



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

NetApp database solutions

NetApp
August 18, 2025

目次

TR-4987: NFS を使用したAzure NetApp Filesへの簡素化された自動 Oracle デプロイメント	1
目的	1
観客	1
ソリューションのテストおよび検証環境	1
アーキテクチャ	2
ハードウェアおよびソフトウェアコンポーネント	2
ラボ環境での Oracle データベース構成	3
導入検討の重要な要素	3
ソリューションの展開	3
展開の前提条件	4
自動化パラメータファイル	6
パラメータファイルの構成	6
プレイブックの実行	9
実行後の検証	11
Oracle データベースを Azure に移行する	18
SnapCenterによる Oracle のバックアップ、リストア、クローン作成	18
詳細情報の入手方法	19

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

アレン・カオ、ニヤズ・モハメド、NetApp

このソリューションでは、NFS プロトコルを使用したプライマリ データベース ストレージとしての Microsoft Azure NetApp Files での Oracle の自動デプロイの概要と詳細を提供し、Oracle データベースは dNFS が有効になっているコンテナ データベースとしてデプロイされます。

目的

パフォーマンスが重視され、レイテンシが重視される Oracle ワークロードをクラウドで実行するのは難しい場合があります。Azure NetApp Files (ANF) を使用すると、企業の基幹業務 (LOB) およびストレージ プロフェッショナルは、コードを変更することなく、要求の厳しい Oracle ワークロードを簡単に移行して実行できます。Azure NetApp Files は、オンプレミスから Azure への Oracle データベースの新規展開や移行 (リフトアンドシフト) など、さまざまなシナリオで基盤となる共有ファイル ストレージ サービスとして広く使用されています。

このドキュメントでは、Ansible 自動化を使用した NFS マウント経由で Azure NetApp ファイルに Oracle データベースを簡略化してデプロイする方法を説明します。Oracle データベースは、パフォーマンスを向上させるために Oracle dNFS プロトコルが有効になっているコンテナ データベース (CDB) およびプラガブル データベース (PDB) 構成でデプロイされます。さらに、自動化された PDB 再配置方法を使用して、サービスの中断を最小限に抑えながら、オンプレミスの Oracle 単一インスタンス データベースまたは PDB を Azure に新しくデプロイされたコンテナ データベースに移行できます。また、Azure クラウドの NetApp SnapCenter UI ツールを使用した Oracle データベースの高速バックアップ、復元、クローン作成に関する情報も提供します。

このソリューションは、次のユースケースに対応します。

- Azure NetApp ファイルへの Oracle コンテナ データベースの自動デプロイ
- オンプレミスと Azure クラウド間の Oracle データベースの自動移行

観客

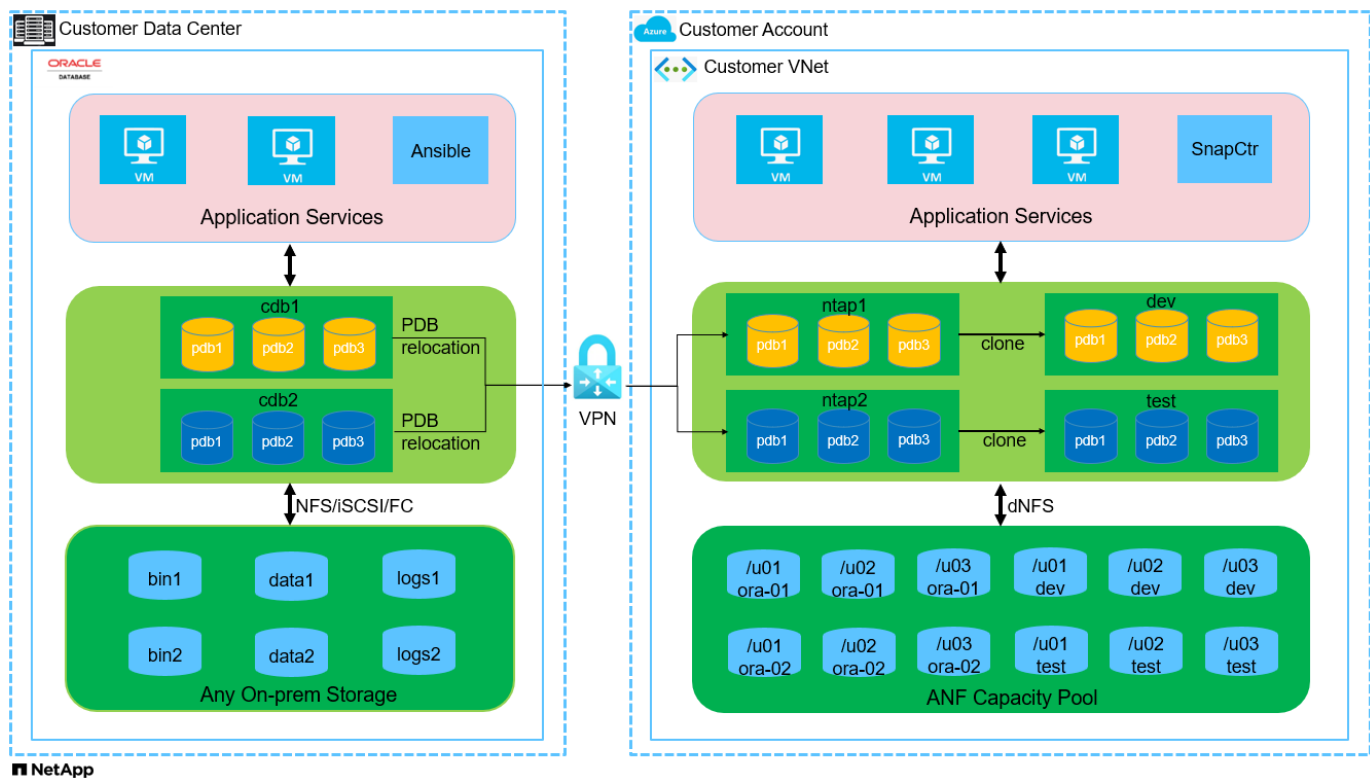
このソリューションは次の人々を対象としています。

- Azure NetApp Files に Oracle をデプロイしたい DBA。
- Azure NetApp Files で Oracle ワークロードをテストしたいデータベース ソリューション アーキテクト。
- Azure NetApp Files 上に Oracle データベースを展開および管理したいストレージ管理者。
- Azure NetApp Files 上に Oracle データベースを立ち上げたいアプリケーション所有者。

ソリューションのテストおよび検証環境

このソリューションのテストと検証は、最終的な展開環境と一致しない可能性のあるラボ設定で実行されました。セクションを参照[\[導入検討の重要な要素\]](#)詳細についてはこちらをご覧ください。

アーキテクチャ



ハードウェアおよびソフトウェアコンポーネント

ハードウェア		
Azure NetApp Files	Microsoft による Azure の現在のサービス	プレミアム サービス レベルの 1 つの容量プール
DB サーバー用の Azure VM	Standard_B4ms - 4 つの vCPU、16 GiB	同時展開用の 2 つの Linux 仮想マシンインスタンス
SnapCenter用の Azure VM	Standard_B4ms - 4 つの vCPU、16 GiB	1つのWindows仮想マシンインスタンス
ソフトウェア		
レッドハットリナックス	RHEL Linux 8.6 (LVM) - x64 Gen2	テスト用にRedHatサブスクリプションを導入
Windows Server	2022 DataCenter; Azure エディション ホットパッチ - x64 Gen2	SnapCenterサーバーのホスティング
Oracle Database	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	バージョン 12.2.0.1.36	最新パッチ p6880880_190000_Linux-x86-64.zip
SnapCenter Server	バージョン5.0	ワークグループ展開

Open JDK	バージョン java-11-openjdk	DB VM でのSnapCenterプラグインの要件
NFS	バージョン3.0	Oracle dNFS 対応
Ansible	コア 2.16.2	Python 3.6.8

ラボ環境での Oracle データベース構成

サーバ	データベース	DBストレージ
オラ-01	NTAP1(NTAP1_PDB1、NTAP1_PDB2、NTAP1_PDB3)	/u01、/u02、/u03 NFS は ANF 容量プールにマウントされます
オラ-02	NTAP2(NTAP2_PDB1、NTAP2_PDB2、NTAP2_PDB3)	/u01、/u02、/u03 NFS は ANF 容量プールにマウントされます

導入検討の重要な要素

- ***Oracle データベースのストレージ レイアウト。***この自動化された Oracle デプロイメントでは、デフォルトで Oracle バイナリ、データ、およびログをホストするために、データベースごとに 3 つのデータベース ボリュームをプロビジョニングします。ボリュームは、NFS 経由で Oracle DB サーバーに /u01 - バイナリ、/u02 - データ、/u03 - ログとしてマウントされます。冗長性のために、/u02 および /u03 マウント ポイントにデュアル制御ファイルが構成されています。
- ***複数の DB サーバーの展開。***自動化ソリューションでは、Ansible プレイブックを 1 回実行するだけで、Oracle コンテナ データベースを複数の DB サーバーにデプロイできます。DB サーバーの数に関係なく、プレイブックの実行は同じままです。異なるデータベース インスタンス ID (Oracle SID) を使用してデプロイメントを繰り返すことで、複数のコンテナ データベースを単一の VM インスタンスにデプロイできます。ただし、デプロイされたデータベースをサポートするためにホストに十分なメモリがあることを確認してください。
- **dNFS 構成。**dNFS (Oracle 11g 以降で利用可能) を使用すると、Azure 仮想マシン上で実行される Oracle データベースは、ネイティブ NFS クライアントよりも大幅に多くの I/O を処理できます。自動化された Oracle デプロイメントでは、デフォルトで NFSv3 上に dNFS が構成されます。
- **展開を高速化するために大きなサイズのボリュームを割り当てます。**ANF ファイル システムの IO スループットは、ボリュームのサイズに基づいて調整されます。初期展開では、大きなサイズのボリュームを割り当てることで展開を高速化できます。その後、アプリケーションに影響を与えることなく、ボリュームのサイズを動的に縮小できます。
- **データベースのバックアップ。**NetApp は、ユーザーフレンドリーな UI インターフェイスを備えたデータベースのバックアップ、リストア、クローン作成用の SnapCenter software スイートを提供しています。NetApp、高速 (1 分未満) スナップショット バックアップ、迅速な (数分) データベース リストア、およびデータベース クローンを実現するために、このような管理ツールを実装することを推奨しています。

ソリューションの展開

次のセクションでは、NFS 経由で Azure VM に直接マウントされたデータベース ボリュームを使用して、Azure NetApp Files で Oracle 19c を自動化してデータベースを移行するための手順を段階的に説明します。

展開の前提条件

展開には次の前提条件が必要です。

1. Azure アカウントが設定され、必要な VNet とネットワーク セグメントが Azure アカウント内に作成されています。
2. Azure クラウド ポータルから、Azure Linux VM を Oracle DB サーバーとしてデプロイします。Oracle データベース用の Azure NetApp Files 容量プールとデータベース ボリュームを作成します。DB サーバーへの azureuser の VM SSH 秘密/公開キー認証を有効にします。環境設定の詳細については、前のセクションのアーキテクチャ図を参照してください。参照：["Azure VM および Azure NetApp Files での Oracle の展開手順"](#) 詳細情報については。



ローカル ディスク冗長性を使用してデプロイされた Azure VM の場合、Oracle インストール ファイルをステージングし、OS スワップ ファイルを追加するための十分な領域を確保するために、VM ルート ディスクに少なくとも 128 G が割り当てられていることを確認してください。それに応じて、/tmplv および /rootlv OS パーティションを拡張します。1G 未満の場合、rootvg-homelv に 1G の空き領域を追加します。データベース ボリュームの命名が VMname-u01、VMname-u02、および VMname-u03 規則に従っていることを確認します。

```
sudo lvresize -r -L +20G /dev/mapper/rootvg-rootlv
```

```
sudo lvresize -r -L +10G /dev/mapper/rootvg-tmplv
```

```
sudo lvresize -r -L +1G /dev/mapper/rootvg-homelv
```

3. Azure クラウド ポータルから、最新バージョンの NetApp SnapCenter UI ツールを実行するための Windows サーバー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 Server のインストール"](#)
4. 最新バージョンの Ansible と Git がインストールされた Linux VM を Ansible コントローラー ノードとして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 ソリューション自動化入門"](#) 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.



Ansible コントローラー ノードは、SSH ポート経由で Azure DB VM に到達できる限り、オンプレミスまたは Azure クラウドのいずれかに配置できます。

5. NFS 用の NetApp Oracle 導入自動化ツールキットのコピーを複製します。

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

6. 777 権限を持つ Azure DB VM の /tmp/archive ディレクトリに次の Oracle 19c インストール ファイルをステージングします。

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

7. 次のビデオをご覧ください:

[NFS を使用した Azure NetApp Files への Oracle の簡素化と自動化](#)

自動化パラメータファイル

Ansible プレイブックは、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および構成タスクを実行します。この Oracle 自動化ソリューションには、プレイブックの実行前にユーザー入力が必要な 3 つのユーザー定義パラメータ ファイルがあります。

- ホスト - 自動化プレイブックが実行され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 - 名前付き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。私たちのユースケースでは、これらは Oracle DB サーバーです。

これらのユーザー定義変数ファイルに加えて、必要がない限り変更する必要のないデフォルト パラメータを含むデフォルト変数ファイルがいくつかあります。次のセクションでは、ユーザー定義変数ファイルを構成する方法を示します。

パラメータファイルの構成

1. Ansibleターゲット `hosts` ファイル構成:

```
# 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 user configuration variables
#####
##### Consolidate all variables from ONTAP, linux and oracle
#####
#####
#####

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

# Prerequisite to create three volumes in NetApp ONTAP storage from
System Manager or cloud dashboard with following naming convention:
# db_hostname_u01 - Oracle binary
# db_hostname_u02 - Oracle data
# db_hostname_u03 - Oracle redo
# It is important to strictly follow the name convention or the
automation will fail.

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

redhat_sub_username: XXXXXXXXX
redhat_sub_password: XXXXXXXXX

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

# Database domain name
db_domain: solutions.netapp.com

# Set initial password for all required Oracle passwords. Change
them after installation.
initial_pwd_all: XXXXXXXXX
```

3. ローカルDBサーバー `host\_vars/host\_name.yml` ora_01.yml、 ora_02.yml などの構成...

```
# 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

# Local NFS lif ip address to access database volumes
nfs_lif: 172.30.136.68
```

プレイブックの実行

自動化ツールキットには合計 5 つのプレイブックがあります。それぞれ異なるタスク ブロックを実行し、異なる目的を果たします。

```
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.
4-oracle_config.yml - install and configure Oracle on DB servers 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には、3 つのオプションがあります。

1. すべてのデプロイメント プレイブックを 1 回の実行でまとめて実行します。

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

2. 1 ～ 4 の番号シーケンスを使用して、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 azureuser -e
@vars/vars.yml
```

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

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

3. タグ付きで 0-all_playbook.yml を実行します。

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

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

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

4. 環境を元に戻す

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

実行後の検証

プレイブックの実行後、Oracle DB サーバー VM にログインして、Oracle がインストールおよび構成され、コンテナ データベースが正常に作成されたことを確認します。以下は、ホスト ora-01 上の Oracle データベース検証の例です。

1. NFSマウントを検証する

```
[azureuser@ora-01 ~]$ cat /etc/fstab

#
# /etc/fstab
# Created by anaconda on Thu Sep 14 11:04:01 2023
#
# Accessible filesystems, by reference, are maintained under
# '/dev/disk/'.
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for
# more info.
#
# After editing this file, run 'systemctl daemon-reload' to update
# systemd
# units generated from this file.
#
/dev/mapper/rootvg-rootlv / xfs defaults
0 0
UUID=268633bd-f9bb-446d-9a1d-8fca4609a1e1 /boot
xfs defaults 0 0
UUID=89D8-B037 /boot/efi vfat
defaults,uid=0,gid=0,umask=077,shortname=winnt 0 2
/dev/mapper/rootvg-homelv /home xfs defaults
0 0
/dev/mapper/rootvg-tmplv /tmp xfs defaults
0 0
/dev/mapper/rootvg-usrlv /usr xfs defaults
0 0
/dev/mapper/rootvg-varlv /var xfs defaults
0 0
/mnt/swapfile swap swap defaults 0 0
172.30.136.68:/ora-01-u01 /u01 nfs
rw,bg,hard,vers=3,proto=tcp,timeo=600,rsiz=65536,wsiz=65536 0 0
172.30.136.68:/ora-01-u02 /u02 nfs
rw,bg,hard,vers=3,proto=tcp,timeo=600,rsiz=65536,wsiz=65536 0 0
172.30.136.68:/ora-01-u03 /u03 nfs
rw,bg,hard,vers=3,proto=tcp,timeo=600,rsiz=65536,wsiz=65536 0 0

[azureuser@ora-01 ~]$ df -h
Filesystem Size Used Avail Use% Mounted on
```

devtmpfs	7.7G	0	7.7G	0%	/dev
tmpfs	7.8G	0	7.8G	0%	/dev/shm
tmpfs	7.8G	8.6M	7.7G	1%	/run
tmpfs	7.8G	0	7.8G	0%	/sys/fs/cgroup
/dev/mapper/rootvg-rootlv	22G	17G	5.8G	74%	/
/dev/mapper/rootvg-usrlv	10G	2.0G	8.1G	20%	/usr
/dev/mapper/rootvg-varlv	8.0G	890M	7.2G	11%	/var
/dev/sda1	496M	106M	390M	22%	/boot
/dev/mapper/rootvg-homelv	1014M	40M	975M	4%	/home
/dev/sda15	495M	5.9M	489M	2%	/boot/efi
/dev/mapper/rootvg-tmplv	12G	8.4G	3.7G	70%	/tmp
tmpfs	1.6G	0	1.6G	0%	/run/user/54321
172.30.136.68:/ora-01-u01	500G	11G	490G	3%	/u01
172.30.136.68:/ora-01-u03	250G	1.2G	249G	1%	/u03
172.30.136.68:/ora-01-u02	250G	7.1G	243G	3%	/u02
tmpfs	1.6G	0	1.6G	0%	/run/user/1000

2. Oracleリスナーを検証する

```
[azureuser@ora-01 ~]$ sudo su
[root@ora-01 azureuser]# su - oracle
Last login: Thu Feb  1 16:13:44 UTC 2024
[oracle@ora-01 ~]$ lsnrctl status listener.ntap1

LSNRCTL for Linux: Version 19.0.0.0.0 - Production on 01-FEB-2024
16:25:37

Copyright (c) 1991, 2022, Oracle. All rights reserved.

Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=TCP) (HOST=ora-
01.internal.cloudapp.net) (PORT=1521)))
STATUS of the LISTENER
-----
Alias                     LISTENER.NTAP1
Version                   TNSLSNR for Linux: Version 19.0.0.0.0 -
Production
Start Date                01-FEB-2024 16:13:49
Uptime                    0 days 0 hr. 11 min. 49 sec
Trace Level               off
Security                  ON: Local OS Authentication
SNMP                      OFF
Listener Parameter File   /u01/app/oracle/product/19.0.0/NTAP1/network/admin/listener.ora
Listener Log File         /u01/app/oracle/diag/tnslsnr/ora-
01/listener.ntap1/alert/log.xml
```

Listening Endpoints Summary...

```
(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp) (HOST=ora-01.hr2z2nbmhnqutdsxgscjtuxizd.jx.internal.cloudapp.net) (PORT=1521)))
(DESCRIPTION=(ADDRESS=(PROTOCOL=ipc) (KEY=EXTPROC1521)))
(DESCRIPTION=(ADDRESS=(PROTOCOL=tcps) (HOST=ora-01.hr2z2nbmhnqutdsxgscjtuxizd.jx.internal.cloudapp.net) (PORT=5500)) (
Security=(my_wallet_directory=/u01/app/oracle/product/19.0.0/NTAP1/admin/NTAP1/xdb_wallet)) (Presentation=HTTP) (Session=RAW))
```

Services Summary...

Service "104409ac02da6352e063bb891eacf34a.solutions.netapp.com" has 1 instance(s).

Instance "NTAP1", status READY, has 1 handler(s) for this service...

Service "104412c14c2c63cae063bb891eacf64d.solutions.netapp.com" has 1 instance(s).

Instance "NTAP1", status READY, has 1 handler(s) for this service...

Service "1044174670ad63ffe063bb891eac6b34.solutions.netapp.com" has 1 instance(s).

Instance "NTAP1", status READY, has 1 handler(s) for this service...

Service "NTAP1.solutions.netapp.com" has 1 instance(s).

Instance "NTAP1", status READY, has 1 handler(s) for this service...

Service "NTAP1XDB.solutions.netapp.com" has 1 instance(s).

Instance "NTAP1", status READY, has 1 handler(s) for this service...

Service "ntap1_pdb1.solutions.netapp.com" has 1 instance(s).

Instance "NTAP1", status READY, has 1 handler(s) for this service...

Service "ntap1_pdb2.solutions.netapp.com" has 1 instance(s).

Instance "NTAP1", status READY, has 1 handler(s) for this service...

Service "ntap1_pdb3.solutions.netapp.com" has 1 instance(s).

Instance "NTAP1", status READY, has 1 handler(s) for this service...

The command completed successfully

3. OracleデータベースとdNFSを検証する

```
[oracle@ora-01 ~]$ cat /etc/oratab
#
# This file is used by ORACLE utilities.  It is created by root.sh
# and updated by either Database Configuration Assistant while
# creating
```



```
# a database or ASM Configuration Assistant while creating ASM
instance.

# A colon, ':', is used as the field terminator.  A new line
terminates
# the entry.  Lines beginning with a pound sign, '#', are comments.
#
# Entries are of the form:
#   $ORACLE_SID:$ORACLE_HOME:<N|Y>:
#
# The first and second fields are the system identifier and home
# directory of the database respectively.  The third field indicates
# to the dbstart utility that the database should , "Y", or should
not,
# "N", be brought up at system boot time.
#
# Multiple entries with the same $ORACLE_SID are not allowed.
#
#
NTAP1:/u01/app/oracle/product/19.0.0/NTAP1:Y
```

```
[oracle@ora-01 ~]$ sqlplus / as sysdba
```

```
SQL*Plus: Release 19.0.0.0.0 - Production on Thu Feb 1 16:37:51 2024
Version 19.18.0.0.0
```

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

```
Connected to:
```

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

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

NAME	OPEN_MODE	LOG_MODE
NTAP1	READ WRITE	ARCHIVELOG

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

NAME

```
-----
-----
/u02/oradata/NTAP1/system01.dbf
/u02/oradata/NTAP1/sysaux01.dbf
/u02/oradata/NTAP1/undotbs01.dbf
/u02/oradata/NTAP1/pdbseed/system01.dbf
/u02/oradata/NTAP1/pdbseed/sysaux01.dbf
/u02/oradata/NTAP1/users01.dbf
/u02/oradata/NTAP1/pdbseed/undotbs01.dbf
/u02/oradata/NTAP1/NTAP1_pdb1/system01.dbf
/u02/oradata/NTAP1/NTAP1_pdb1/sysaux01.dbf
/u02/oradata/NTAP1/NTAP1_pdb1/undotbs01.dbf
/u02/oradata/NTAP1/NTAP1_pdb1/users01.dbf
```

NAME

```
-----
-----
/u02/oradata/NTAP1/NTAP1_pdb2/system01.dbf
/u02/oradata/NTAP1/NTAP1_pdb2/sysaux01.dbf
/u02/oradata/NTAP1/NTAP1_pdb2/undotbs01.dbf
/u02/oradata/NTAP1/NTAP1_pdb2/users01.dbf
/u02/oradata/NTAP1/NTAP1_pdb3/system01.dbf
/u02/oradata/NTAP1/NTAP1_pdb3/sysaux01.dbf
/u02/oradata/NTAP1/NTAP1_pdb3/undotbs01.dbf
/u02/oradata/NTAP1/NTAP1_pdb3/users01.dbf
```

19 rows selected.

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

NAME

```
-----
-----
/u02/oradata/NTAP1/control01.ctl
/u03/orareco/NTAP1/control02.ctl
```

```
SQL> select member from v$logfile;
```

MEMBER

```
-----
-----
```

```
/u03/orareco/NTAP1/onlineelog/redo03.log
```

```
/u03/orareco/NTAP1/onlineelog/redo02.log
```

```
/u03/orareco/NTAP1/onlineelog/redo01.log
```

```
SQL> select svrname, dirname, nfsversion from v$dnfs_servers;
```

```
SVRNAME
```

```
-----  
-----
```

```
DIRNAME
```

```
-----  
-----
```

```
NFSVERSION
```

```
-----
```

```
172.30.136.68
```

```
/ora-01-u02
```

```
NFSv3.0
```

```
172.30.136.68
```

```
/ora-01-u03
```

```
NFSv3.0
```

```
SVRNAME
```

```
-----  
-----
```

```
DIRNAME
```

```
-----  
-----
```

```
NFSVERSION
```

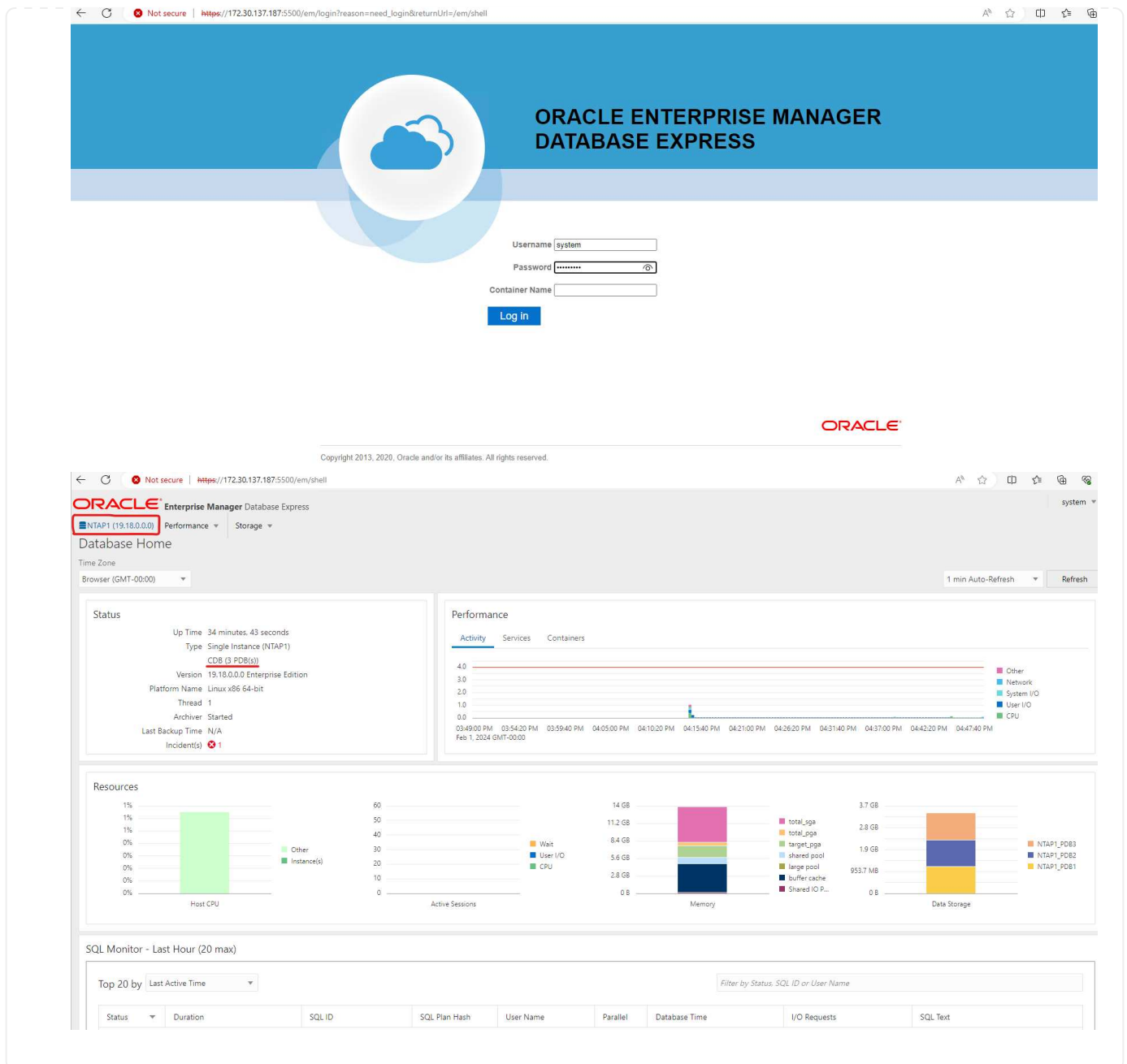
```
-----
```

```
172.30.136.68
```

```
/ora-01-u01
```

```
NFSv3.0
```

4. データベースを検証するには、Oracle Enterprise Manager Express にログインします。



Oracle データベースを Azure に移行する

オンプレミスからクラウドへの Oracle データベースの移行は大変な作業です。適切な戦略と自動化を使用すると、プロセスをスムーズにし、サービスの中断とダウンタイムを最小限に抑えることができます。この詳細な指示に従ってください["オンプレミスから Azure クラウドへのデータベース移行"](#)データベース移行のプロセスをガイドします。

SnapCenterによる Oracle のバックアップ、リストア、クローン作成

NetApp は、Azure クラウドに展開された Oracle データベースを管理するために SnapCenter UI ツールを推奨しています。TR-4988を参照してください。["SnapCenterを使用した ANF 上の Oracle データベースのバックアップ、リカバリ、およびクローン"](#)詳細については。

詳細情報の入手方法

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

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

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

- Azure NetApp Files

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

- Oracle Direct NFSの導入

["https://docs.oracle.com/en/database/oracle/oracle-database/19/ladbi/deploying-dnfs.html#GUID-D06079DB-8C71-4F68-A1E3-A75D7D96DCE2"](https://docs.oracle.com/en/database/oracle/oracle-database/19/ladbi/deploying-dnfs.html#GUID-D06079DB-8C71-4F68-A1E3-A75D7D96DCE2)

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

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

著作権に関する情報

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

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

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

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

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

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

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

商標に関する情報

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