TR-4983: Simplified, Automated Oracle Deployment on NetApp ASA with iSCSI

NetApp database solutions

NetApp
July 31, 2025

Table of Contents

TR-4983: Simplified, Automated Oracle Deployment on NetApp ASA with iSCSI

Allen Cao, Niyaz Mohamed, NetApp

This solution provides overview and details for automated Oracle deployment and protection in NetApp ASA array as primary database storage with iSCSI protocol and Oracle database configured in standalone ReStart using asm as volume manager.

Purpose

NetApp ASA systems deliver modern solutions to your SAN infrastructure. They simplify at scale and enable you to accelerate your business-critical applications such as databases, make sure that your data is always available (99.9999% uptime), and reduce TCO and carbon footprint. The NetApp ASA systems include A-Series models designed for the most performance-demanding applications and C-Series models optimized for cost-effective, large-capacity deployments. Together, the ASA A-Series and C-Series systems deliver exceptional performance to improve customer experience and reduce time to results, keep business-critical data available, protected, and secure, and provide more effective capacity for any workload, backed by the industry's most effective guarantee.

This documentation demonstrates the simplified deployment of Oracle databases in a SAN environment built with ASA systems using Ansible automation. The Oracle database is deployed in a standalone ReStart configuration with iSCSI protocol for data access and Oracle ASM for database disks management on the ASA storage array. It also provides information on Oracle database backup, restore, and clone using the NetApp SnapCenter UI tool for storage-efficient database operation in NetApp ASA systems.

This solution addresses the following use cases:

- Automated Oracle database deployment in NetApp ASA systems as primary database storage
- Oracle database backup and restore in NetApp ASA systems using NetApp SnapCenter tool
- Oracle database clone for dev/test or other use cases in NetApp ASA systems using NetApp SnapCenter tool

Audience

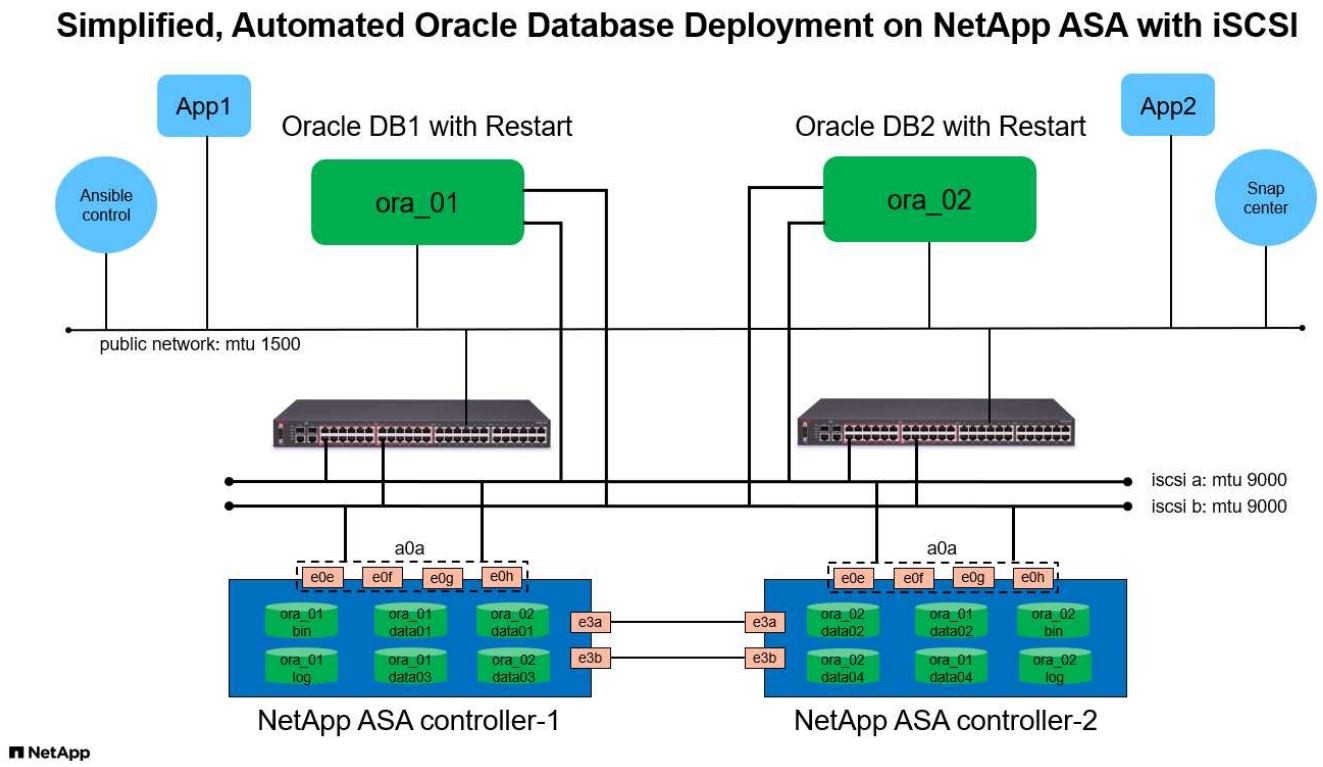
This solution is intended for the following people:

- A DBA who would like to deploy Oracle in NetApp ASA systems.
- A database solution architect who would like to test Oracle workloads in NetApp ASA systems.
- A storage administrator who would like to deploy and manage an Oracle database on NetApp ASA systems.
- An application owner who would like to stand up an Oracle database in NetApp ASA systems.

Solution test and validation environment

The testing and validation of this solution were performed in a lab setting that might not match the final deployment environment. See the section Key factors for deployment consideration for more information.

Architecture



Hardware and software components

Hardware		
NetApp ASA A400	Version 9.13.1P1	2 NS224 shelves, 48 NVMe AFF drives with total 69.3 TiB capacity
UCSB-B200-M4	Intel® Xeon® CPU E5-2690 v4 @ 2.60GHz	4-node VMware ESXi cluster
Software		
RedHat Linux	RHEL-8.6, 4.18.0-372.9.1.el8.x86_64 kernel	Deployed RedHat subscription for testing
Windows Server	2022 Standard, 10.0.20348 Build 20348	Hosting SnapCenter server
Oracle Grid Infrastructure	Version 19.18	Applied RU patch p34762026_190000_Linux-x86-64.zip
Oracle Database	Version 19.18	Applied RU patch p34765931_190000_Linux-x86-64.zip
Oracle OPatch	Version 12.2.0.1.36	Latest patch p6880880_190000_Linux-x86-64.zip
SnapCenter Server	Version 4.9P1	Workgroup deployment

VMware vSphere Hypervisor	version 6.5.0.20000	VMware Tools, Version: 11365 - Linux, 12352 - Windows
Open JDK	Version java-1.8.0-openjdk.x86_64	SnapCenter plugin requirement on DB VMs

Oracle database configuration in the lab environment

Server	Database	DB Storage
ora_01	NTAP1(NTAP1_PDB1,NTAP1_PDB2,NTAP1_PDB3)	iSCSI luns on ASA A400
ora_02	NTAP2(NTAP2_PDB1,NTAP2_PDB2,NTAP2_PDB3)	iSCSI luns on ASA A400

Key factors for deployment consideration

- **Oracle database storage layout.** In this automated Oracle deployment, we provision four database volumes to host Oracle binary, data, and logs by default. We then create two ASM disk groups from data and logs luns. Within the +DATA asm disk group, we provision two data luns in a volume on each ASA A400 cluster node. Within the +LOGS asm disk group, we create two luns in a log volume on a single ASA A400 node. Multiple luns laid out within an ONTAP volume provides better performance in general.
- **Multiple DB servers deployment.** The automation solution can deploy an Oracle container database to multiple DB servers in a single Ansible playbook run. Regardless of the number of DB servers, the playbook execution remains the same. In the event of multi-DB server deployments, the playbook builds with an algorithm to place database luns on dual controllers of ASA A400 optimally. The binary and logs luns of odd number DB server in server hosts index place on controller 1. The binary and logs luns of even number DB server in the server hosts index place on controller 2. The DB data luns evenly distributed to two controllers. Oracle ASM combines the data luns on two controllers into a single ASM disk group to fully utilize the processing power of both controllers.
- **iSCSI configuration.** The database VMs connect to ASA storage with the iSCSI protocol for storage access. You should configure dual paths on each controller node for redundancy and set up iSCSI multi-path on the DB server for multi-path storage access. Enable jumbo frame on storage network to maximize performance and throughput.
- **Oracle ASM redundancy level to use for each Oracle ASM disk group that you create.** Because the ASA A400 configures storage in RAID DP for data protection at the cluster disk level, you should use `External Redundancy`, which means that the option does not allow Oracle ASM to mirror the contents of the disk group.
- **Database backup.** NetApp provides a SnapCenter software suite for database backup, restore, and cloning with a user-friendly UI interface. NetApp recommends implementing such a management tool to achieve fast (under a minute) SnapShot backup, quick (minutes) database restore, and database clone.

Solution deployment

The following sections provide step-by-step procedures for automated Oracle 19c deployment and protection in NetApp ASA A400 with directly mounted database luns via iSCSI to DB VM in a single node Restart configuration with Oracle ASM as database volume manager.

Prerequisites for deployment

Deployment requires the following prerequisites.

1. It is assumed that the NetApp ASA storage array has been installed and configured. This includes iSCSI broadcast domain, LACP interface groups a0a on both controller nodes, iSCSI VLAN ports (a0a-<iscsi-a-vlan-id>, a0a-<iscsi-b-vlan-id>) on both controller nodes. The following link provides detailed step-by-step instructions if help is needed. Detailed guide - ASA A400
2. Provision a Linux VM as an Ansible controller node with the latest version of Ansible and Git installed. Refer to the following link for details: Getting Started with NetApp solution automation^ in section - `Setup the Ansible Control Node for CLI deployments on RHEL / CentOS` or `Setup the Ansible Control Node for CLI deployments on Ubuntu / Debian`.
3. Clone a copy of the NetApp Oracle deployment automation toolkit for iSCSI.

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

4. Provision a Windows server to run the NetApp SnapCenter UI tool with the latest version. Refer to the following link for details: Install the SnapCenter Server
5. Build two RHEL Oracle DB servers either bare metal or virtualized VM. Create an admin user on DB servers with sudo without password privilege and enable SSH private/public key authentication between Ansible host and Oracle DB server hosts. Stage following Oracle 19c installation files on DB servers /tmp/archive directory.

```
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"
```

Ensure that you have allocated at least 50G in Oracle VM root volume to have sufficient space to stage Oracle installation files.

6. Watch the following video:

   Simplified and automated Oracle deployment on NetApp ASA with iSCSI

Automation parameter files

Ansible playbook executes database installation and configuration tasks with predefined parameters. For this Oracle automation solution, there are three user-defined parameter files that need user input before playbook execution.

- hosts - define targets that the automation playbook is running against.
- vars/vars.yml - the global variable file that defines variables that apply to all targets.
- host_vars/host_name.yml - the local variable file that defines variables that apply only to a local target. In our use case, these are the Oracle DB servers.

In addition to these user-defined variable files, there are several default variable files that contain default parameters that do not require change unless necessary. The following sections show how the user-defined variable files are configured.

Parameter files configuration

1. Ansible target `hosts` file configuration:

```
# 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
```

1. Global `vars/vars.yml` file configuration

```
###########################################################################
########################################
######                  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 }}_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
```

1. Local DB server `host_vars/host_name.yml` configuration

```
# 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 execution

There are a total of six playbooks in the automation toolkit. Each performs different task blocks and serves different purposes.

```
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.
```

There are three options to run the playbooks with the following commands.

1. Execute all deployment playbooks in one combined run.

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

2. Execute playbooks one at a time with the number sequence from 1-4.

   ```
   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. Execute 0-all_playbook.yml with a tag.

```
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. Undo the environment

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

Post execution validation

After the playbook run, login to the Oracle DB server as oracle user to validate that Oracle grid infrastructure and database are created successfully. Following is an example of Oracle database validation on host ora_01.

1. Validate the grid infrastructure and resources created.

```
[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
--------------------------------------------------------------------------------
```

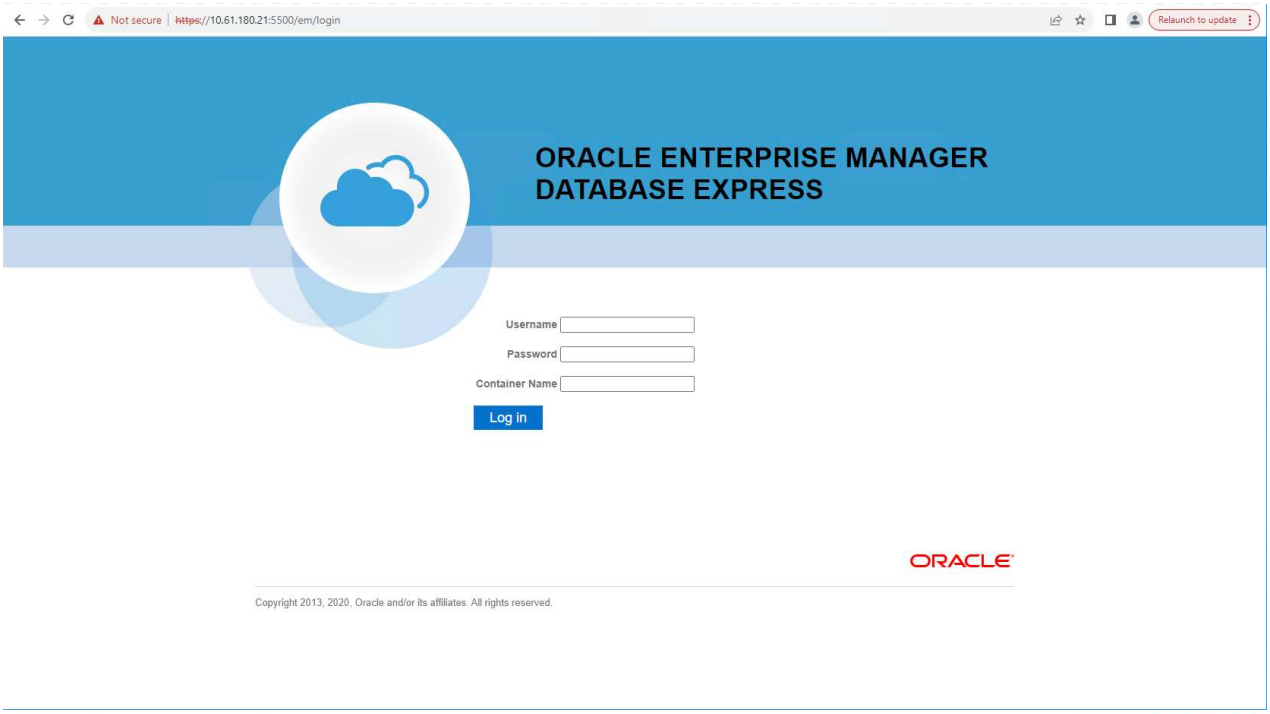
```
------------
ora.cssd
      1        ONLINE  ONLINE       ora_01                   STABLE
ora.diskmon
      1        OFFLINE OFFLINE                               STABLE
ora.driver.afd
      1        ONLINE  ONLINE       ora_01                   STABLE
ora.evmd
      1        ONLINE  ONLINE       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 ~]$
```

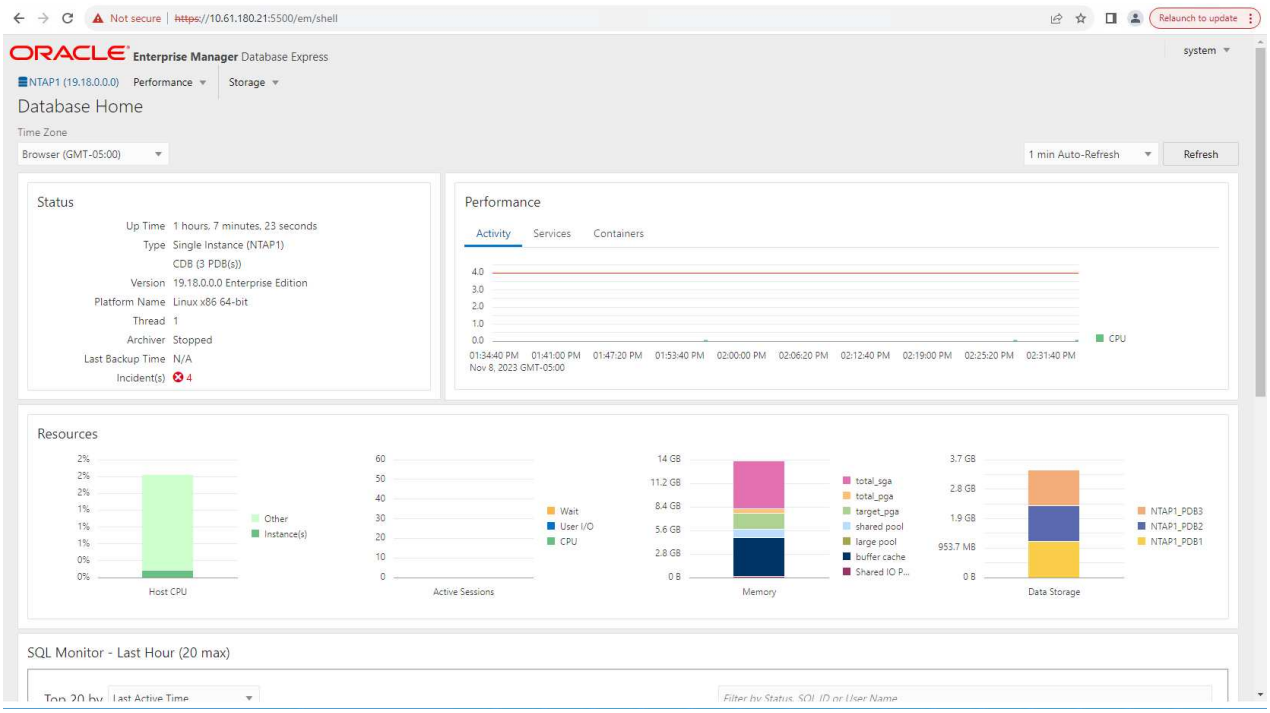
Ignore the `Not All Endpoints Registered` in State details. This results from a conflict of manual and dynamic database registration with the listener and can be safely ignored.

2. Validate ASM filter driver is working as expected.

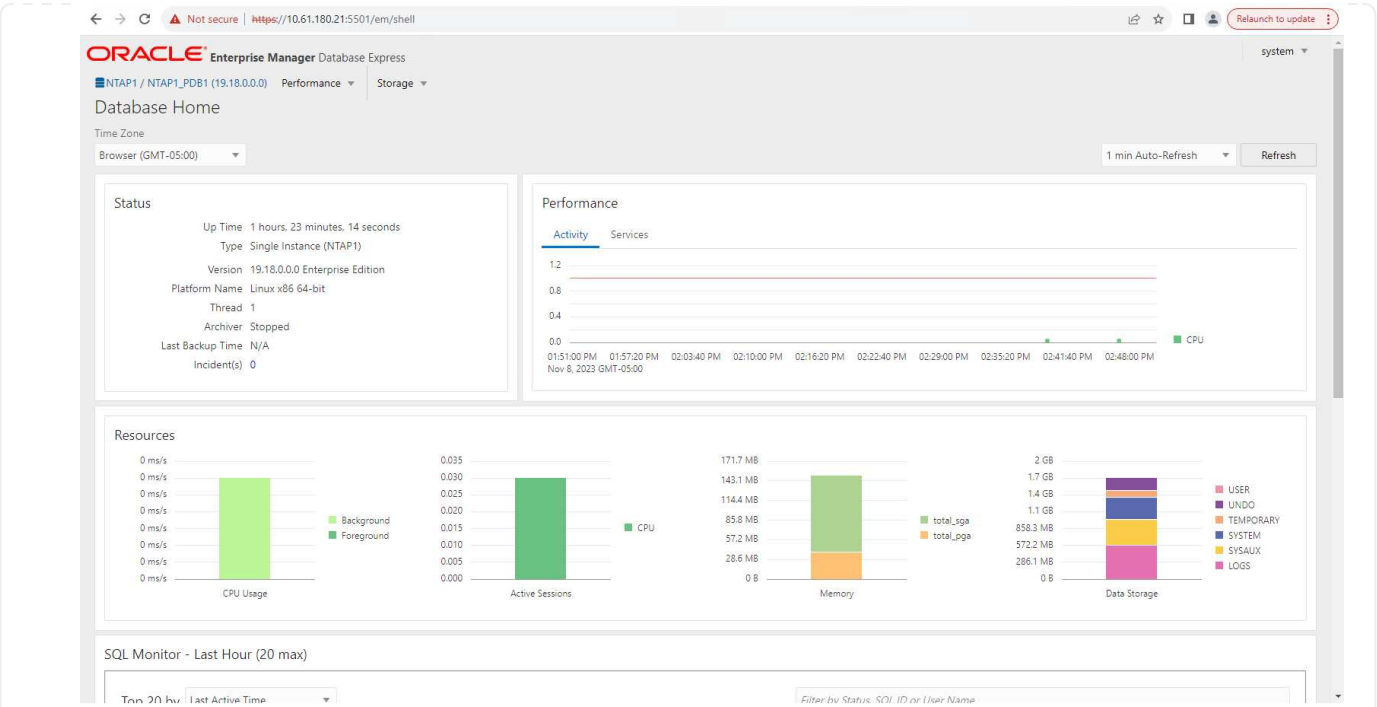
```
[oracle@ora_01 ~]$ asmcmd
ASMCMD> lsdg
State    Type    Rebal  Sector  Logical_Sector  Block       AU
Total_MB  Free_MB  Req_mir_free_MB  Usable_file_MB  Offline_disks
Voting_files  Name
MOUNTED  EXTERN  N         512             512   4096  4194304
327680   318644                0          318644              0
N  DATA/
MOUNTED  EXTERN  N         512             512   4096  4194304
81920    78880                0           78880              0
N  LOGS/
ASMCMD> lsdsk
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. Login to Oracle Enterprise Manager Express to validate database.


Not secure | https://10.61.180.21:5500/em/login
Relaunch to update
ORACLE ENTERPRISE MANAGER
DATABASE EXPRESS
Username
Password
Container Name
Log in
ORACLE
Copyright 2013, 2020, Oracle and/or its affiliates. All rights reserved.


Not secure | https://10.61.180.21:5500/em/shell
Relaunch to update
ORACLE Enterprise Manager Database Express
system
NTAP1 (19.18.0.0.0)
Performance
Storage
Database Home
Time Zone
Browser (GMT-05:00)
1 min Auto-Refresh
Refresh
Status
Up Time 1 hours, 7 minutes, 23 seconds
Type Single Instance (NTAP1)
CDB (3 PDB(s))
Version 19.18.0.0.0 Enterprise Edition
Platform Name Linux x86 64-bit
Thread 1
Archiver Stopped
Last Backup Time N/A
Incident(s) 4
Performance
Activity
Services
Containers
CPU
Resources
Host CPU
Other
Instance(s)
Active Sessions
Wait
User I/O
CPU
Memory
total_sga
total_pga
target_pga
shared pool
large pool
buffer cache
Shared IO P...
Data Storage
NTAP1_PDB3
NTAP1_PDB2
NTAP1_PDB1
SQL Monitor - Last Hour (20 max)
Top 20 by
Last Active Time
Filter by Status, SQL ID or User Name

```
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.GETHTTPSPORT()
------------------------------
                             0

SQL> exec DBMS_XDB_CONFIG.SETHTTPSPORT(5501);

PL/SQL procedure successfully completed.

SQL> select dbms_xdb_config.gethttpsport() from dual;

DBMS_XDB_CONFIG.GETHTTPSPORT()
------------------------------
                          5501

login to NTAP1_PDB1 from port 5501.
```



Oracle backup, restore, and clone with SnapCenter

Refer to TR-4979 Simplified, self-managed Oracle in VMware Cloud on AWS with guest-mounted FSx ONTAP section `Oracle backup, restore, and clone with SnapCenter` for details on setting up SnapCenter and executing the database backup, restore, and clone workflows.

Where to find additional information

To learn more about the information described in this document, review the following documents and/or websites:

- NETAPP ASA: ALL-FLASH SAN ARRAY

  https://www.netapp.com/data-storage/all-flash-san-storage-array/

- Installing Oracle Grid Infrastructure for a Standalone Server with a New Database Installation

  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

- Installing and Configuring Oracle Database Using Response Files

  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

- Use Red Hat Enterprise Linux 8.2 with ONTAP

  https://docs.netapp.com/us-en/ontap-sanhost/hu_rhel_82.html#all-san-array-configurations

Copyright information

Trademark information