracle host_vars を構成します。
   b. Oracle SID を別の命名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. 修正したホスト変数をコピーして、「ホスト構成の詳細」タブの「Oracle ホスト変数」フィールドに貼り付けます。
2. oracle_config タグのみを使用してデプロイメント ジョブ テンプレートを起動します。
3. Oracle ユーザーとして Oracle サーバーにログインし、次のコマンドを実行します。

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

> インストールが期待通りに完了し、Oracle DBが起動した場合、Oracleプロセスが一覧表示されます。

4. データベースにログインして、db 構成設定と次のコマンド セットで作成された 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。

どこに助けを求めたらいいですか**?**

ツールキットに関するサポートが必要な場合は、"NetAppソリューション オートメーション コミュニティ サポート Slack チャンネル"質問や問い合わせを投稿するには、ソリューション自動化チャネルを探してください。

ステップバイステップの展開手順

このドキュメントでは、自動化コマンド 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 デプロイメント Oracle 19c データベース

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デプロイするために必要な手順について説明します。必ず確認し、"開始と要件のセクション"それに応じて環境を準備しました。

Oracle19cリポジトリをダウンロード

1. Ansible コントローラーから、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後、ディレクトリを na_oracle19c_deploy <cd na_oracle19c_deploy> に変更します。

ホストファイルを編集する

展開前に次の作業を完了してください。

1. ホスト ファイル na_oracle19c_deploy ディレクトリを編集します。
2. [ontap] の下で、IP アドレスをクラスター管理 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。ホスト名は、DNS またはホスト ファイルを通じて IP アドレスに解決されるか、ホストで指定される必要があります。
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、プレイブックを実行し、2 つの Oracle DB サーバーに同時に Oracle 19c をデプロイします。 1 つの DB サーバーだけでテストすることもできます。その場合、ホスト変数ファイルを 1 つだけ構成する必要があります。

> プレイブックは、デプロイする Oracle ホストとデータベースの数に関係なく、同じように実行されます。

host_varsの下の**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。編集してコピーする `host_vars`ホストVARS設定セクションからコピーして、希望する場所に貼り付けます。 `host_name.yml`ファイル。

> 青色の項目は、環境に合わせて変更する必要があります。

ホストVARS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"}
```

vars.ymlファイルを編集する

その `vars.yml`ファイルは、Oracle デプロイメントのすべての環境固有の変数 (ONTAP、Linux、または Oracle) を統合します。

1. VARSセクションから変数を編集してコピーし、これらの変数を `vars.yml`ファイル。

```
#######################################################################
###### 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 alternatingly
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する

必要な環境の前提条件を完了し、変数をコピーした後、 `vars.yml`そして `your_host.yml`これで、プレイブックをデプロイ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クラスタのユーザー名を渡して、 ONTAPプレイブックを実行します。 ONTAPクラスターのパスワードを入力し、プロンプトが表示されたら 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 プレイブックを実行して、デプロイメントの Linux 部分を実行します。管理者の 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 プレイブックを実行して、デプロイメントの Oracle 部分を実行します。管理者の 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する

プレイブックの Oracle 部分では、実行ごとに Oracle サーバー上に単一の Oracle コンテナ データベースが作成されます。同じサーバー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 変数を修正します。
   a. ステップ3に戻って編集します `host_name.yml`ファイル `host_vars`。
   b. Oracle SID を別の命名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. 修正したホスト変数をコピーして、次のOracleホスト変数ファイルに貼り付けます。 `host_vars` 。
2. プレイブックを実行するには、 `oracle_config`上記のタグ[プレイブックを実行する]。

Oracleのインストールを検証する

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

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

> インストールが期待通りに完了し、Oracle DBが起動した場合、Oracleプロセスが一覧表示されます。

2. データベースにログインして、db 構成設定と次のコマンド セットで作成された 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。

どこに助けを求めたらいいですか**?**

ツールキットに関するサポートが必要な場合は、"NetAppソリューション オートメーション コミュニティ サポート Slack チャンネル"質問や問い合わせを投稿するには、ソリューション自動化チャネルを探してください。

著作権に関する情報

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

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

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

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

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

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

本書に含まれるデータは商用製品および / または商用サービス（FAR 2.101の定義に基づく）に関係し、データの所有権はNetApp, Inc.にあります。本契約に基づき提供されるすべてのネットアップの技術データおよびコンピュータ ソフトウェアは、商用目的であり、私費のみで開発されたものです。米国政府は本データに対し、非独占的かつ移転およびサブライセンス不可で、全世界を対象とする取り消し不能の制限付き使用権を有し、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権については、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。