



TR-4987: 使用 NFS 在 Azure NetApp Files 上简化、自动化 Oracle 部署

NetApp database solutions

NetApp
August 18, 2025

目录

TR-4987: 使用 NFS 在Azure NetApp Files上简化、自动化 Oracle 部署	1
目的	1
受众	1
解决方案测试和验证环境	1
架构	1
硬件和软件组件	2
实验室环境中的 Oracle 数据库配置	3
部署考虑的关键因素	3
解决方案部署	3
部署先决条件	3
自动化参数文件	5
参数文件配置	5
剧本执行	8
执行后验证	10
将 Oracle 数据库迁移到 AzureMigrate Oracle database to Azure	17
使用SnapCenter进行 Oracle 备份、恢复和克隆	17
在哪里可以找到更多信息	18

TR-4987：使用 NFS 在 Azure NetApp Files 上简化、自动化 Oracle 部署

Allen Cao、Niyaz Mohamed，NetApp

该解决方案提供了在 Microsoft Azure NetApp Files 中自动部署 Oracle 的概述和详细信息，该部署作为具有 NFS 协议的主数据库存储，并且 Oracle 数据库部署为启用 dNFS 的容器数据库。

目的

在云中运行性能密集型和延迟敏感的 Oracle 工作负载可能具有挑战性。Azure NetApp Files (ANF) 使企业业务线 (LOB) 和存储专业人员可以轻松迁移和运行要求苛刻的 Oracle 工作负载，而无需更改代码。Azure NetApp Files 被广泛用作各种场景中的底层共享文件存储服务，例如 Oracle 数据库的新部署或从本地到 Azure 的迁移（提升和转移）。

本文档演示了使用 Ansible 自动化通过 NFS 挂载在 Azure NetApp 文件中简化 Oracle 数据库的部署。Oracle 数据库部署在容器数据库 (CDB) 和可插拔数据库 (PDB) 配置中，并启用 Oracle dNFS 协议以提高性能。此外，可以使用自动化 PDB 迁移方法将本地 Oracle 单实例数据库或 PDB 迁移到 Azure 中新部署的容器数据库中，同时最大程度地减少服务中断。它还提供有关使用 Azure Cloud 中的 NetApp SnapCenter UI 工具快速备份、还原和克隆 Oracle 数据库的信息。

此解决方案适用于以下用例：

- Azure NetApp 文件上的自动 Oracle 容器数据库部署
- 本地和 Azure 云之间的自动 Oracle 数据库迁移

受众

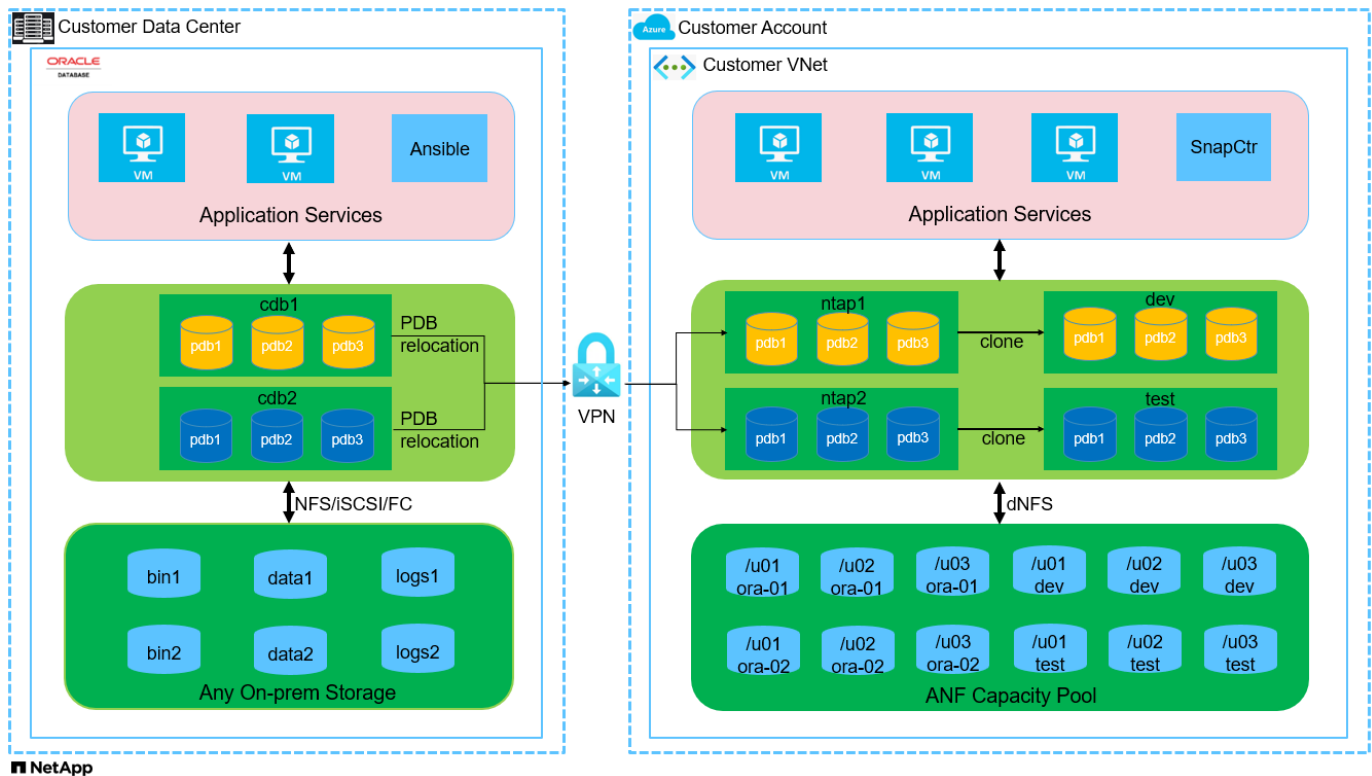
此解决方案适用于以下人群：

- 想要在 Azure NetApp Files 上部署 Oracle 的 DBA。
- 想要在 Azure NetApp Files 上测试 Oracle 工作负载的数据库解决方案架构师。
- 想要在 Azure NetApp Files 上部署和管理 Oracle 数据库的存储管理员。
- 希望在 Azure NetApp Files 上建立 Oracle 数据库的应用程序所有者。

解决方案测试和验证环境

该解决方案的测试和验证是在实验室环境中进行的，可能与最终部署环境不匹配。请参阅[部署考虑的关键因素](#)了解更多信息。

架构



硬件和软件组件

硬件		
Azure NetApp Files	Microsoft 目前在 Azure 中提供的产品	一个具有高级服务级别的容量池
用于数据库服务器的 Azure VM	Standard_B4ms - 4 个 vCPU , 16GiB	两个 Linux 虚拟机实例用于并发部署
适用于 SnapCenter 的 Azure VM	Standard_B4ms - 4 个 vCPU , 16GiB	一个 Windows 虚拟机实例
软件		
红帽 Linux	RHEL Linux 8.6 (LVM) -x64 Gen2	部署 RedHat 订阅进行测试
Windows Server	2022 数据中心; Azure 版本热补丁 - x64 Gen2	托管 SnapCenter 服务器
Oracle 数据库	19.18 版	已应用 RU 补丁 p34765931_190000_Linux-x86-64.zip
Oracle OPatch	版本 12.2.0.1.36	最新补丁 p6880880_190000_Linux-x86-64.zip
SnapCenter 服务器	5.0 版	工作组部署
打开 JDK	版本 java-11-openjdk	数据库虚拟机上的 SnapCenter 插件要求
NFS	3.0 版	已启用 Oracle dNFS

Ansible	核心 2.16.2	Python 3.6.8
---------	-----------	--------------

实验室环境中的 Oracle 数据库配置

服务器	数据库	数据库存储
ora-01	NTAP1(NTAP1_PDB1、NTAP1_PDB2、NTAP1_PDB3)	/u01、/u02、/u03 在 ANF 容量池上挂载 NFS
ora-02	NTAP2(NTAP2_PDB1、NTAP2_PDB2、NTAP2_PDB3)	/u01、/u02、/u03 在 ANF 容量池上挂载 NFS

部署考虑的关键因素

- Oracle 数据库存储布局。在此自动化 Oracle 部署中，我们默认为每个数据库配置三个数据库卷来托管 Oracle 二进制文件、数据和日志。这些卷通过 NFS 安装在 Oracle DB 服务器上，作为 /u01 - 二进制、/u02 - 数据、/u03 - 日志。在 /u02 和 /u03 挂载点上配置双控制文件以实现冗余。
- *多个数据库服务器部署。*自动化解决方案可以在单个 Ansible 剧本运行中将 Oracle 容器数据库部署到多个 DB 服务器。无论数据库服务器的数量有多少，剧本的执行都保持不变。您可以通过使用不同的数据库实例 ID（Oracle SID）重复部署将多个容器数据库部署到单个 VM 实例。但要确保主机上有足够的内存来支持部署的数据库。
- *dNFS 配置。*通过使用 dNFS（自 Oracle 11g 起可用），在 Azure 虚拟机上运行的 Oracle 数据库可以比本机 NFS 客户端驱动更多的 I/O。自动化 Oracle 部署默认在 NFSv3 上配置 dNFS。
- 分配大尺寸卷以加快部署速度。ANF 文件系统 IO 吞吐量根据卷的大小进行调节。对于初始部署，分配大尺寸卷可以加快部署速度。随后可以动态缩小卷的大小，而不会对应用程序产生影响。
- 数据库备份。NetApp提供了SnapCenter software套件，用于数据库备份、恢复和克隆，并具有用户友好的 UI 界面。NetApp建议实施这样的管理工具，以实现快速（一分钟内）快照备份、快速（几分钟内）数据库恢复和数据库克隆。

解决方案部署

以下部分提供了在 Azure NetApp Files 上自动部署 Oracle 19c 和数据库迁移的分步过程，其中数据库卷通过 NFS 直接装载到 Azure VM。

部署先决条件

部署需要以下先决条件。

1. 已设置 Azure 帐户，并在您的 Azure 帐户内创建了必要的 VNet 和网络段。
2. 从 Azure 云门户，将 Azure Linux VM 部署为 Oracle DB 服务器。为 Oracle 数据库创建 Azure NetApp Files 容量池和数据库卷。为 azureuser 到 DB 服务器启用 VM SSH 私钥/公钥身份验证。有关环境设置的详细信息，请参阅上一节中的架构图。也提到["Azure VM 和 Azure NetApp Files 上的分步 Oracle 部署过程"](#)了解详细信息。



对于使用本地磁盘冗余部署的 Azure VM，请确保在 VM 根磁盘中分配至少 128G，以便有足够的空间来暂存 Oracle 安装文件并添加 OS 交换文件。相应地扩展 /tmp 和 /root 分区。如果 rootvg-homes 的可用空间小于 1G，则添加 1G 的可用空间。确保数据库卷命名遵循 VMname-u01、VMname-u02 和 VMname-u03 约定。

```
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-homeslv
```

3. 从 Azure 云门户配置 Windows 服务器以运行最新版本的 NetApp SnapCenter UI 工具。详细信息请参考以下链接：["安装 SnapCenter 服务器"](#)
4. 将 Linux VM 配置为 Ansible 控制器节点，并安装最新版本的 Ansible 和 Git。详细信息请参考以下链接：["NetApp 解决方案自动化入门"](#)在第 -
Setup the Ansible Control Node for CLI deployments on RHEL / CentOS`或者
`Setup the Ansible Control Node for CLI deployments on Ubuntu / Debian。



Ansible 控制器节点可以位于本地或 Azure 云中，只要它可以通过 ssh 端口访问 Azure DB VM。

5. 克隆适用于 NFS 的 NetApp Oracle 部署自动化工具包的副本。

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

6. 在 Azure DB VM /tmp/archive 目录上以 777 权限执行以下 Oracle 19c 安装文件阶段。

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

7. 观看以下视频：

[使用 NFS 在 Azure NetApp Files 上简化和自动化 Oracle 部署](#)

自动化参数文件

Ansible playbook 使用预定义参数执行数据库安装和配置任务。对于此 Oracle 自动化解决方案，有三个用户定义的参数文件需要在剧本执行之前用户输入。

- 主机 - 定义自动化剧本运行的目标。
- vars/vars.yml - 定义适用于所有目标的变量的全局变量文件。
- host_vars/host_name.yml - 定义仅适用于命名目标的变量的本地变量文件。在我们的用例中，这些是 Oracle DB 服务器。

除了这些用户定义的变量文件之外，还有几个默认变量文件，其中包含默认参数，除非必要，否则不需要更改。以下部分介绍如何配置用户定义的变量文件。

参数文件配置

1. Ansible 目标 `hosts` 文件配置：

```
# 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. 全球的 `vars/vars.yml` 文件配置

```
#####
##
##### 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. 本地数据库服务器 `host\_vars/host\_name.yml` 配置如ora_01.yml, ora_02.yml...

```
# 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
```

剧本执行

自动化工具包中共有五个剧本。每个执行不同的任务块并服务于不同的目的。

```
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.
```

有三个选项可以使用以下命令运行剧本。

1. 在一次组合运行中执行所有部署剧本。

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

2. 按照 1-4 的数字序列逐个执行剧本。

```
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. 使用标签执行 0-all_playbook.yml。

```
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. 撤消环境

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

执行后验证

脚本运行后，登录到 Oracle DB 服务器 VM 以验证 Oracle 是否已安装和配置，以及是否已成功创建容器数据库。以下是主机 ora-01 上的 Oracle 数据库验证的示例。

1. 验证 NFS 挂载

```
[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. 验证 Oracle 侦听器

```
[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/a
dmin/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. 验证 Oracle 数据库和 dNFS

```

[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

```
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
```

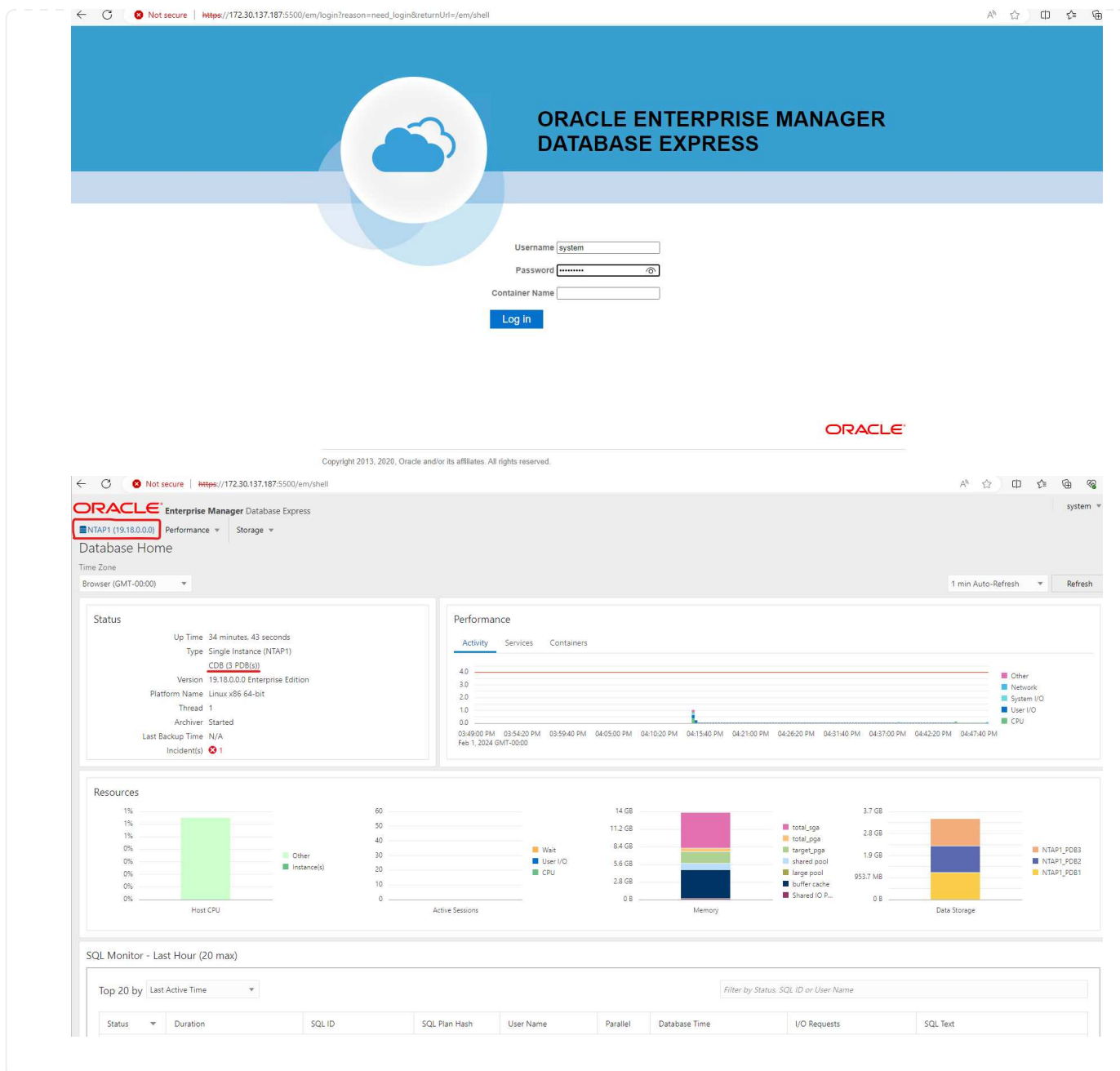
```
-----
```

```
172.30.136.68
```

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

```
NFSv3.0
```

4. 登录 Oracle Enterprise Manager Express 来验证数据库。



将 Oracle 数据库迁移到 AzureMigrate Oracle database to Azure

Oracle 数据库从本地迁移到云端是一项繁重的工作。使用正确的策略和自动化可以使流程更加顺畅，并最大限度地减少服务中断和停机时间。按照这个详细的说明操作["数据库从本地迁移到 Azure 云"](#)指导您的数据库迁移之旅。

使用SnapCenter进行 Oracle 备份、恢复和克隆

NetApp推荐使用SnapCenter UI 工具来管理部署在 Azure 云中的 Oracle 数据库。请参考TR-4988：["使用SnapCenter在 ANF 上进行 Oracle 数据库备份、恢复和克隆"](#)了解详情。

在哪里可以找到更多信息

要了解有关本文档中描述的信息的更多信息，请查看以下文档和/或网站：

- 使用SnapCenter在 ANF 上进行 Oracle 数据库备份、恢复和克隆

["使用SnapCenter在 ANF 上进行 Oracle 数据库备份、恢复和克隆"](#)

- Azure NetApp Files

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

- 部署 Oracle Direct NFS

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

- 使用响应文件安装和配置 Oracle 数据库

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

版权信息

版权所有 © 2025 NetApp, Inc.。保留所有权利。中国印刷。未经版权所有者事先书面许可，本文档中受版权保护的任何部分不得以任何形式或通过任何手段（图片、电子或机械方式，包括影印、录音、录像或存储在电子检索系统中）进行复制。

从受版权保护的 NetApp 资料派生的软件受以下许可和免责声明的约束：

本软件由 NetApp 按“原样”提供，不含任何明示或暗示担保，包括但不限于适销性以及针对特定用途的适用性的隐含担保，特此声明不承担任何责任。在任何情况下，对于因使用本软件而以任何方式造成的任何直接性、间接性、偶然性、特殊性、惩罚性或后果性损失（包括但不限于购买替代商品或服务；使用、数据或利润方面的损失；或者业务中断），无论原因如何以及基于何种责任理论，无论出于合同、严格责任或侵权行为（包括疏忽或其他行为），NetApp 均不承担责任，即使已被告知存在上述损失的可能性。

NetApp 保留在不另行通知的情况下随时对本文档所述的任何产品进行更改的权利。除非 NetApp 以书面形式明确同意，否则 NetApp 不承担因使用本文档所述产品而产生的任何责任或义务。使用或购买本产品不表示获得 NetApp 的任何专利权、商标权或任何其他知识产权许可。

本手册中描述的产品可能受一项或多项美国专利、外国专利或正在申请的专利的保护。

有限权利说明：政府使用、复制或公开本文档受 DFARS 252.227-7013（2014 年 2 月）和 FAR 52.227-19（2007 年 12 月）中“技术数据权利 — 非商用”条款第 (b)(3) 条规定的限制条件的约束。

本文档中所含数据与商业产品和/或商业服务（定义见 FAR 2.101）相关，属于 NetApp, Inc. 的专有信息。根据本协议提供的所有 NetApp 技术数据和计算机软件具有商业性质，并完全由私人出资开发。美国政府对这些数据的使用权具有非排他性、全球性、受限且不可撤销的许可，该许可既不可转让，也不可再许可，但仅限在与交付数据所依据的美国政府合同有关且受合同支持的情况下使用。除本文档规定的情形外，未经 NetApp, Inc. 事先书面批准，不得使用、披露、复制、修改、操作或显示这些数据。美国政府对国防部的授权仅限于 DFARS 的第 252.227-7015(b)（2014 年 2 月）条款中明确的权利。

商标信息

NetApp、NetApp 标识和 <http://www.netapp.com/TM> 上所列的商标是 NetApp, Inc. 的商标。其他公司和产品名称可能是其各自所有者的商标。