



内部部署/混合云 NetApp Solutions

NetApp
May 17, 2024

目录

- 内部部署/混合云 1
 - TR-4983：《在采用iSCSI的NetApp ASA上简化的自动化Oracle部署》 1
 - NVA-1155：《基于FlexPod 数据中心且采用Cisco UCS和基于FC的NetApp AFF A800的Oracle 19c RAC数据库—设计和部署指南》 17
 - TR-4250：SAP与基于UNIX的Oracle和NFS以及适用于SAP 3.4的NetApp集群模式Data ONTAP和SnapManager 17
 - 部署 Oracle 数据库 17
 - 解决方案概述 38
 - TR-4794：《基于NetApp EF系列的Oracle数据库》 60

内部部署/混合云

TR-4983：《在采用iSCSI的NetApp ASA上简化的自动化Oracle部署》

NetApp公司Allen Cao、Niyaz Mohamed

本解决方案提供了有关在NetApp ASA阵列中自动部署和保护Oracle的概述和详细信息、这些阵列使用iSCSI协议作为主数据库存储、而Oracle数据库则使用ASM作为卷管理器在独立重新启动时进行配置。

目的

NetApp ASA系统为您的SAN基础架构提供现代化的解决方案。它们可以大规模简化并帮助您加快数据库等业务关键型应用程序的运行速度、确保数据始终可用(99.9999%的正常运行时间)、并降低TCO和碳排放量。NetApp ASA系统包括专为要求性能最苛刻的应用程序而设计的A系列型号和针对经济高效的大容量部署而优化的C系列型号。ASA A系列和C系列系统相结合、可提供卓越的性能、以改善客户体验并缩短取得成果的时间、保持业务关键型数据可用、受到保护和安全、并通过行业最有效的担保为任何工作负载提供更有效的容量。

本文档演示了如何在使用ASA系统构建的SAN环境中使用Andsilp自动化简化Oracle数据库的部署。Oracle数据库部署在独立重新启动配置中、使用iSCSI协议在ASA存储阵列上进行数据访问、并使用Oracle ASM进行数据库磁盘管理。此外、还提供了有关使用NetApp SnapCenter UI工具在NetApp ASA系统中执行高效存储数据库操作的Oracle数据库备份、还原和克隆的信息。

此解决方案 可解决以下使用情形：

- 在NetApp ASA系统中自动部署Oracle数据库作为主数据库存储
- 使用NetApp SnapCenter工具在NetApp ASA系统中进行Oracle数据库备份和还原
- 使用NetApp SnapCenter工具在NetApp ASA系统中为开发/测试或其他使用情形创建Oracle数据库克隆

audience

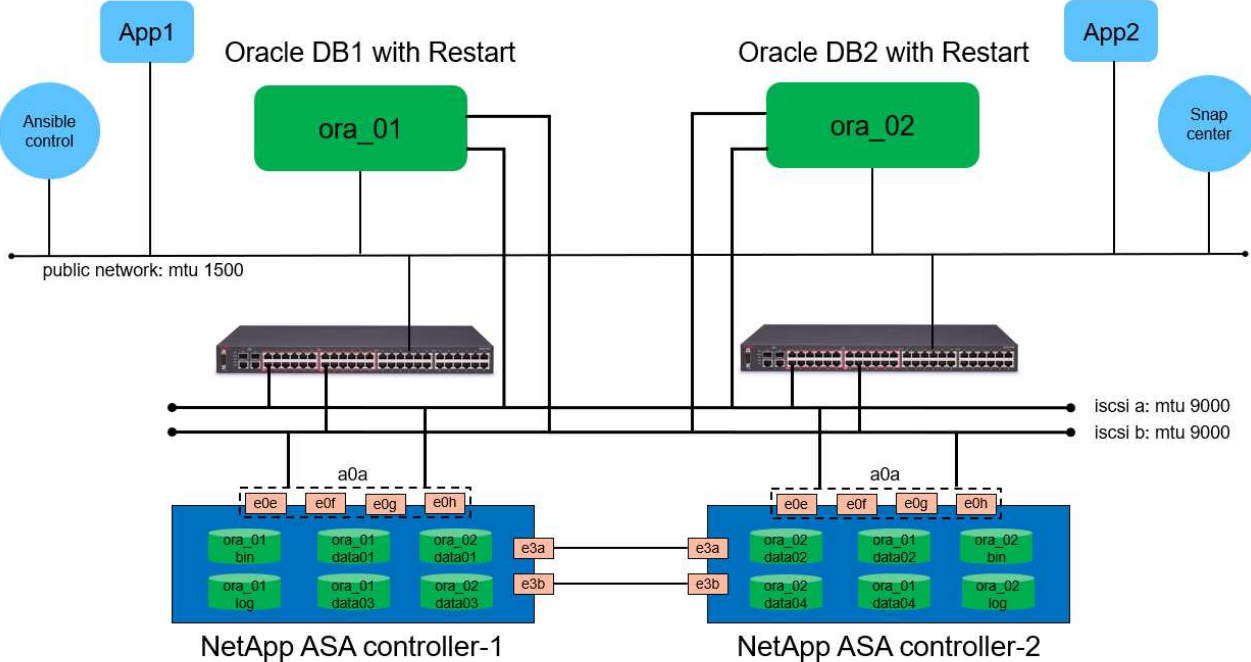
此解决方案 适用于以下人员：

- 希望在NetApp ASA系统中部署Oracle的BA。
- 希望在NetApp ASA系统中测试Oracle工作负载的数据库解决方案架构师。
- 希望在NetApp ASA系统上部署和管理Oracle数据库的存储管理员。
- 希望在NetApp ASA系统中建立Oracle数据库的应用程序所有者。

解决方案 测试和验证环境

此解决方案的测试和验证是在实验室环境中执行的、可能与最终部署环境不匹配。请参见一节 [\[部署注意事项的关键因素\]](#) 有关详细信息 ...

Simplified, Automated Oracle Database Deployment on NetApp ASA with iSCSI



硬件和软件组件

* 硬件 *		
NetApp ASA A400	版本9.13.1P1	2个NS224磁盘架、48个NVMe AFF驱动器、总容量为69.3 TiB
UCSB-B200-M4	Intel ® Xeon ® CPU E5-2690 v4 @ 2.60GHz	4节点VMware ESXi集群
软件		
RedHat Linux	reli-8.6、4.18.0-372.9.1.el8.x86_64内核	已部署RedHat订阅以进行测试
Windows服务器	2022标准版、10.0.20348内部版本20348	托管SnapCenter服务器
Oracle网格基础架构	版本19.18	已应用RU修补程序p34762026_190000_Linux-x86-64.zip
Oracle 数据库	版本19.18	已应用RU修补程序p34765931_190000_Linux-x86-64.zip
Oracle OPatch	版本12.2.0.1.36	最新修补程序p6880880_190000_Linux-x86-64.zip
SnapCenter 服务器	版本4.9P1	工作组部署

VMware vSphere 虚拟机管理程序	版本6.5.0.0000	VMware Tools、版本：11365 - Linux、1352 - Windows
打开JDK	版本java-1.8.0-OpenJDK.x86_64	DB VM上的SnapCenter插件要求

实验室环境中的Oracle数据库配置

* 服务器 *	* 数据库 *	DB存储
ORA_01	NTAP1 (NTAP1_PDB1、NTAP1_PDB2、NTAP_PDB3)	ASA A400上的iSCSI LUN
ORA_02	NTAP2 (NTAP2_PDB1、NTAP2_PDB2、NTAP2_PDB3)	ASA A400上的iSCSI LUN

部署注意事项的关键因素

- *Oracle数据库存储布局。*在此自动化Oracle部署中、我们默认配置四个数据库卷来托管Oracle二进制文件、数据和日志。然后、我们将使用数据和日志LUN创建两个ASM磁盘组。在+data ASM磁盘组中、我们会在每个ASA A400集群节点上的一个卷中配置两个数据LUN。在+logs ASM磁盘组中、我们会在一个ASA A400节点的日志卷中创建两个LUN。一般来说、在一个ONTAP卷中部署多个LUN可提高性能。
- *部署多个数据库服务器。*自动化解决方案可以在一次运行的AnsiblePlaybook中将一个Oracle容器数据库部署到多个数据库服务器。无论数据库服务器的数量如何、该操作手册的执行都保持不变。在部署多数据库服务器时、该操作手册会使用一种算法来构建数据库LUN、以便以最佳方式将其放置在ASA A400的双控制器上。控制器1上服务器主机索引位置中奇数数据库服务器的二进制和日志LUN。控制器2上服务器主机索引位置中偶数数据库服务器的二进制文件和日志LUN。数据库数据LUN平均分布到两个控制器。Oracle ASM将两个控制器上的数据LUN组合成一个ASM磁盘组、以充分利用这两个控制器的处理能力。
- *iSCSI配置。*数据库VM使用iSCSI协议连接到ASA存储以进行存储访问。您应在每个控制器节点上配置双路径以实现冗余、并在数据库服务器上设置iSCSI多路径以实现多路径存储访问。在存储网络上启用巨型帧、以最大程度地提高性能和吞吐量。
- *Oracle ASM冗余级别用于创建的每个Oracle ASM磁盘组。*由于ASA A400会在RAID DP中配置存储、以便在集群磁盘级别进行数据保护、因此应使用 External Redundancy，表示选项不允许Oracle ASM镜像磁盘组的内容。
- 数据库备份。NetApp提供了一个SnapCenter软件套件、可通过用户友好的用户界面进行数据库备份、还原和克隆。NetApp建议实施此类管理工具、以实现快速(不到一分钟)的快照备份、快速(几分钟)的数据库还原和数据库克隆。

解决方案 部署

以下各节提供了在NetApp ASA A400中使用直接挂载的数据库LUN通过iSCSI自动部署和保护Oracle 19c的分步过程、并将其部署到单节点中的数据库VM使用Oracle ASM作为数据库卷管理器的重新启动配置。

部署的前提条件

部署需要满足以下前提条件。

1. 假定已安装并配置NetApp ASA存储阵列。这包括两个控制器节点上的iSCSI广播域、LACP接口组a0a、两个控制器节点上的iSCSI <iscsi-a-vlan-id>端口(a0a-lacp-<iscsi-b-vlan-id>)。以下链接提供了详细的分步说明(如果需要帮助)。 ["详细指南—ASA A400"](#)
2. 将Linux VM配置为安装了最新版本的Ansv近 和Git的Ansv可 控制器节点。有关详细信息、请参见以下链接: ["NetApp解决方案 自动化入门"](#) 在第-节中 Setup the Ansible Control Node for CLI deployments on RHEL / CentOS 或 Setup the Ansible Control Node for CLI deployments on Ubuntu / Debian。
3. 克隆一份适用于iSCSI的NetApp Oracle部署自动化工具包副本。

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

4. 配置Windows服务器以使用最新版本运行NetApp SnapCenter UI工具。有关详细信息、请参见以下链接: ["安装 SnapCenter 服务器"](#)
5. 构建两个RHEL Oracle数据库服务器、可以是裸机VM、也可以是虚拟化VM。在不具有密码权限的sudo数据库服务器上创建一个管理员用户、并在Andsle主机和Oracle数据库服务器主机之间启用SSH专用/公共密钥身份验证。DB服务器/tmp/archive目录上的Oracle 19c安装文件后的阶段。

```
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"
```



请确保已在Oracle VM根卷中至少分配50G、以便有足够的空间来暂存Oracle安装文件。

6. 观看以下视频:

[借助iSCSI在NetApp ASA上简化和自动化Oracle部署](#)

自动化参数文件

Ans可 通过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、有三个用户定义参数文件需要用户输入才能执行操作手册。

- 主机—定义运行自动化操作手册的目标。
- vars/vars.yml—用于定义应用于所有目标的变量的全局变量文件。
- host_vars/host_name.yml—用于定义仅适用于本地目标的变量的本地变量文件。在我们的使用情形中、这些是Oracle数据库服务器。

除了这些用户定义的变量文件之外、还有多个默认变量文件包含默认参数、除非必要、否则不需要更改这些参数。以下各节说明了如何配置用户定义的变量文件。

参数文件配置

1. 可逆目标 hosts 文件配置:

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

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

2. 全局 vars/vars.yml 文件配置

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

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

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

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

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

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



```

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

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

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

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

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

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

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

```

3. 本地数据库服务器 host_vars/host_name.yml 配置

```
# 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
```

执行操作手册

自动化工具包中共有六本操作手册。每个任务执行不同的任务块、并用于不同的用途。

```
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.
```

使用以下命令可通过三个选项运行这些操作手册。

1. 一次运行即可执行所有部署操作手册。

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

2. 使用1-4的数字顺序执行一次一个操作手册。

```
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. 使用标记执行0-all_playbook.yml。

```
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. 撤消环境

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

执行后验证

运行该操作手册后、以Oracle用户身份登录到Oracle数据库服务器、以验证是否已成功创建Oracle网格基础架构和数据库。以下是在主机ora_01上验证Oracle数据库的示例。

1. 验证创建的网格基础架构和资源。

```
[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 ~]$

```



忽略 Not All Endpoints Registered 在状态详细信息中。这是由于在侦听器中手动和动态数据库注册发生冲突而导致的、可以放心地忽略。

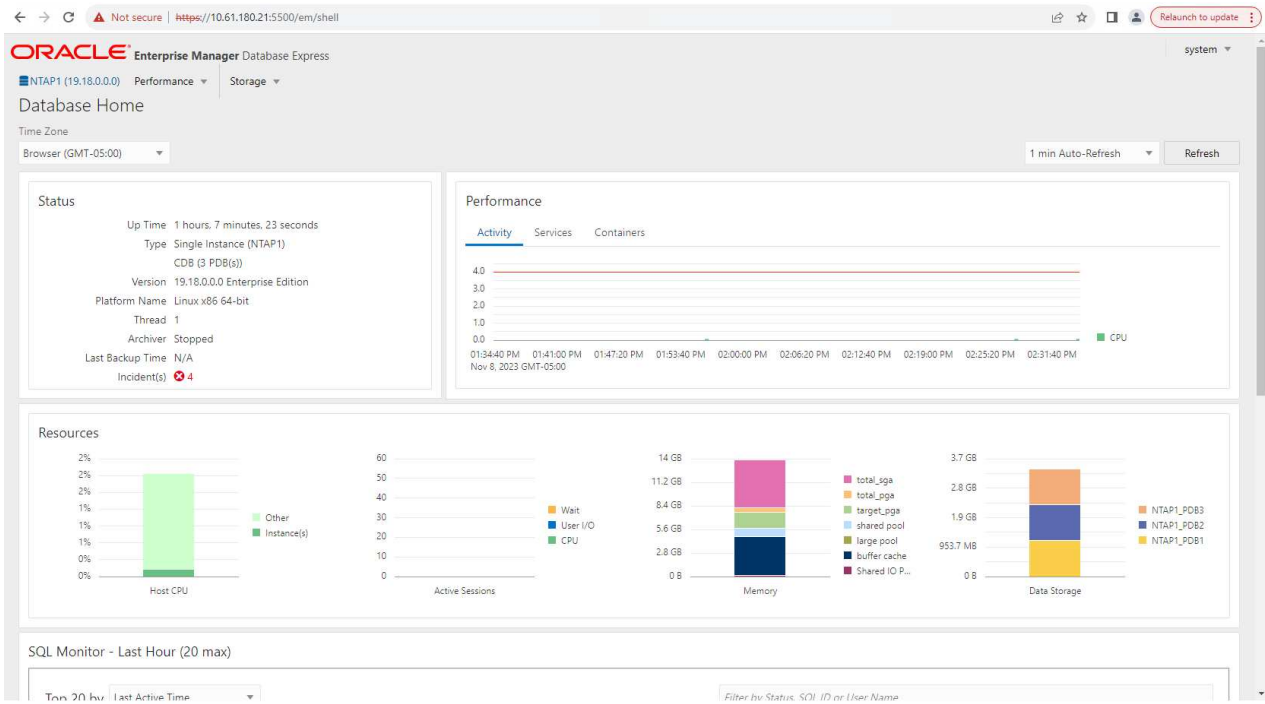
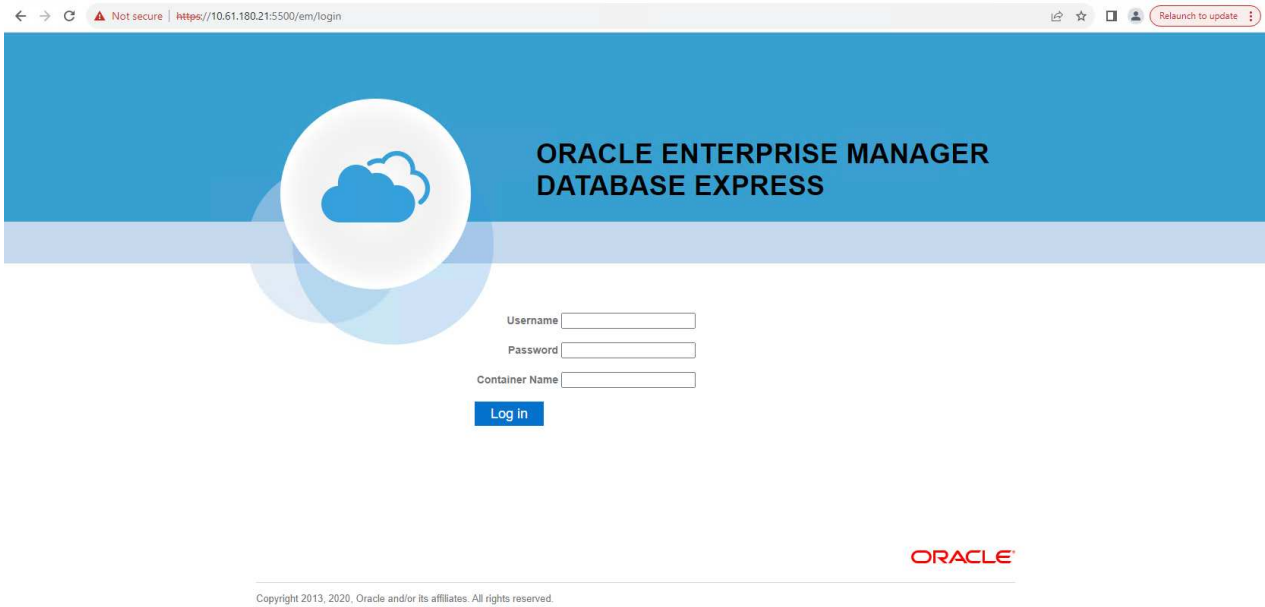
2. 验证ASM筛选器驱动程序是否按预期工作。

```

[oracle@ora_01 ~]$ asmcmd
ASMCMD> lsdg
State      Type      Rebal  Sector  Logical_Sector  Block      AU
Total_MB   Free_MB   Req_mir_free_MB  Usable_file_MB  Offline_disks
Voting_files  Name
MOUNTED    EXTERN    N      512      512      4096      4194304
327680     318644      0      318644      0
N  DATA/
MOUNTED    EXTERN    N      512      512      4096      4194304
81920      78880      0      78880      0
N  LOGS/
ASMCMD> lsdk
Path
AFD:ORA_01_DAT1_01
AFD:ORA_01_DAT1_03
AFD:ORA_01_DAT1_05
AFD:ORA_01_DAT1_07
AFD:ORA_01_DAT2_02
AFD:ORA_01_DAT2_04
AFD:ORA_01_DAT2_06
AFD:ORA_01_DAT2_08
AFD:ORA_01_LOGS_01
AFD:ORA_01_LOGS_02
ASMCMD> afd_state
ASMCMD-9526: The AFD state is 'LOADED' and filtering is 'ENABLED' on
host 'ora_01'
ASMCMD>

```

3. 登录到Oracle Enterprise Manager Express以验证数据库。



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
```

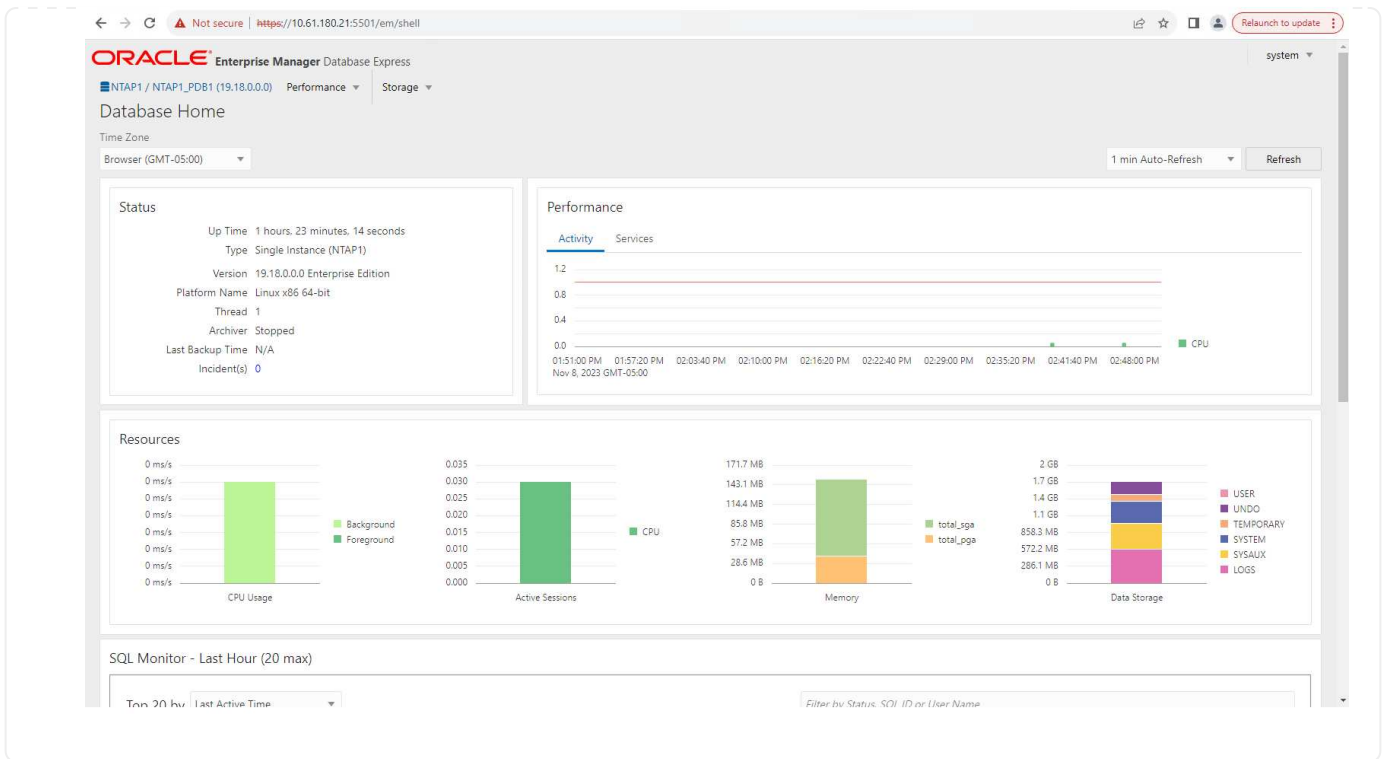
```
SQL> exec DBMS_XDB_CONFIG.SETHTTPSPORT(5501);
```

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.



使用SnapCenter进行Oracle备份、还原和克隆

请参阅TR-4979 "借助子系统装载的FSx ONTAP、在基于AWS的VMware Cloud中简化自我管理Oracle" 部分。Oracle backup, restore, and clone with SnapCenter 有关设置SnapCenter以及执行数据库备份、还原和克隆工作流的详细信息。

从何处查找追加信息

要了解有关本文档中所述信息的更多信息，请查看以下文档和 / 或网站：

- NetApp ASA：全闪存SAN阵列

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

- 在安装新数据库的情况下为独立服务器安装Oracle网格基础架构

["https://docs.oracle.com/en/database/oracle/oracle-database/19/ladbi/installing-oracle-grid-infrastructure-for-a-standalone-server-with-a-new-database-installation.html#GUID-0B1CEE8C-C893-46AA-8A6A-7B5FAAEC72B3"](https://docs.oracle.com/en/database/oracle/oracle-database/19/ladbi/installing-oracle-grid-infrastructure-for-a-standalone-server-with-a-new-database-installation.html#GUID-0B1CEE8C-C893-46AA-8A6A-7B5FAAEC72B3)

- 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/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)

- 将Red Hat Enterprise Linux 8.2与ONTAP结合使用

["https://docs.netapp.com/us-en/ontap-sanhost/hu_rhel_82.html#all-san-array-configurations"](https://docs.netapp.com/us-en/ontap-sanhost/hu_rhel_82.html#all-san-array-configurations)

NVA-1155: 《基于FlexPod 数据中心且采用Cisco UCS和基于FC的NetApp AFF A800的Oracle 19c RAC数据库—设计和部署指南》

NetApp公司Allen Cao

本《基于Cisco UCS的FlexPod Datacenter上的Oracle 19c RAC数据库设计和部署指南》以及《基于FC的NetApp AFF A800》详细介绍了在采用Oracle Linux 8.2的最新FlexPod Datacenter基础架构上托管Oracle RAC数据库的解决方案 设计以及分步部署过程操作系统和Red Hat兼容内核。

"NVA-1155: FlexPod Datacenter上的Oracle 19c RAC数据库、采用Cisco UCS和基于FC的NetApp AFF A800"

TR-4250: SAP与基于UNIX的Oracle和NFS以及适用于SAP 3.4的NetApp集群模式Data ONTAP 和SnapManager

NetApp 公司 Nil Bauser

TR-4250解决了设计存储解决方案以使用Oracle数据库支持SAP业务套件产品所面临的挑战。本文档的主要重点是使用最新一代 SAP 解决方案的业务和 IT 主管面临的常见存储基础架构设计，部署，操作和管理挑战。本文档中的建议是通用的；它们并不特定于 SAP 应用程序或 SAP 实施的规模和范围。TR-4250假定读者已基本了解NetApp和SAP产品的技术和操作。TR-4250是在NetApp、SAP、Oracle和我们的客户的技术人员互动的基础上开发的。

"TR-4250: SAP与基于UNIX的Oracle和NFS以及适用于SAP 3.4的NetApp集群模式Data ONTAP和SnapManager"

部署 Oracle 数据库

解决方案概述

此页面介绍了在 NetApp ONTAP 存储上部署 Oracle19c 的自动化方法。

在 NFS 上自动部署适用于 ONTAP 的 Oracle19c

企业正在对其环境进行自动化，以提高效率，加快部署速度并减少手动操作。Ansible 等配置管理工具正在用于简化企业数据库操作。在此解决方案中，我们将演示如何使用 Ansible 通过 NetApp ONTAP 自动配置和配置 Oracle 19c。通过使存储管理员，系统管理员和 DBA 能够一致快速地部署新存储，配置数据库服务器并安装 Oracle 19c 软件，您可以获得以下优势：

- 消除设计复杂性和人为错误，并实施可重复的一致部署和最佳实践
- 缩短配置存储，配置数据库主机和安装 Oracle 的时间
- 提高数据库管理员，系统和存储管理员的工作效率
- 轻松扩展存储和数据库

NetApp 为客户提供经过验证的 Ansible 模块和角色，以加快 Oracle 数据库环境的部署，配置和生命周期管理。此解决方案提供了相关说明和 Ansible 攻略手册代码，可帮助您：

- 为 Oracle 数据库创建和配置 ONTAP NFS 存储
- 在 RedHat Enterprise Linux 7/8 或 Oracle Linux 7/8 上安装 Oracle 19c
- 在 ONTAP NFS 存储上配置 Oracle 19c

有关更多详细信息或要开始，请参见下面的概述视频。

AWX/ 塔式部署

第 1 部分：入门，要求，自动化详细信息和初始 AWX/ 塔式配置

[AWX部署](#)

第 2 部分：变量和运行攻略手册

[AWX操作手册运行](#)

CLI 部署

第 1 部分：入门，要求，自动化详细信息和可变控制主机设置

[CLI 部署](#)

第 2 部分：变量和运行攻略手册

[命令行界面操作手册运行](#)

入门

此解决方案可以在 AWX/Tower 环境中运行，也可以通过命令行界面在 Ansible 控制主机上运行。

AWX/ 塔式

对于 AWX/Tower 环境，系统将引导您创建 ONTAP 集群管理和 Oracle 服务器（IP 和主机名）清单，创建凭据，配置从 NetApp Automation Github 提取 Ansible 代码的项目以及启动自动化的作业模板。

1. 填写特定于您的环境的变量，然后将其复制并粘贴到作业模板中的额外 VAR 字段中。
2. 将额外的变量添加到作业模板后，您可以启动自动化。
3. 通过为 `ontap_config`，`linux_config` 和 `oracle_config` 指定标记，作业模板将分三个阶段运行。

通过 Ansible 控制主机执行 CLI

1. 配置 Linux 主机，以使用作 Ansible 控制主机"[单击此处了解详细说明](#)"
2. 配置 Ansible 控制主机后，您可以克隆 Ansible Automation 存储库。
3. 使用 ONTAP 集群管理的 IP 和 / 或主机名以及 Oracle 服务器的管理 IP 编辑 hosts 文件。

4. 填写特定于您的环境的变量，然后将其复制并粘贴到 `vars.yml` 文件中。
5. 每个 Oracle 主机都有一个可变文件，该文件由其主机名标识，其中包含主机专用变量。
6. 完成所有可变文件后，您可以通过为 `ontap_config`，`linux_config` 和 `oracle_config` 指定标记来分三个阶段运行此攻略手册。

要求

environment	要求
* 可逆环境 *	AWX/Tower 或 Linux 主机作为 Ansible 控制主机
	Ansible v.10 及更高版本
	Python 3.
	Python 库— netapp-lib — xmltodict — jMespath
* ONTAP *	ONTAP 版本 9.3 - 9.7
	两个数据聚合
	已创建 NFS VLAN 和 ifgrp
* Oracle 服务器 *	RHEL 7/8
	Oracle Linux 7/8
	NFS ，公有和可选管理的网络接口
	Oracle 服务器上的 Oracle 安装文件

自动化详细信息

此自动化部署采用一本 Ansible 攻略手册设计，该攻略手册包含三个不同的角色。这些角色适用于 ONTAP ，Linux 和 Oracle 配置。下表介绍了正在自动执行的任务。

Role	任务
* ontap_config*	预检查 ONTAP 环境
	为 Oracle 创建基于 NFS 的 SVM
	创建导出策略
	为 Oracle 创建卷
	创建 NFS LIF

Role	任务
* Linux 配置 *	创建挂载点并挂载 NFS 卷
	验证 NFS 挂载
	特定于操作系统的配置
	创建 Oracle 目录
	配置页面
	禁用 SELinux 和防火墙守护进程
	启用并启动 chronyd 服务
	增加文件描述符硬限制
	创建 pam 会话文件
* ORACLE_CONFIG *	Oracle 软件安装
	创建 Oracle 侦听器
	创建 Oracle 数据库
	Oracle 环境配置
	保存 PDB 状态
	启用实例归档模式
	启用 DNFS 客户端
	在操作系统重新启动之间启用数据库自动启动和关闭

默认参数

为了简化自动化，我们已使用默认值预设了许多必需的 Oracle 部署参数。通常，无需更改大多数部署的默认参数。更高级的用户可以谨慎地更改默认参数。默认参数位于每个角色文件夹的默认目录下。

部署说明

开始之前，请下载以下 Oracle 安装和修补程序文件，并将其放置在 `/tmp/archive` 目录中，以便每个要部署的数据库服务器上的所有用户都能进行读取，写入和执行访问。自动化任务会在该特定目录中查找命名的安装文件，以便进行 Oracle 安装和配置。

```
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
```

许可证

您应按照 Github 存储库中的说明读取许可证信息。访问，下载，安装或使用此存储库中的内容即表示您同意所规定的许可证条款 ["此处"](#)。

请注意，在生成和 / 或共享任何派生作品时，此存储库中的内容存在一定的限制。请务必阅读的条款 ["许可证"](#) 在使用内容之前。如果您不同意所有条款，请勿访问，下载或使用此存储库中的内容。

准备就绪后，单击 ["此处可查看 AWX/ 塔式部署的详细步骤"](#) 或 ["此处用于命令行界面部署"](#)。

分步部署操作步骤

此页面介绍了在 NetApp ONTAP 存储上部署 Oracle19c 的自动化方法。

AWX/Tower 部署 Oracle 19c 数据库

1. 为您的环境创建清单，组，主机和凭据

本节介绍如何在 AWX/Ansible 塔中设置清单，组，主机和访问凭据，以便为使用 NetApp 自动化解决方案的环境做好准备。

1. 配置清单。

- a. 导航到资源→清单→添加，然后单击添加清单。
- b. 提供名称和组织详细信息，然后单击保存。
- c. 在清单页面上，单击已创建的清单。
- d. 如果存在任何清单变量，请将其粘贴到变量字段中。
- e. 导航到组子菜单，然后单击添加。
- f. 提供 ONTAP 的组名称，粘贴组变量（如果有），然后单击保存。
- g. 对 Oracle 的另一个组重复此过程。
- h. 选择已创建的 ONTAP 组，转到主机子菜单，然后单击添加新主机。
- i. 提供 ONTAP 集群管理 IP 的 IP 地址，粘贴主机变量（如果有），然后单击保存。
- j. 必须对 Oracle 组和 Oracle 主机管理 IP/ 主机名重复此过程。

2. 创建凭据类型。对于涉及 ONTAP 的解决方案，您必须配置凭据类型以匹配用户名和密码条目。

- a. 导航到管理→凭据类型，然后单击添加。
- b. 提供名称和问题描述。
- c. 将以下内容粘贴到输入配置中：

```
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. 将以下内容粘贴到注射器配置中：

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

1. 配置凭据。

- 导航到资源→凭据，然后单击添加。
- 输入 ONTAP 的名称和组织详细信息。
- 选择为 ONTAP 创建的自定义凭据类型。
- 在 Type Details 下，输入用户名，密码和 vsadmin_password。
- 单击 Back to Credential，然后单击 Add。
- 输入 Oracle 的名称和组织详细信息。
- 选择计算机凭据类型。
- 在 Type Details 下，输入 Oracle 主机的 Username 和 Password。
- 选择正确的权限升级方法，然后输入用户名和密码。

2. 创建项目

- 转至 "Resources" → "projects"，然后单击 "Add"。
 - 输入名称和组织详细信息。
 - 在源控制凭据类型字段中选择 Git。
 - 输入 ... <https://github.com/NetApp-Automation/na_oracle19c_deploy.git> 作为源控制 URL。
 - 单击保存。
 - 当源代码发生更改时，项目可能偶尔需要同步。

3. 配置 Oracle host_vars

本节中定义的变量将应用于每个 Oracle 服务器和数据库。

- 以以下嵌入式 Oracle hosts 变量或 host_vars 形式输入环境特定的参数。



必须根据您的环境更改蓝色项。

主机 VARS 配置

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



```

# Add your Oracle Host
ansible_host: "10.61.180.15"

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

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

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

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

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

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

```

```
lif: "172.21.94.200", size: "25"}
```

1. 填写蓝色字段中的所有变量。
2. 完成变量输入后，单击表单上的复制按钮以复制要传输到 AWX 或塔式的所有变量。
3. 导航回 AWX 或塔式，然后转到资源→主机，选择并打开 Oracle 服务器配置页面。
4. 在详细信息选项卡下，单击编辑并将从步骤 1 复制的变量粘贴到 YAML 选项卡下的变量字段。
5. 单击保存。
6. 对系统中的任何其他 Oracle 服务器重复此过程。

4. 配置全局变量

本节中定义的变量适用于所有 Oracle 主机，数据库和 ONTAP 集群。

1. 在以下嵌入式全局变量或变量表单中输入环境特定的参数。



必须根据您的环境更改蓝色项。

```
#####
##### 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. 在蓝色字段中填写所有变量。
2. 完成变量输入后，单击表单上的复制按钮，将要传输到 AWX 或塔式的所有变量复制到以下作业模板中。

5. 配置并启动作业模板。

1. 创建作业模板。

- a. 导航到资源→模板→添加，然后单击添加作业模板。
- b. 输入名称和问题描述
- c. 选择作业类型；运行将根据攻略手册配置系统，而检查将在不实际配置系统的情况下执行攻略手册的试运行。
- d. 为攻略手册选择相应的清单，项目，攻略手册和凭据。
- e. 选择 all_playbook.yml 作为要执行的默认攻略手册。
- f. 将从步骤 4 复制的全局变量粘贴到 YAML 选项卡下的模板变量字段中。
- g. 选中作业标记字段中的启动时提示框。
- h. 单击保存。

2. 启动作业模板。

- a. 导航到资源→模板。
- b. 单击所需模板，然后单击启动。
- c. 在启动作业标记时系统提示时，键入 requirements_config。您可能需要单击 requirements_config 下方的 Create Job Tag 行以输入作业标记。



requirements_config 可确保您有正确的库来运行其他角色。

1. 单击下一步，然后单击启动以启动作业。
2. 单击查看→作业以监控作业输出和进度。
3. 在启动作业标记时，系统提示您键入 ontap_config。您可能需要单击 ontap_config 下方的 Create "Job Tag (创建作业标记)" 行以输入作业标记。
4. 单击下一步，然后单击启动以启动作业。
5. 单击查看→作业以监控作业输出和进度
6. 完成 ontap_config 角色后，对 linux_config 再次运行此过程。
7. 导航到资源→模板。
8. 选择所需模板，然后单击启动。
9. 在 Linux_config 中启动作业标记类型时，系统提示您选择 Linux_config 下方的创建 "作业标记" 行以输入作业标记。
10. 单击下一步，然后单击启动以启动作业。
11. 选择 View → Jobs 以监控作业输出和进度。
12. 完成 Linux_config 角色后，对 ORACLE_CONFIG 再次运行此过程。
13. 转至资源→模板。
14. 选择所需模板，然后单击启动。
15. 在启动作业标记时，系统提示您键入 ORACLE_CONFIG。您可能需要选择 ORACLE_CONFIG 下方的 "Create Job Tag (创建作业标记)" 行以输入作业标记。

16. 单击下一步，然后单击启动以启动作业。

17. 选择 View → Jobs 以监控作业输出和进度。

6. 在同一 Oracle 主机上部署其他数据库

此攻略手册的 Oracle 部分会每次在 Oracle 服务器上创建一个 Oracle 容器数据库。要在同一服务器上创建其他容器数据库，请完成以下步骤。

1. 修改 host_vars 变量。
 - a. 返回到步骤 2 —配置 Oracle host_vars 。
 - b. 将 Oracle SID 更改为其他命名字符串。
 - c. 将侦听器端口更改为其他编号。
 - d. 如果要安装 EM Express ，请将 EM Express 端口更改为其他编号。
 - e. 将修订后的主机变量复制并粘贴到主机配置详细信息选项卡中的 Oracle 主机变量字段中。
2. 启动仅包含 ORACLE_CONFIG 标记的部署作业模板。
3. 以Oracle用户身份登录到Oracle服务器并执行以下命令：

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



如果安装按预期完成且 Oracle 数据库已启动，则此列表将列出 Oracle 进程

4. 登录到数据库以检查数据库配置设置以及使用以下命令集创建的PDB。

```
[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
```

这可确认DNFS运行正常。

5. 使用以下命令通过侦听器连接到数据库以检查Oracle侦听器配置。更改为相应的侦听器端口和数据库服务名称。

```
[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
```

这样可以确认Oracle侦听器工作正常。

如何获取帮助？

如果您需要有关该工具包的帮助，请加入 ["NetApp 解决方案自动化社区支持 Slack 通道"](#) 并寻找解决方案自动化渠道来发布您的问题或询问。

分步部署操作步骤

本文档详细介绍了如何使用自动化命令行界面(CLI)部署Oracle 19c。

CLI 部署 Oracle 19c 数据库

本节介绍使用 CLI 准备和部署 Oracle19c 数据库所需的步骤。确保您已查看 ["入门和要求部分"](#) 并相应地准备好您的环境。

下载 Oracle19c repo

1. 从您的Ans负责人 控制器中、运行以下命令：

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

2. 下载存储库后、将目录更改为na_oracle19c_Deploy <cd na_oracle19c_deploy>。

编辑 **hosts** 文件

在部署之前完成以下操作：

- 1. 编辑 hosts 文件 na_oracle19c_deploy 目录。
- 2. 在 ONTAP 下，将 IP 地址更改为集群管理 IP 。
- 3. 在 Oracle 组下，添加 Oracle 主机名称。主机名必须通过 DNS 或 hosts 文件解析为其 IP 地址，或者必须在主机中指定。
- 4. 完成这些步骤后，请保存所做的任何更改。

以下示例说明了一个主机文件：

```
#ONTAP Host

[ontap]

"10.61.184.183"

#Oracle hosts

[oracle]

"rtpora01"

"rtpora02"
```

此示例将执行该攻略手册，并在两个 Oracle DB 服务器上同时部署 Oracle 19c 。您也可以仅使用一个数据库服务器进行测试。在这种情况下，您只需要配置一个主机变量文件。



无论您部署多少 Oracle 主机和数据库，本攻略手册的执行方式都是相同的。

编辑 **host_vars** 下的 **host_name.yml** 文件

每个 Oracle 主机都有其主机变量文件，该文件由包含主机专用变量的主机名标识。您可以为主机指定任何名称。从 Host VARS Config 部分编辑并复制 host_vars ，然后将其粘贴到所需的 host_name.yml 文件中。



必须根据您的环境更改蓝色项。

主机 **VARS** 配置

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

# Add your Oracle Host
```

```

ansible_host: "10.61.180.15"

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

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

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

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

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

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

```

编辑 vars.yml 文件

vars.yml 文件整合了所有环境特定的变量（ONTAP，Linux 或 Oracle），用于 Oracle 部署。

1. 编辑并复制 VARS 部分中的变量，然后将这些变量粘贴到 vars.yml 文件中。

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

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

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

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

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

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

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

#SVM name
```

```

svm_name: "ora_svm"

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

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

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

#NFS LIFs IP address and netmask

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

#NFS client match

client_match: "172.21.94.0/24"

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

#NFS Mount points for Oracle DB volumes

mount_points:

```

```

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

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

hugepages_nr: "1234"

# RedHat subscription username and password

redhat_sub_username: "xxx"
redhat_sub_password: "xxx"

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

db_domain: "your.domain.com"

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

initial_pwd_all: "netappl23"

```

运行攻略手册

完成所需的环境前提条件并将变量复制到 vars.yml 和 yor_host.yml 中后，您便可部署攻略手册了。



必须根据您的环境更改 < 用户名 >。

1. 通过传递正确的标记和ONTAP集群用户名来运行ONTAP操作手册。根据提示填写ONTAP集群的密码、然后填写vsadmin。

```

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

```

2. 运行Linux操作手册以执行Linux部署部分。输入admin ssh密码和sudo密码。

```

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

```

3. 运行Oracle操作手册以执行Oracle部署部分。输入admin ssh密码和sudo密码。

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

在同一 **Oracle** 主机上部署其他数据库

此攻略手册的 Oracle 部分会每次在 Oracle 服务器上创建一个 Oracle 容器数据库。要在同一服务器上创建其他容器数据库，请完成以下步骤：

1. 修改 host_vars 变量。
 - a. 返回到步骤 3 - 编辑 host_vars 下的 host_name.yml 文件。
 - b. 将 Oracle SID 更改为其他命名字符串。
 - c. 将侦听器端口更改为其他编号。
 - d. 如果已安装 EM Express，请将 EM Express 端口更改为其他编号。
 - e. 将修订后的主机变量复制并粘贴到 host_vars 下的 Oracle 主机变量文件中。
2. 使用 ORACLE_CONFIG 标记执行攻略手册，如中所示 [\[运行攻略手册\]](#)。

验证 **Oracle** 安装

1. 以Oracle用户身份登录到Oracle服务器并执行以下命令：

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



如果安装按预期完成且 Oracle 数据库已启动，则此列表将列出 Oracle 进程

2. 登录到数据库以检查数据库配置设置以及使用以下命令集创建的PDB。

```
[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
```

这可确认DNFS运行正常。

3. 使用以下命令通过侦听器连接到数据库以检查Oracle侦听器配置。更改为相应的侦听器端口和数据库服务名称。

```
[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
```

这样可以确认Oracle侦听器工作正常。

如何获取帮助？

如果您需要有关该工具包的帮助，请加入 ["NetApp 解决方案自动化社区支持 Slack 通道"](#) 并寻找解决方案自动化渠道来发布您的问题或询问。

解决方案概述

此页面介绍了在 NetApp ONTAP 存储上部署 Oracle19c 的自动化方法。

为 **Oracle** 数据库提供自动化数据保护

企业正在对其环境进行自动化，以提高效率，加快部署速度并减少手动操作。Ansible 等配置管理工具正在用于简化企业数据库操作。在本解决方案中，我们演示了如何使用 Ansible 通过 NetApp ONTAP 自动保护 Oracle 数据。通过使存储管理员，系统管理员和 DBA 能够一致快速地将数据复制到异地数据中心或公有云，您可以获得以下优势：

- 消除设计复杂性和人为错误，并实施可重复的一致部署和最佳实践
- 缩短配置集群间复制，CVO 实例化和恢复 Oracle 数据库的时间
- 提高数据库管理员，系统和存储管理员的工作效率
- 提供数据库恢复工作流，以便于测试灾难恢复场景。

NetApp 为客户提供经过验证的 Ansible 模块和角色，以加快 Oracle 数据库环境的部署，配置和生命周期管理。

此解决方案提供了相关说明和 Ansible 攻略手册代码，可帮助您：

在内部复制到内部复制

- 在源和目标上创建集群间 LIF
- 建立集群和 SVM 对等关系
- 创建并初始化 Oracle 卷的 SnapMirror
- 通过 AWX/Tower 为 Oracle 二进制文件，数据库和日志创建复制计划
- 还原目标上的 Oracle 数据库，并使数据库联机

在内部迁移到 AWS 中的 CVO

- 创建 AWS 连接器
- 在 AWS 中创建 CVO 实例
- 将内部集群添加到 Cloud Manager
- 在源上创建集群间 LIF
- 建立集群和 SVM 对等关系
- 创建并初始化 Oracle 卷的 SnapMirror
- 通过 AWX/Tower 为 Oracle 二进制文件，数据库和日志创建复制计划
- 还原目标上的 Oracle 数据库，并使数据库联机

准备就绪后，单击 ["请单击此处了解解决方案入门"](#)。

入门

此解决方案设计为在 AWX/ 塔式环境中运行。

AWX/ 塔式

对于 AWX/Tower 环境，系统将引导您创建 ONTAP 集群管理和 Oracle 服务器（IP 和主机名）清单，创建凭据，配置从 NetApp Automation Github 提取 Ansible 代码的项目以及启动自动化的作业模板。

1. 解决方案设计用于在私有云环境（内部到内部）和混合云（内部到公有 Cloud Cloud Volumes ONTAP （CVO））中运行
2. 填写特定于您的环境的变量，然后将其复制并粘贴到作业模板中的额外 VAR 字段中。
3. 将额外的变量添加到作业模板后，您可以启动自动化。
4. 此自动化设置为运行三个阶段（安装，Oracle 二进制文件复制计划，数据库，日志和仅适用于日志的复制计划），以及第四个阶段以在灾难恢复站点恢复数据库。
5. 有关获取 CVO 数据保护所需密钥和令牌的详细说明，请访问 ["收集 CVO 和连接器部署的前提条件"](#)

要求

<Strong class="bi"> 内部部署 |

environment	要求
* 可逆环境 *	AWX/ 塔式
	Ansible v.10 及更高版本
	Python 3.
	Python 库— netapp-lib — xmltodict — jMespath
* ONTAP *	ONTAP 版本 9.8 及更高版本
	两个数据聚合
	已创建 NFS VLAN 和 ifgrp
* Oracle 服务器 *	RHEL 7/8
	Oracle Linux 7/8
	NFS ， 公有和可选管理的网络接口
	源上的现有 Oracle 环境以及目标上的等效 Linux 操作系统（灾难恢复站点或公有云）

<strong class="bi">CVO

environment	要求
* 可逆环境 *	AWX/ 塔式
	Ansible v.10 及更高版本
	Python 3.
	Python 库— netapp-lib — xmltodict — jMespath
* ONTAP *	ONTAP 版本 9.8 及更高版本
	两个数据聚合
	已创建 NFS VLAN 和 ifgrp
* Oracle 服务器 *	RHEL 7/8
	Oracle Linux 7/8
	NFS ， 公有和可选管理的网络接口
	源上的现有 Oracle 环境以及目标上的等效 Linux 操作系统（灾难恢复站点或公有云）
	在 Oracle EC2 实例上设置适当的交换空间，默认情况下，某些 EC2 实例使用 0 交换部署
* Cloud Manager/AWS*	AWS 访问 / 机密密钥
	NetApp Cloud Manager 帐户
	NetApp Cloud Manager 刷新令牌

<Strong class="bi"> 内部部署 |

此自动化部署采用一本 Ansible 攻略手册设计，该攻略手册包含三个不同的角色。这些角色适用于 ONTAP，Linux 和 Oracle 配置。下表介绍了正在自动执行的任务。

攻略手册	任务
* ontap_setup*	预检查 ONTAP 环境
	在源集群上创建集群间 LIF （可选）
	在目标集群上创建集群间 LIF （可选）
	创建集群和 SVM 对等关系
	创建目标 SnapMirror 并初始化指定的 Oracle 卷
* 复制 _cg	在 /etc/oratab 中为每个数据库启用备份模式
	为 Oracle 二进制卷和数据库卷创建的快照
	已更新 SnapMirror
	关闭 /etc/oratab 中每个数据库的备份模式
* 。 ora_replication ; log*	切换 /etc/oratab 中每个数据库的当前日志
	为 Oracle 日志卷创建的快照
	已更新 SnapMirror
* ora_recovery*	中断 SnapMirror
	在目标上启用 NFS 并为 Oracle 卷创建接合路径
	配置 DR Oracle 主机
	挂载并验证 Oracle 卷
	恢复并启动 Oracle 数据库

<strong class="bi">CVO

此自动化部署采用一本 Ansible 攻略手册设计，该攻略手册包含三个不同的角色。这些角色适用于 ONTAP，Linux 和 Oracle 配置。下表介绍了正在自动执行的任务。

攻略手册	任务
* CVO_setup*	对环境进行预检查
	AWS 配置 /AWS 访问密钥 ID/ 机密密钥 / 默认区域
	创建 AWS 角色
	在 AWS 中创建 NetApp Cloud Manager Connector 实例
	在 AWS 中创建 Cloud Volumes ONTAP （ CVO ）实例
	将内部源 ONTAP 集群添加到 NetApp Cloud Manager 中
	创建目标 SnapMirror 并初始化指定的 Oracle 卷

攻略手册	任务
* 复制 _cg	在 /etc/oratab 中为每个数据库启用备份模式
	为 Oracle 二进制卷和数据库卷创建的快照
	已更新 SnapMirror
	关闭 /etc/oratab 中每个数据库的备份模式
* 。 ora_replication ; log*	切换 /etc/oratab 中每个数据库的当前日志
	为 Oracle 日志卷创建的快照
	已更新 SnapMirror
* ora_recovery*	中断 SnapMirror
	在目标 CVO 上启用 NFS 并为 Oracle 卷创建接合路径
	配置 DR Oracle 主机
	挂载并验证 Oracle 卷
	恢复并启动 Oracle 数据库

默认参数

为了简化自动化，我们已使用默认值预设了许多必需的 Oracle 参数。通常，无需更改大多数部署的默认参数。更高级的用户可以谨慎地更改默认参数。默认参数位于每个角色文件夹的默认目录下。

许可证

您应按照 Github 存储库中的说明读取许可证信息。访问，下载，安装或使用此存储库中的内容即表示您同意所规定的许可证条款 ["此处"](#)。

请注意，在生成和 / 或共享任何派生作品时，此存储库中的内容存在一定的限制。请务必阅读的条款 ["许可证"](#) 在使用内容之前。如果您不同意所有条款，请勿访问，下载或使用此存储库中的内容。

准备就绪后，单击 ["此处可查看 AWX/ 塔式服务器的详细流程"](#)。

分步部署操作步骤

本页介绍了 NetApp ONTAP 存储上 Oracle19c 的自动数据保护。

AWX/ 塔式 Oracle 数据保护

为您的环境创建清单、组、主机和凭据

本节介绍如何在 AWX/Ansible 塔中设置清单，组，主机和访问凭据，以便为使用 NetApp 自动化解决方案的环境做好准备。

- 配置清单。
 - 导航到资源→清单→添加，然后单击添加清单。
 - 提供名称和组织详细信息，然后单击保存。

- c. 在清单页面上，单击已创建的清单。
- d. 导航到组子菜单，然后单击添加。
- e. 提供第一个组的名称 `oracle`，然后单击 `Save`。
- f. 对名为 `dr_oracle` 的第二个组重复此过程。
- g. 选择已创建的 `Oracle` 组，转至主机子菜单，然后单击添加新主机。
- h. 提供源 `Oracle` 主机管理 IP 的 IP 地址，然后单击保存。
- i. 必须对 `dr_oracle` 组重复此过程，并添加 `DR/Destination Oracle` 主机的管理 IP/ 主机名。



下面介绍了如何使用 ONTAP 为内部部署或 AWS 上的 CVO 创建凭据类型和凭据。

内部

1. 配置凭据。
2. 创建凭据类型。对于涉及 ONTAP 的解决方案，您必须配置凭据类型以匹配用户名和密码条目。
 - a. 导航到管理→凭据类型，然后单击添加。
 - b. 提供名称和问题描述。
 - c. 将以下内容粘贴到输入配置中：

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

- d. 将以下内容粘贴到 " 注入器配置 " 中，然后单击保存：

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

3. 为 ONTAP 创建凭据
 - a. 导航到资源→凭据，然后单击添加。
 - b. 输入 ONTAP 凭据的名称和组织详细信息
 - c. 选择上一步中创建的凭据类型。
 - d. 在 Type Details 下，输入源集群和目标集群的用户名和密码。
 - e. 单击保存。
4. 为 Oracle 创建凭据
 - a. 导航到资源→凭据，然后单击添加。

- b. 输入 Oracle 的名称和组织详细信息。
- c. 选择计算机凭据类型。
- d. 在 Type Details 下，输入 Oracle 主机的 Username 和 Password。
- e. 选择正确的权限升级方法，然后输入用户名和密码。
- f. 单击保存。
- g. 如果需要为 dr_oracle 主机配置其他凭据，请重复此过程。

CVO

- 1. 配置凭据。
- 2. 创建凭据类型。对于涉及 ONTAP 的解决方案，您必须配置凭据类型以匹配用户名和密码条目，我们还会为 Cloud Central 和 AWS 添加条目。
 - a. 导航到管理→凭据类型，然后单击添加。
 - b. 提供名称和问题描述。
 - c. 将以下内容粘贴到输入配置中：


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

d. 将以下内容粘贴到 "Injector Configuration" 中，然后单击 "Save"：

```

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

```

3. 为 ontap/CVO/AWS 创建凭据

- a. 导航到资源→凭据，然后单击添加。
- b. 输入 ONTAP 凭据的名称和组织详细信息
- c. 选择上一步中创建的凭据类型。
- d. 在 Type Details 下，输入源集群和 CVO 集群， Cloud Central/Manager ， AWS 访问 / 机密密钥和 Cloud Central 刷新令牌的用户名和密码。
- e. 单击保存。

4. 为 Oracle 创建凭据（源）

- a. 导航到资源→凭据，然后单击添加。
- b. 输入 Oracle 主机的名称和组织详细信息
- c. 选择计算机凭据类型。
- d. 在 Type Details 下，输入 Oracle 主机的 Username 和 Password 。
- e. 选择正确的权限升级方法，然后输入用户名和密码。
- f. 单击保存。

5. 为 Oracle 目标创建凭据

- a. 导航到资源→凭据，然后单击添加。
- b. 输入 DR Oracle 主机的名称和组织详细信息
- c. 选择计算机凭据类型。
- d. 在 Type Details 下，输入 Username （ EC2-user ， 或者如果您已对其进行了默认更改，请输入该用户名）和 SSH 私钥
- e. 选择正确的权限升级方法（ sudo ），然后根据需要输入用户名和密码。
- f. 单击保存。

创建项目

1. 转至 "Resources" → "projects" ，然后单击 "Add" 。
 - a. 输入名称和组织详细信息。
 - b. 在源控制凭据类型字段中选择 Git 。
 - c. 输入 ... <https://github.com/NetApp-Automation/na_oracle19c_data_protection.git> 作为源控制 URL 。
 - d. 单击保存。
 - e. 当源代码发生更改时，项目可能偶尔需要同步。

配置全局变量

本节中定义的变量适用于所有 Oracle 主机，数据库和 ONTAP 集群。

1. 在以下嵌入式全局变量或变量表单中输入环境特定的参数。



必须根据您的环境更改蓝色项。

内部

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

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

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

create_source_intercluster_lifs: "yes"

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

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

create_destination_intercluster_lifs: "yes"
```

```

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

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

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

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

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

```

```

snapmirror_policy: "async_policy_oracle"

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

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

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

```

CVO

```

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

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

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

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

#Names of the Nodes in the Destination CVO Cluster

```

```

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

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

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

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

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

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

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

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

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

```

```

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

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

# Name your connector.
connector_name: "awx_connector"

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

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

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

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

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

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

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

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

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

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

```



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

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

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

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

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

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

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

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

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

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

```

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

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

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

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

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

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

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

```

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

自动化攻略手册

需要运行四本单独的攻略手册。

1. 用于设置环境的攻略手册，内部部署或 CVO 。
2. 用于按计划复制 Oracle 二进制文件和数据库的攻略手册
3. 用于按计划复制 Oracle 日志的攻略手册
4. 用于在目标主机上恢复数据库的攻略手册

ONTAP/CVO 设置

ONTAP 和 CVO 设置

配置并启动作业模板。

1. 创建作业模板。
 - a. 导航到资源→模板→添加，然后单击添加作业模板。
 - b. 输入名称 ontap/CVO 设置
 - c. 选择作业类型；运行将根据攻略手册配置系统。
 - d. 为攻略手册选择相应的清单，项目，攻略手册和凭据。
 - e. 为内部环境选择 ontap_setup.yml 攻略手册，或者选择 CVO_setup.yml 以复制到 CVO 实例。
 - f. 将从步骤 4 复制的全局变量粘贴到 YAML 选项卡下的模板变量字段中。
 - g. 单击保存。
2. 启动作业模板。
 - a. 导航到资源→模板。
 - b. 单击所需模板，然后单击启动。



我们将使用此模板并将其复制到其他攻略手册中。

二进制卷和数据库卷的复制

计划二进制文件和数据库复制攻略手册

配置并启动作业模板。

1. 复制先前创建的作业模板。
 - a. 导航到资源→模板。
 - b. 找到 ontap/CVO 设置模板，然后在最右侧单击复制模板
 - c. 单击复制的模板上的编辑模板，然后将名称更改为二进制和数据库复制攻略手册。
 - d. 保留该模板的相同清单，项目和凭据。
 - e. 选择 ora_replication cc.yml 作为要执行的攻略手册。
 - f. 这些变量将保持不变，但需要在变量 dst_cluster_ip 中设置 CVO 集群 IP。
 - g. 单击保存。
2. 计划作业模板。
 - a. 导航到资源→模板。
 - b. 单击二进制和数据库复制攻略手册模板，然后单击顶部选项集的计划。
 - c. 单击添加，为二进制和数据库复制添加名称计划，在小时开始时选择开始日期 / 时间，选择您的本地时区以及运行频率。运行频率通常会更新 SnapMirror 复制。



系统将为日志卷复制创建一个单独的计划，以便可以更频繁地进行复制。

复制日志卷

计划日志复制攻略手册

配置并启动作业模板。

1. 复制先前创建的作业模板。
 - a. 导航到资源→模板。
 - b. 找到 ontap/CVO 设置模板，然后在最右侧单击复制模板
 - c. 单击复制的模板上的 Edit Template ，然后将名称更改为 Log Replication 攻略手册。
 - d. 保留该模板的相同清单，项目和凭据。
 - e. 选择 ora_replication logs.yml 作为要执行的攻略手册。
 - f. 这些变量将保持不变，但需要在变量 dst_cluster_ip 中设置 CVO 集群 IP 。
 - g. 单击保存。
2. 计划作业模板。
 - a. 导航到资源→模板。
 - b. 单击日志复制攻略手册模板，然后单击顶部选项集的计划。
 - c. 单击添加，为日志复制添加名称计划，选择开始日期 / 时间，选择本地时区和运行频率。运行频率通常会更新 SnapMirror 复制。



建议将日志计划设置为每小时更新一次，以确保恢复到上一个每小时更新。

还原和恢复数据库

计划日志复制攻略手册

配置并启动作业模板。

1. 复制先前创建的作业模板。
 - a. 导航到资源→模板。
 - b. 找到 ontap/CVO 设置模板，然后在最右侧单击复制模板
 - c. 单击复制的模板上的 Edit Template ，然后将名称更改为 Restore and Recovery Playbook 。
 - d. 保留该模板的相同清单，项目和凭据。
 - e. 选择 ora_recovery.yml 作为要执行的攻略手册。
 - f. 这些变量将保持不变，但需要在变量 dst_cluster_ip 中设置 CVO 集群 IP 。
 - g. 单击保存。



只有在准备好在远程站点还原数据库后，才会运行此攻略手册。

1. 内部生产 Oracle 数据库数据卷通过 NetApp SnapMirror 复制到二级数据中心的冗余 ONTAP 集群或公有云中的 Cloud Volume ONTAP 进行保护。在完全配置的灾难恢复环境中，二级数据中心或公有云中的恢复计算实例处于备用状态，可以在发生灾难时恢复生产数据库。备用计算实例通过在操作系统内核修补程序上运行准面更新或在一个锁定步骤中进行升级，与内部实例保持同步。
2. 在此解决方案演示中，Oracle 二进制卷会复制到目标并挂载到目标实例，以启动 Oracle 软件堆栈。与灾难发生时的最后一分钟全新安装 Oracle 相比，这种恢复 Oracle 的方法更有优势。它可以保证 Oracle 安装与当前内部生产软件安装和修补程序级别等完全同步。但是，根据 Oracle 软件许可的结构，这可能会对恢复站点上复制的 Oracle 二进制卷产生额外的软件许可影响，也可能不会产生额外的软件许可影响。建议用户在决定使用相同方法之前，先咨询其软件许可人员，以评估潜在的 Oracle 许可要求。
3. 目标位置的备用 Oracle 主机已配置 Oracle 前提条件配置。
4. SnapMirrors 已损坏，卷可写并挂载到备用 Oracle 主机。
5. 在备用计算实例上挂载所有数据库卷后，Oracle 恢复模块将在恢复站点执行以下恢复和启动 Oracle 任务。
 - a. 同步控制文件：我们在不同的数据库卷上部署了重复的 Oracle 控制文件，以保护关键数据库控制文件。一个位于数据卷上，另一个位于日志卷上。由于数据卷和日志卷的复制频率不同，因此在恢复时它们将不同步。
 - b. 重新链接 Oracle 二进制文件：由于 Oracle 二进制文件已重新定位到新主机，因此需要重新链接。
 - c. 恢复 Oracle 数据库：恢复机制从控制文件中检索 Oracle 日志卷中最后一个可用归档日志中的最后一个系统更改编号，并恢复 Oracle 数据库以重新处理在发生故障时能够复制到灾难恢复站点的所有业务事务。然后，数据库将在新的形式中启动，以在恢复站点执行用户连接和业务事务。



运行恢复攻略手册之前，请确保您具备以下条件：确保它通过 /etc/oratab 和 /etc/orainst.oc 从源 Oracle 主机复制到目标主机

TR-4794：《基于NetApp EF系列的Oracle数据库》

NetApp公司Ebin Kadavy的Mitch Blackburn

TR-4794旨在帮助存储管理员和数据库管理员在NetApp EF系列存储上成功部署Oracle。

"TR-4794：《基于NetApp EF系列的Oracle数据库》"

版权信息

版权所有 © 2024 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。