



온프레미스/하이브리드 클라우드 NetApp Solutions

NetApp
May 03, 2024

목차

온프레미스/하이브리드 클라우드	1
TR-4983: iSCSI를 지원하는 NetApp ASA에서 간편하고 자동화된 Oracle 배포	1
NVA-1155: FC-Design 및 구축 가이드를 통해 FlexPod 데이터 센터 및 Cisco UCS 및 NetApp AFF A800에 대한 Oracle 19c RAC 데이터베이스	17
TR-4250: SAP with Oracle on UNIX 및 NFS with NetApp Clustered Data ONTAP and SnapManager for SAP 3.4	17
Oracle Database 구축	17
솔루션 개요	38
TR-4794: NetApp EF-Series 기반의 Oracle 데이터베이스	60

온프레미스/하이브리드 클라우드

TR-4983: iSCSI를 지원하는 NetApp ASA에서 간편하고 자동화된 Oracle 배포

Allen Cao, Niyaz Mohamed, NetApp

목적

NetApp ASA 시스템은 SAN 인프라에 최신 솔루션을 제공합니다. 규모에 따라 간소화하고, 데이터베이스와 같은 비즈니스 크리티컬 애플리케이션의 성능을 높이고, 데이터 가용성(99.9999% 가동 시간)을 보장하고, TCO와 탄소 발자국을 줄일 수 있도록 지원합니다. NetApp ASA 시스템에는 가장 높은 성능이 요구되는 애플리케이션을 위해 설계된 A-Series 모델과 비용 효율적인 대용량 구축을 위해 최적화된 C-Series 모델이 포함됩니다. ASA A-Series 및 C-Series 시스템은 탁월한 성능을 제공하여 고객 경험을 향상하고 결과 도출 시간을 단축하고, 비즈니스 크리티컬 데이터의 가용성을 유지하고, 안전하게 보호하며, 모든 워크로드에 대해 업계에서 가장 효율적인 보장을 바탕으로 더 많은 실제 용량을 제공합니다.

이 문서에서는 Ansible 자동화를 사용하여 ASA 시스템으로 구축된 SAN 환경에서 Oracle 데이터베이스를 간단하게 구현하는 방법을 설명합니다. Oracle 데이터베이스는 데이터 액세스를 위한 iSCSI 프로토콜 및 ASA 스토리지 시스템에서 데이터베이스 디스크 관리를 위한 Oracle ASM을 사용하는 독립 실행형 재시작 구성으로 구축됩니다. 또한 NetApp ASA 시스템에서 스토리지 효율적인 데이터베이스 작업을 위해 NetApp SnapCenter UI 툴을 사용하여 Oracle 데이터베이스 백업, 복원 및 복제에 대한 정보도 제공합니다.

이 솔루션은 다음과 같은 사용 사례를 해결합니다.

- NetApp ASA 시스템에서 기본 데이터베이스 스토리지로 Oracle 데이터베이스 구현을 자동화합니다
- NetApp SnapCenter 툴을 사용하여 NetApp ASA 시스템에서 Oracle 데이터베이스 백업 및 복원
- NetApp SnapCenter 툴을 사용하여 NetApp ASA 시스템에서 개발/테스트나 기타 사용 사례를 위한 Oracle 데이터베이스 클론

대상

이 솔루션은 다음과 같은 사용자를 대상으로 합니다.

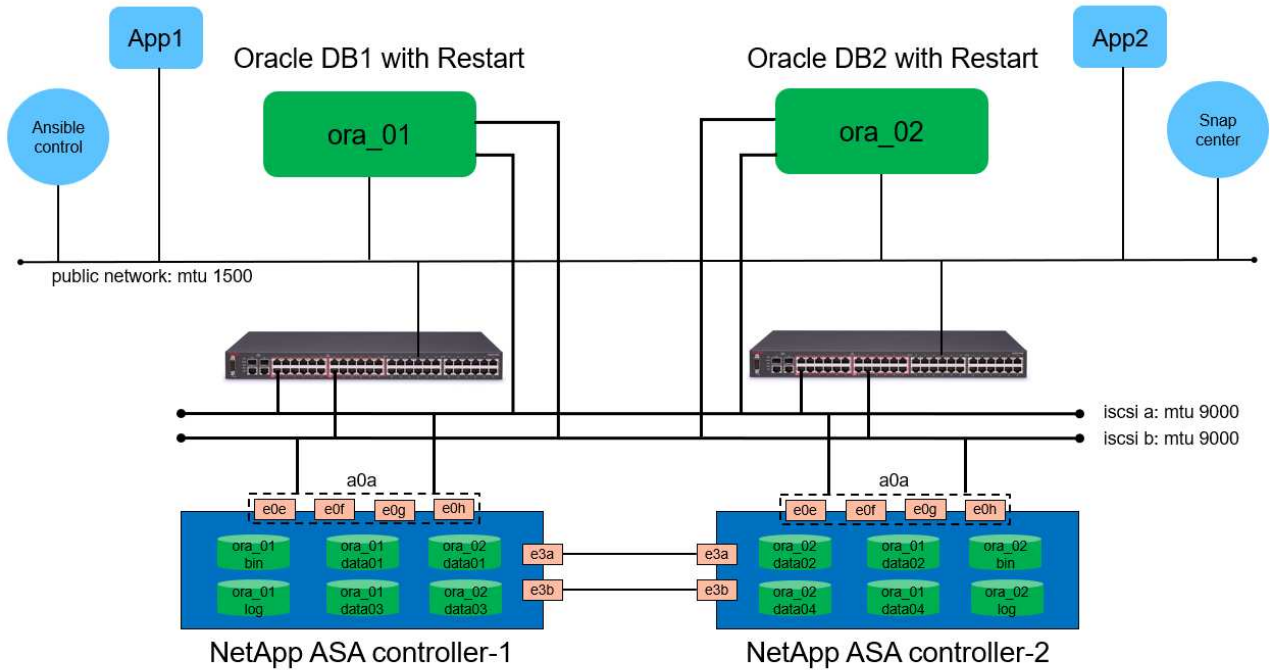
- NetApp ASA 시스템에 Oracle을 구축하려는 DBA입니다.
- NetApp ASA 시스템에서 Oracle 워크로드를 테스트하려는 데이터베이스 솔루션 설계자입니다.
- NetApp ASA 시스템에서 Oracle 데이터베이스를 구축하고 관리하려는 스토리지 관리자
- NetApp ASA 시스템에서 Oracle 데이터베이스를 사용하고자 하는 애플리케이션 소유자

솔루션 테스트 및 검증 환경

이 솔루션의 테스트 및 검증은 최종 구축 환경과 일치하지 않을 수 있는 랩 설정에서 수행되었습니다. 섹션을 참조하십시오 [\[Key Factors for Deployment Consideration\]](#) 를 참조하십시오.

있습니다

Simplified, Automated Oracle Database Deployment on NetApp ASA with iSCSI



NetApp

하드웨어 및 소프트웨어 구성 요소

* 하드웨어 *		
NetApp ASA A400	버전 9.13.1P1	NS224 쉘프 2개, 총 69.3TiB 용량의 NVMe AFF 드라이브 48개
UCSB-B200-M4를 참조하십시오	인텔® 제온® CPU E5-2690 v4 @ 2.60GHz	4노드 VMware ESXi 클러스터
* 소프트웨어 *		
RedHat Linux	RHEL-8.6, 4.18.0-372.9.1.el8.x86_64 커널	테스트를 위해 RedHat 서브스크립션을 배포했습니다
Windows Server를 선택합니다	2022 표준, 10.0.20348 빌드 20348	호스팅 SnapCenter 서버
Oracle Grid Infrastructure	버전 19.18	RU 패치 p34762026_190000_Linux-x86-64.zip 를 적용했습니다
Oracle 데이터베이스	버전 19.18	RU 패치 p34765931_190000_Linux-x86-64.zip 를 적용했습니다
Oracle OPatch	버전 12.2.0.1.36	최신 패치 p6880880_190000_Linux-x86-64.zip
SnapCenter 서버	버전 4.9P1	작업 그룹 배포
VMware vSphere 하이퍼바이저	버전 6.5.0.20000	VMware Tools, 버전: 11365 - Linux, 12352 - Windows

JDK를 엽니다	버전 java-1.8.0-openjdk.x86_64	DB VM에 대한 SnapCenter 플러그인 요구 사항
----------	------------------------------	---------------------------------

실습 환경의 Oracle 데이터베이스 구성

* 서버 *	* 데이터베이스 *	* DB 스토리지 *
오라_01	NTAP1(NTAP1_PDB1, NTAP1_PDB2, NTAP1_PDB3)	ASA A400의 iSCSI LUN
오라_02	NTAP2(NTAP2_PDB1, NTAP2_PDB2, NTAP2_PDB3)	ASA A400의 iSCSI LUN

구축 시 고려해야 할 주요 요소

- * Oracle 데이터베이스 스토리지 레이아웃. * 이 자동화된 Oracle 배포에서는 기본적으로 Oracle 바이너리, 데이터 및 로그를 호스팅하기 위해 4개의 데이터베이스 볼륨을 프로비저닝합니다. 그런 다음 데이터 및 로그 LUN에서 2개의 ASM 디스크 그룹을 생성합니다. Data ASM 디스크 그룹 내에서 각 ASA A400 클러스터 노드의 볼륨에 2개의 데이터 LUN을 프로비저닝합니다. logs ASM 디스크 그룹 내에서는 단일 ASA A400 노드의 로그 볼륨에 2개의 LUN을 생성합니다. ONTAP 볼륨 내에 여러 LUN이 배치되므로 일반적으로 성능이 향상됩니다.
- * 여러 DB 서버 배포. * 자동화 솔루션은 하나의 Ansible 플레이북 실행에서 여러 DB 서버에 Oracle 컨테이너 데이터베이스를 구축할 수 있습니다. DB 서버 수에 관계없이 Playbook 실행은 동일하게 유지됩니다. 멀티 DB 서버 배포의 경우, 플레이북은 데이터베이스 LUN을 ASA A400의 이중 컨트롤러에 최적으로 배치하는 알고리즘을 사용하여 구축됩니다. 홀수 DB 서버의 이진 및 로그 LUN은 컨트롤러 1의 서버 호스트 인덱스 위치에 있습니다. 짝수 DB 서버의 이진 및 로그 LUN은 컨트롤러 2의 서버 호스트 인덱스 위치에 있습니다. DB 데이터 LUN이 두 컨트롤러에 균등하게 분산됩니다. Oracle ASM은 두 컨트롤러의 데이터 LUN을 단일 ASM 디스크 그룹으로 결합하여 두 컨트롤러의 처리 기능을 최대한 활용합니다.
- * iSCSI 구성. * 데이터베이스 VM은 스토리지 액세스를 위한 iSCSI 프로토콜을 사용하여 ASA 스토리지에 연결됩니다. 이중화를 위해 각 컨트롤러 노드에서 이중 경로를 구성하고 다중 경로 스토리지 액세스를 위해 DB 서버에서 iSCSI 다중 경로를 설정해야 합니다. 스토리지 네트워크에서 점보 프레임 활성화하여 성능과 처리량을 극대화합니다.
- * 귀하가 생성하는 각 **Oracle ASM** 디스크 그룹에 사용할 **Oracle ASM** 이중화 수준. ASA A400은 클러스터 디스크 수준에서 데이터 보호를 위해 RAID DP의 스토리지를 구성하므로 이를 사용해야 합니다. `External Redundancy` 다시 말해, 이 옵션이 Oracle ASM이 디스크 그룹의 콘텐츠를 미러링하도록 허용하지 않습니다.
- * 데이터베이스 백업. * NetApp은 사용자에게 친숙한 UI 인터페이스를 통해 데이터베이스 백업, 복원 및 복제를 위한 SnapCenter 소프트웨어 제품군을 제공합니다. NetApp은 이와 같은 관리 툴을 구현하여 1분 이내에 신속하게 스냅샷 백업, 신속한(분) 데이터베이스 복원 및 데이터베이스 복제를 수행할 것을 권장합니다.

솔루션 구축

다음 섹션에서는 Oracle ASM을 데이터베이스 볼륨 관리자로 사용하는 단일 노드 재시작 구성에서 iSCSI를 통해 DB VM에 직접 마운트된 데이터베이스 LUN을 사용하는 NetApp ASA A400의 자동화된 Oracle 19c 배포 및 보호를 위한 단계별 절차를 제공합니다.

배포를 위한 사전 요구 사항

배포에는 다음과 같은 사전 요구 사항이 필요합니다.

1. NetApp ASA 스토리지 시스템이 설치 및 구성된 것으로 가정합니다. 여기에는 iSCSI 브로드캐스트 도메인, 두 컨트롤러 노드의 LACP 인터페이스 그룹 a0a, 두 컨트롤러 노드의 iSCSI VLAN 포트(a0a-<iscsi-a-vlan-id>, a0a-<iscsi-b-vlan-id>)가 포함됩니다. 다음 링크는 도움이 필요한 경우 자세한 단계별 지침을 제공합니다. "[자세한 가이드 - ASA A400](#)"
2. 최신 버전의 Ansible 및 Git가 설치된 Ansible 컨트롤러 노드로 Linux VM을 프로비저닝합니다. 자세한 내용은 다음 링크를 참조하십시오. "[NetApp 솔루션 자동화 시작하기](#)" 섹션 - Setup the Ansible Control Node for CLI deployments on RHEL / CentOS 또는 Setup the Ansible Control Node for CLI deployments on Ubuntu / Debian.
3. NetApp Oracle Deployment Automation Toolkit for iSCSI의 복제본을 복제합니다.

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

4. NetApp SnapCenter UI 도구를 최신 버전으로 실행할 Windows 서버를 프로비저닝합니다. 자세한 내용은 다음 링크를 참조하십시오. "[SnapCenter 서버를 설치합니다](#)"
5. 베어 메탈 또는 가상화된 VM 중 RHEL Oracle DB 서버 두 대를 구축합니다. 암호 권한이 없는 sudo를 사용하여 DB 서버에 관리자 사용자를 생성하고 Ansible 호스트와 Oracle DB 서버 호스트 간에 SSH 개인 /공개 키 인증을 활성화합니다. DB server/tmp/archive 디렉토리에 Oracle 19c 설치 파일 스테이징

```
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"
```



Oracle 설치 파일을 스테이징할 수 있는 충분한 공간을 확보하기 위해 Oracle VM 루트 볼륨에 50G 이상을 할당했는지 확인합니다.

6. 다음 동영상을 시청하십시오.

[iSCSI를 사용하여 NetApp ASA에서 Oracle을 간편하고 자동으로 구현합니다](#)

자동화 매개 변수 파일

Ansible 플레이북은 사전 정의된 매개 변수를 사용하여 데이터베이스 설치 및 구성 작업을 실행합니다. 이 Oracle 자동화 솔루션의 경우 플레이북을 실행하기 전에 사용자가 입력해야 하는 세 가지 사용자 정의 매개 변수 파일이 있습니다.

- 호스트 - 자동화 플레이북이 실행되는 타겟을 정의합니다.
- vars/vars.yml - 모든 대상에 적용되는 변수를 정의하는 전역 변수 파일입니다.
- host_vars/host_name.yml - 로컬 대상에만 적용되는 변수를 정의하는 지역 변수 파일입니다. 본 사용 사례에서는 Oracle DB 서버가 해당됩니다.

이러한 사용자 정의 변수 파일 외에도 필요한 경우가 아니면 변경할 필요가 없는 기본 매개 변수가 포함된 여러 기본 변수 파일이 있습니다. 다음 섹션에서는 사용자 정의 변수 파일을 구성하는 방법을 보여 줍니다.

매개 변수 파일 구성

1. Ansible 대상 hosts 파일 구성:

```
# 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. 글로벌 vars/vars.yml 파일 구성

```
#####
#####
#####
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: "xxxxxxx"

#####
#####
###
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. 로컬 DB 서버 host_vars/host_name.yml 구성

```
# 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
```

플레이북 실행

자동화 툴킷에는 총 6개의 플레이북이 있습니다. 각 작업 블록은 서로 다른 작업 블록을 수행하며 서로 다른 용도로 사용됩니다.

```
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.
```

다음 명령을 사용하여 플레이북을 실행할 수 있는 세 가지 옵션이 있습니다.

1. 모든 구현 플레이북을 하나의 실행 방식으로 실행합니다.

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

2. 1-4의 숫자 순서에 따라 한 번에 하나씩 플레이북을 실행합니다.

```
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. 태그를 사용하여 0-ALL_Playbook.yml을 실행합니다.

```
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. 환경을 실행 취소하십시오

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

사후 실행 검증

플레이백을 실행한 후 Oracle DB 서버에 Oracle 사용자로 로그인하여 Oracle 그리드 인프라 및 데이터베이스가 성공적으로 생성되었는지 확인합니다. 다음은 호스트 ora_01에서 Oracle 데이터베이스 검증의 예입니다.

1. 생성된 그리드 인프라 및 리소스를 검증합니다.

```
[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
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 ~]$

```



를 무시합니다 Not All Endpoints Registered 상태 세부 정보. 이는 수신기와 수동 및 동적 데이터베이스 등록이 충돌하여 발생하므로 무시해도 됩니다.

2. ASM 필터 드라이버가 예상대로 작동하는지 확인합니다.


```

[oracle@ora_01 ~]$ asmcmd
ASMCMDB> 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
N  DATA/
MOUNTED  EXTERN  N      512     512     4096    4194304
81920   78880      0      78880
N  LOGS/
ASMCMDB> 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
ASMCMDB> afd_state
ASMCMDB-9526: The AFD state is 'LOADED' and filtering is 'ENABLED' on
host 'ora_01'
ASMCMDB>


```

3. Oracle Enterprise Manager Express에 로그인하여 데이터베이스를 검증합니다.

← → ↻ Not secure | https://10.61.180.21:5500/em/login



ORACLE ENTERPRISE MANAGER DATABASE EXPRESS



Copyright 2013, 2020, Oracle and/or its affiliates. All rights reserved.

← → ↻ Not secure | https://10.61.180.21:5500/em/shell

ORACLE Enterprise Manager Database Express
system ▾

NTAP1 (19.18.0.0.0) Performance ▾ Storage ▾

Database Home

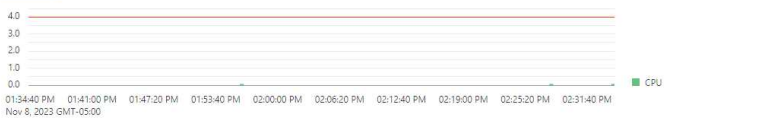
Time Zone: Browser (GMT-05:00) ▾ 1 min Auto-Refresh ▾ Refresh

Status

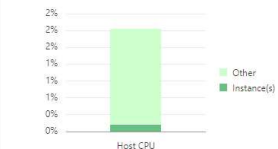

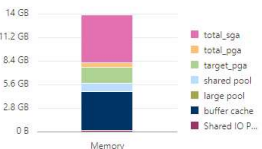
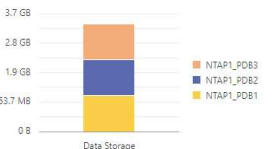
Up Time 1 hours, 7 minutes, 23 seconds
 Type Single Instance (NTAP1)
 CDB (3 PDB(s))
 Version 19.18.0.0.0 Enterprise Edition
 Platform Name Linux x86 64-bit
 Thread 1
 Archiver Stopped
 Last Backup Time N/A
 Incident(s) ❗ 4

Performance

Activity Services Containers



Resources

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.GETHTTPS()
-----
                                0
```

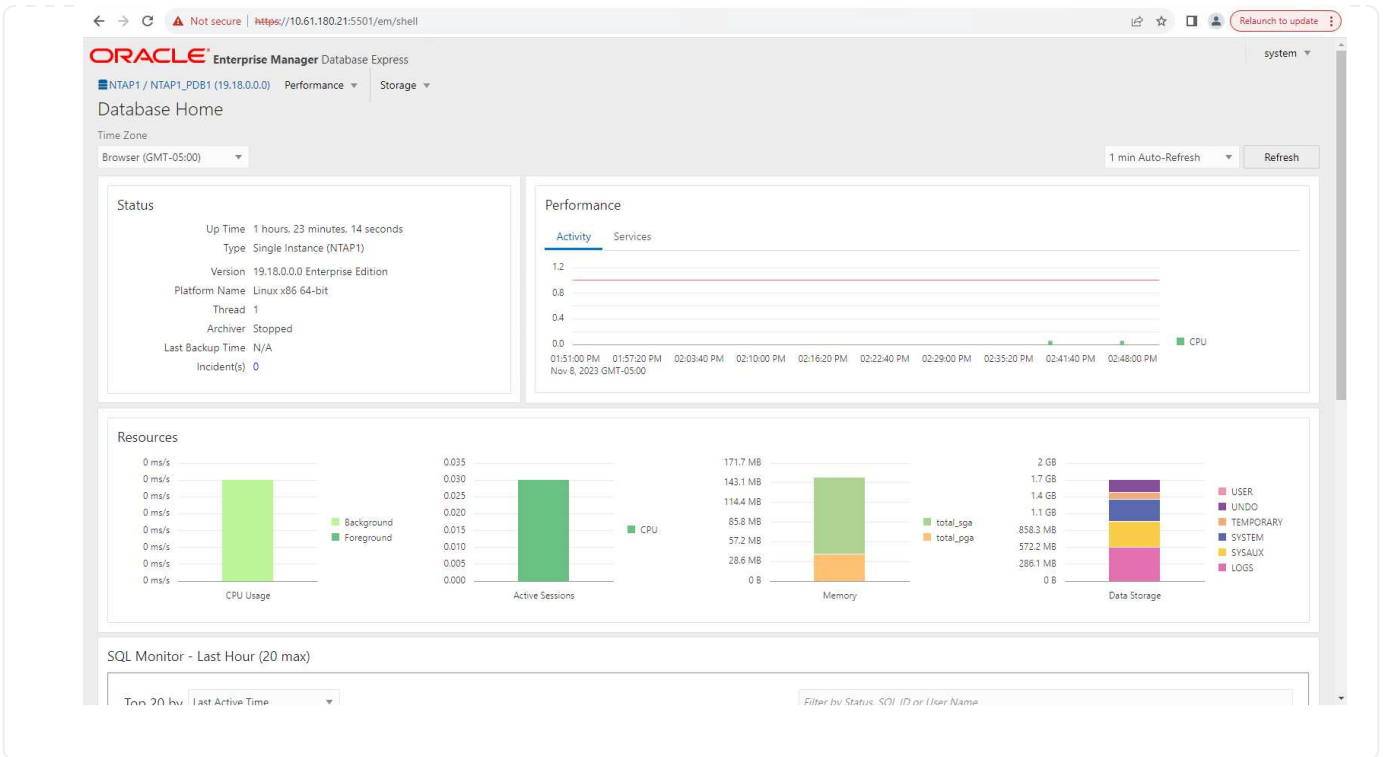
```
SQL> exec DBMS_XDB_CONFIG.SETHTTPS(5501);
```

PL/SQL procedure successfully completed.

```
SQL> select dbms_xdb_config.gethttpsport() from dual;
```

```
DBMS_XDB_CONFIG.GETHTTPS()
-----
                                5501
```

login to NTAP1_PDB1 from port 5501.



SnapCenter를 사용하여 Oracle 백업, 복원 및 클론 복제를 수행합니다

TR-4979를 참조하십시오 ["게스트 마운트 FSx ONTAP를 사용하여 AWS 기반 VMware Cloud에서 간편하게 자가 관리 가능한 Oracle"](#) 섹션을 참조하십시오 Oracle backup, restore, and clone with SnapCenter SnapCenter 설정 및 데이터베이스 백업, 복원 및 클론 워크플로우의 실행에 대한 자세한 내용을 참조하십시오.

추가 정보를 찾을 수 있는 위치

이 문서에 설명된 정보에 대한 자세한 내용은 다음 문서 및/또는 웹 사이트를 참조하십시오.

- NetApp ASA: All-Flash SAN 어레이

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

- 새 데이터베이스 설치를 통해 독립 실행형 서버용 Oracle Grid Infrastructure 설치

["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)

- 응답 파일을 사용하여 Oracle 데이터베이스 설치 및 구성

["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)

- ONTAP와 함께 Red Hat Enterprise Linux 8.2를 사용하십시오

["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: FC-Design 및 구축 가이드를 통해 FlexPod 데이터 센터 및 Cisco UCS 및 NetApp AFF A800에 대한 Oracle 19c RAC 데이터베이스

NetApp, Allen Cao

FC를 통한 FlexPod 데이터 센터 및 Cisco UCS와 NetApp AFF A800의 Oracle 19c RAC 데이터베이스에 대한 이 설계 및 구축 가이드는 Oracle Linux 8.2를 사용하는 최신 FlexPod 데이터 센터 인프라에 Oracle RAC 데이터베이스를 호스팅하기 위한 단계별 구축 프로세스와 솔루션 설계에 대한 세부 정보를 제공합니다 운영 체제 및 Red Hat 호환 커널.

["NVA-1155: FC를 통해 FlexPod 데이터 센터 및 Cisco UCS 및 NetApp AFF A800의 Oracle 19c RAC 데이터베이스"](#)

TR-4250: SAP with Oracle on UNIX 및 NFS with NetApp Clustered Data ONTAP and SnapManager for SAP 3.4

NetApp, Nils Bauer

TR-4250은 Oracle 데이터베이스를 사용하여 SAP 비즈니스 제품군 제품을 지원하는 스토리지 솔루션을 설계하는 데 따르는 문제를 해결합니다. 이 문서는 최신 SAP 솔루션을 사용하는 비즈니스 및 IT 리더가 직면한 일반적인 스토리지 인프라 설계, 구축, 운영 및 관리 문제를 중점적으로 다룹니다. 이 문서의 권장 사항은 일반적인 내용이며 SAP 애플리케이션 또는 SAP 구축 규모와 범위에 국한되지 않습니다. TR-4250은 독자가 NetApp 및 SAP 제품의 기술과 운영에 대해 기본적인 지식을 갖추고 있다고 가정합니다. TR-4250은 NetApp, SAP, Oracle, 고객 기술 직원의 상호 작용을 기반으로 개발되었습니다.

["TR-4250: SAP with Oracle on UNIX 및 NFS with NetApp Clustered Data ONTAP and SnapManager for SAP 3.4"](#)

Oracle Database 구축

솔루션 개요

NFS에서 ONTAP용 Oracle19c의 자동 배포

조직은 효율성을 높이고, 구현을 앞당기고, 수동 작업을 줄이기 위해 환경을 자동화하고 있습니다. Ansible과 같은 구성 관리 툴을 사용하여 엔터프라이즈 데이터베이스 운영을 간소화하고 있습니다. 이 솔루션에서는 Ansible을 사용하여 NetApp ONTAP을 사용하여 Oracle 19c의 프로비저닝과 구성을 자동화하는 방법을 보여줍니다. 스토리지 관리자, 시스템 관리자 및 DBA가 새 스토리지를 일관성 있게 신속하게 구축하고 데이터베이스 서버를 구성하며 Oracle 19c 소프트웨어를 설치할 수 있도록 함으로써 다음과 같은 이점을 얻을 수 있습니다.

- 설계 복잡성과 인적 오류를 제거하고 반복 가능한 일관된 구축 및 모범 사례를 구현합니다
- 스토리지 프로비저닝, DB 호스트 구성, Oracle 설치 시간 단축
- 데이터베이스 관리자, 시스템 및 스토리지 관리자의 생산성 향상
- 스토리지와 데이터베이스를 손쉽게 확장

NetApp은 고객에게 검증된 Ansible 모듈과 역할을 제공하여 Oracle 데이터베이스 환경의 구축, 구성, 라이프사이클 관리를 가속합니다. 이 솔루션은 다음을 지원하기 위한 지침 및 Ansible 플레이북 코드를 제공합니다.

- Oracle 데이터베이스용 ONTAP NFS 스토리지를 생성하고 구성합니다
- RedHat Enterprise Linux 7/8 또는 Oracle Linux 7/8 에 Oracle 19c를 설치합니다
- ONTAP NFS 스토리지에 Oracle 19c를 구성합니다

자세한 내용을 보거나 시작하려면 아래의 개요 비디오를 참조하십시오.

AWX/타워 배포

1부: 시작하기, 요구 사항, 자동화 세부 정보 및 초기 AWX/타워 구성

AWX 배포

2부: 변수 및 Playbook 실행

AWX Playbook 실행

CLI 배포

1부: 시작하기, 요구사항, 자동화 세부 정보 및 Ansible Control 호스트 설정

CLI 배포

2부: 변수 및 Playbook 실행

CLI Playbook 실행

시작하기

이 솔루션은 AWX/Tower 환경 또는 Ansible 제어 호스트의 CLI에서 실행되도록 설계되었습니다.

AWX/타워

AWX/Tower 환경의 경우 ONTAP 클러스터 관리 및 Oracle 서버(IP 및 호스트 이름)의 인벤토리 생성, 자격 증명 생성, NetApp Automation GitHub에서 Ansible 코드를 가져오는 프로젝트 구성, 자동화를 시작하는 작업 템플릿 등이 있습니다.

1. 사용자 환경에 맞는 변수를 입력하고 이를 복사하여 작업 템플릿의 추가 VAR 필드에 붙여 넣습니다.
2. 추가 VAR이 작업 템플릿에 추가되면 자동화를 시작할 수 있습니다.
3. 작업 템플릿은 ONTAP_config, Linux_config 및 Oracle_config에 대한 태그를 지정하여 3단계로 실행됩니다.

Ansible 제어 호스트를 통해 CLI

1. Ansible 제어 호스트로 사용할 수 있도록 Linux 호스트를 구성합니다"[자세한 지침을 보려면 여기를 클릭하십시오](#)"
2. Ansible 제어 호스트를 구성한 후 Ansible 자동화 저장소를 클론 복제할 수 있습니다.
3. ONTAP 클러스터 관리 및 Oracle 서버 관리 IP의 IP 및/또는 호스트 이름으로 hosts 파일을 편집합니다.
4. 해당 환경에 맞는 변수를 작성하고 복사하여 VAR.yml 파일에 붙여 넣습니다.
5. 각 Oracle 호스트에는 호스트별 변수를 포함하는 호스트 이름으로 식별되는 변수 파일이 있습니다.

6. 모든 변수 파일이 완료된 후 'ONTAP_config', 'linux_config', 'oracle_config'에 대한 태그를 지정하여 3단계로 플레이북을 실행할 수 있습니다.

요구 사항

방법입니다	요구 사항
* Ansible 환경 *	Ansible 제어 호스트가 될 AWX/Tower 또는 Linux 호스트
	Ansible v. 2.10 이상
	Python 3
	Python 라이브러리 - NetApp-lib-xmltodict-jmespath
* ONTAP *	ONTAP 버전 9.3-9.7
	두 개의 데이터 애그리게이트
	NFS VLAN 및 ifgrp가 생성되었습니다
* Oracle 서버 *	RHEL 7/8
	Oracle Linux 7/8
	NFS, 공용 및 선택적 관리를 위한 네트워크 인터페이스
	Oracle 서버에 Oracle 설치 파일

자동화 세부 정보

이 자동 배포는 3개의 개별 역할로 구성된 단일 Ansible 플레이북을 통해 설계되었습니다. 역할은 ONTAP, Linux 및 Oracle 구성을 위한 것입니다. 다음 표에서는 자동화되고 있는 작업을 설명합니다.

역할	작업
ONTAP_구성 * 을 참조하십시오	ONTAP 환경 사전 점검
	Oracle용 NFS 기반 SVM 생성
	엑스포트 정책 생성
	Oracle의 볼륨 생성
	NFS LIF 생성
Linux_config *	마운트 지점을 생성하고 NFS 볼륨을 마운트합니다
	NFS 마운트를 확인합니다
	OS별 구성
	Oracle 디렉토리를 생성합니다
	HugePages를 구성합니다
	SELinux 및 방화벽 데몬을 해제합니다
	시간 기록 서비스를 활성화하고 시작합니다
	파일 설명자 하드 제한값을 늘립니다
	PAM.d 세션 파일을 생성합니다

역할	작업
* oracle_config *	Oracle 소프트웨어 설치
	Oracle Listener를 생성합니다
	Oracle 데이터베이스를 생성합니다
	Oracle 환경 구성
	PDB 상태를 저장합니다
	인스턴스 아카이브 모드를 활성화합니다
	DNFS 클라이언트를 활성화합니다
	OS 재부팅 간에 데이터베이스 자동 시작 및 종료를 활성화합니다

기본 매개변수

자동화를 간소화하기 위해 필요한 많은 Oracle 구축 매개 변수를 기본값으로 사전 설정하였습니다. 일반적으로 대부분의 배포에서 기본 매개 변수를 변경할 필요는 없습니다. 고급 사용자는 기본 매개 변수를 주의 깊게 변경할 수 있습니다. 기본 매개 변수는 각 역할 폴더의 defaults 디렉토리에 있습니다.

배포 지침

시작하기 전에 다음 Oracle 설치 및 패치 파일을 다운로드하여 구축할 각 DB 서버의 모든 사용자에게 대한 읽기, 쓰기 및 실행 액세스 권한이 있는 '/tmp/archive' 디렉토리에 배치합니다. 자동화 작업은 Oracle 설치 및 구성을 위해 특정 디렉토리에 명명된 설치 파일을 찾습니다.

```
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
```

라이선스

GitHub 리포지토리에 설명된 대로 라이선스 정보를 읽어야 합니다. 이 리포지토리의 콘텐츠에 액세스, 다운로드, 설치 또는 사용하면 라이선스 조항에 동의하는 것입니다 ["여기"](#).

이 저장소의 콘텐츠에서 파생 저작물을 생성 및/또는 공유하는 데는 특정 제한이 있습니다. 의 약관을 읽었는지 확인하십시오 ["라이선스"](#) 콘텐츠를 사용하기 전에. 모든 약관에 동의하지 않는 경우 이 리포지토리에서 콘텐츠를 액세스, 다운로드 또는 사용하지 마십시오.

준비가 되면 를 클릭합니다 ["자세한 AWX/Tower 배치 절차를 보려면 여기를 클릭하십시오"](#) 또는 ["CLI 배포는 여기에서 확인할 수 있습니다"](#).

단계별 배포 절차

AWX/Tower 구축 Oracle 19c Database

환경의 인벤토리, 그룹, 호스트 및 자격 증명을 생성합니다

이 섹션에서는 NetApp 자동화 솔루션을 사용하는 환경을 준비하기 위해 AWX/Ansible 타워에서 인벤토리, 그룹, 호스트, 액세스 자격 증명을 설정하는 방법을 설명합니다.

1. 인벤토리를 구성합니다.
 - a. Resources(리소스) → Inventory(인벤토리) → Add(추가) 로 이동하고 Add Inventory(재고 추가) 를 클릭합니다.
 - b. 이름 및 조직 세부 정보를 입력하고 저장 을 클릭합니다.
 - c. 재고 페이지에서 생성된 재고를 클릭합니다.
 - d. 재고 변수가 있는 경우 변수 필드에 붙여 넣습니다.
 - e. Groups 하위 메뉴로 이동하여 Add를 클릭합니다.
 - f. ONTAP 그룹 이름을 입력하고 그룹 변수(있는 경우)를 붙여 넣은 다음 저장을 클릭합니다.
 - g. Oracle에 대해 다른 그룹에 대해서도 이 프로세스를 반복합니다.
 - h. 생성된 ONTAP 그룹을 선택하고 Hosts 하위 메뉴로 이동한 다음 Add New Host를 클릭합니다.
 - i. ONTAP 클러스터 관리 IP의 IP 주소를 제공하고 호스트 변수(있는 경우)를 붙여 넣은 다음 저장 을 클릭합니다.
 - j. Oracle 그룹 및 Oracle 호스트 관리 IP/호스트 이름에 대해 이 프로세스를 반복해야 합니다.
2. 자격 증명 유형을 만듭니다. ONTAP와 관련된 솔루션의 경우 자격 증명 유형을 사용자 이름 및 암호 항목과 일치하도록 구성해야 합니다.
 - a. Administration → Credential Types로 이동한 후 Add를 클릭합니다.
 - b. 이름과 설명을 입력합니다.
 - c. 입력 구성에 다음 내용을 붙여 넣습니다.

```
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. 다음 내용을 주입기 구성에 붙여넣습니다.

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

1. 자격 증명을 구성합니다.

- a. Resources → Credentials로 이동한 후 Add를 클릭합니다.
- b. ONTAP의 이름과 조직 세부 정보를 입력합니다.
- c. ONTAP에 대해 만든 사용자 지정 자격 증명 유형을 선택합니다.
- d. 세부 정보 입력 아래에 사용자 이름, 암호 및 vsadmin_password를 입력합니다.
- e. 자격 증명으로 돌아가기 를 클릭하고 추가 를 클릭합니다.
- f. Oracle의 이름 및 조직 세부 정보를 입력합니다.
- g. 시스템 자격 증명 유형을 선택합니다.
- h. 유형 세부 정보 에서 Oracle 호스트의 사용자 이름 및 암호를 입력합니다.
- i. 올바른 권한 에스컬레이션 방법을 선택하고 사용자 이름과 암호를 입력합니다.

프로젝트를 만듭니다

1. Resources → Projects로 이동하여 Add를 클릭합니다.
 - a. 이름 및 조직 세부 정보를 입력합니다.
 - b. 소스 제어 자격 증명 유형 필드에서 Git 를 선택합니다.
 - c. 를 입력합니다 <https://github.com/NetApp-Automation/na_oracle19c_deploy.git> 소스 제어 URL입니다.
 - d. 저장 을 클릭합니다.
 - e. 소스 코드가 변경되면 프로젝트를 가끔 동기화해야 할 수 있습니다.

Oracle host_VAR을 구성합니다

이 섹션에 정의된 변수는 각 개별 Oracle 서버 및 데이터베이스에 적용됩니다.

1. 다음과 같은 내장된 Oracle hosts 변수 또는 host_vars 양식에 환경별 매개 변수를 입력합니다.



파란색 항목은 환경에 맞게 변경해야 합니다.

호스트 VAR 구성

```
#####
##### 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. 파란색 필드에 모든 변수를 입력합니다.
2. 변수 입력을 완료한 후 양식의 복사 버튼을 클릭하여 AWX 또는 타워로 전송할 모든 변수를 복사합니다.
3. AWX 또는 Tower로 돌아가서 Resources → Hosts 로 이동한 다음 Oracle 서버 구성 페이지를 선택하여 엽니다.
4. 세부 정보 탭에서 편집 을 클릭하고 1단계에서 복사한 변수를 YAML 탭의 변수 필드에 붙여 넣습니다.
5. 저장 을 클릭합니다.

6. 시스템에 있는 모든 추가 Oracle 서버에 대해 이 프로세스를 반복합니다.

글로벌 변수를 설정합니다

이 섹션에 정의된 변수는 모든 Oracle 호스트, 데이터베이스 및 ONTAP 클러스터에 적용됩니다.

1. 다음 임베디드 글로벌 변수 또는 VAR 양식에 환경별 매개 변수를 입력합니다.



파란색 항목은 환경에 맞게 변경해야 합니다.

```
#####
##### 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: "netapp123"
```

1. 파란색 필드에 모든 변수를 입력합니다.
2. 변수 입력을 완료한 후 양식의 복사 버튼을 클릭하여 AWX 또는 Tower로 전송할 모든 변수를 다음 작업 템플릿으로 복사합니다.

작업 템플릿을 구성하고 시작합니다.

1. 작업 템플릿을 작성합니다.
 - a. Resources → Templates → Add로 이동하여 Add Job Template을 클릭합니다.
 - b. 이름과 설명을 입력합니다
 - c. 작업 유형을 선택합니다. Run은 Playbook을 기반으로 시스템을 구성하고 Check는 실제로 시스템을 구성하지 않고 Playbook을 건조하게 실행합니다.
 - d. Playbook의 해당 인벤토리, 프로젝트, 플레이북 및 자격 증명을 선택합니다.

- e. All_Playbook.yml을 실행할 기본 플레이북으로 선택합니다.
- f. 4단계에서 복사한 글로벌 변수를 YAML 탭의 템플릿 변수 필드에 붙여 넣습니다.
- g. 작업 태그 필드에서 시작 시 프롬프트 표시 확인란을 선택합니다.
- h. 저장 을 클릭합니다.

2. 작업 템플릿을 시작합니다.

- a. 리소스 → 템플릿 으로 이동합니다.
- b. 원하는 템플릿을 클릭한 다음 실행을 클릭합니다.
- c. 작업 태그를 시작할 때 메시지가 나타나면 requirements_config 를 입력합니다. 작업 태그를 입력하려면 requirements_config 아래의 작업 태그 작성 줄을 클릭해야 할 수도 있습니다.



requirements_config 다른 역할을 실행할 올바른 라이브러리가 있는지 확인합니다.

1. 다음 을 클릭한 다음 시작 을 클릭하여 작업을 시작합니다.
2. 보기 → 작업 을 클릭하여 작업 출력 및 진행률을 모니터링합니다.
3. 작업 태그를 시작할 때 프롬프트가 표시되면 ONTAP_config를 입력합니다. ONTAP_config 바로 아래에 있는 "작업 태그" 생성 라인을 클릭하여 작업 태그를 입력해야 할 수 있습니다.
4. 다음 을 클릭한 다음 시작 을 클릭하여 작업을 시작합니다.
5. 보기 → 작업 을 클릭하여 작업 출력 및 진행률을 모니터링합니다
6. ONTAP_config 역할이 완료된 후 Linux_config에 대해 프로세스를 다시 실행하십시오.
7. 리소스 → 템플릿 으로 이동합니다.
8. 원하는 템플릿을 선택한 다음 실행을 클릭합니다.
9. Linux_config에서 작업 태그 유형을 시작할 때 메시지가 표시되면 Linux_config 바로 아래의 "작업 태그 생성" 행을 선택하여 작업 태그를 입력해야 할 수 있습니다.
10. 다음 을 클릭한 다음 시작 을 클릭하여 작업을 시작합니다.
11. 보기 → 작업 을 선택하여 작업 출력 및 진행률을 모니터링합니다.
12. Linux_config 역할이 완료된 후 ORACLE_config에 대해 프로세스를 다시 실행하십시오.
13. 리소스 → 템플릿 으로 이동합니다.
14. 원하는 템플릿을 선택한 다음 실행을 클릭합니다.
15. 작업 태그 시작 시 메시지가 표시되면 ORACLE_config 를 입력합니다. 작업 태그를 입력하려면 ORACLE_config 바로 아래에 있는 "작업 태그 생성" 행을 선택해야 할 수 있습니다.
16. 다음 을 클릭한 다음 시작 을 클릭하여 작업을 시작합니다.
17. 보기 → 작업 을 선택하여 작업 출력 및 진행률을 모니터링합니다.

동일한 **Oracle** 호스트에 추가 데이터베이스를 구축합니다

플레이북의 Oracle 부분은 실행 당 Oracle 서버에 단일 Oracle 컨테이너 데이터베이스를 생성합니다. 동일한 서버에 추가 컨테이너 데이터베이스를 만들려면 다음 단계를 완료하십시오.

1. host_vars 변수를 수정합니다.

- a. 2단계 - Configure Oracle host_VAR로 돌아갑니다.
 - b. Oracle SID를 다른 명명 문자열로 변경합니다.
 - c. 수신기 포트를 다른 번호로 변경합니다.
 - d. EM Express를 설치할 경우 EM Express 포트를 다른 번호로 변경하십시오.
 - e. 호스트 구성 세부 정보 탭의 Oracle 호스트 변수 필드에 수정된 호스트 변수를 복사하여 붙여 넣습니다.
2. ORACLE_config 태그만 사용하여 구축 작업 템플릿을 시작합니다.
 3. Oracle 서버에 Oracle 사용자로 로그인하고 다음 명령을 실행합니다.

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



설치가 예상대로 완료되고 Oracle DB가 시작되면 Oracle 프로세스가 나열됩니다

4. 데이터베이스에 로그인하여 다음 명령 세트로 생성된 DB 구성 설정과 PDB를 확인합니다.

```
[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

그러면 dNFS가 제대로 작동하고 있음을 확인할 수 있습니다.

5. 다음 명령을 사용하여 listener를 통해 데이터베이스에 연결하여 hte oracle listener 구성을 확인합니다. 적절한 수신기 포트 및 데이터베이스 서비스 이름으로 변경합니다.

```

[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

```

이를 통해 Oracle Listener가 제대로 작동하고 있음을 확인할 수 있습니다.

도움을 받을 수 있는 곳

툴킷에 대한 도움이 필요한 경우 에 가입하십시오 ["NetApp 솔루션 자동화 커뮤니티는 여유 채널을 지원합니다"](#) 질문 또는 질문을 게시할 수 있는 솔루션 자동화 채널을 찾아보십시오.

단계별 배포 절차

이 문서에서는 자동화 CLI(명령줄 인터페이스)를 사용하여 Oracle 19c를 구축하는 방법에 대해 자세히 설명합니다.

CLI 구축 Oracle 19c Database

이 섹션에서는 CLI를 사용하여 Oracle19c 데이터베이스를 준비하고 배포하는 데 필요한 단계를 설명합니다. 을(를) 검토했는지 확인합니다 ["시작하기 및 요구 사항 섹션을 참조하십시오"](#) 적절히 대비했습니다.

Oracle19c repo 를 다운로드합니다

1. Ansible 컨트롤러에서 다음 명령을 실행합니다.

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

2. 리포지토리를 다운로드한 후 디렉토리를 na_oracle19c_deploy <cd na_oracle19c_deploy> 로 변경합니다.

HOSTS 파일을 편집합니다

배포 전에 다음을 완료합니다.

1. 호스트 파일 na_oracle19c_deploy 디렉토리를 편집합니다.
2. [ONTAP] 아래에서 IP 주소를 클러스터 관리 IP로 변경합니다.
3. [Oracle] 그룹에서 Oracle 호스트 이름을 추가합니다. 호스트 이름은 DNS 또는 호스트 파일을 통해 IP 주소로 확인되거나 호스트에 지정되어야 합니다.
4. 이 단계를 완료한 후 변경 사항을 저장합니다.

다음 예에서는 호스트 파일을 보여 줍니다.

```
#ONTAP Host

[ontap]

"10.61.184.183"

#Oracle hosts

[oracle]

"rtpora01"

"rtpora02"
```

이 예에서는 Playbook을 실행하고 두 개의 Oracle DB 서버에 동시에 Oracle 19c를 구축합니다. 하나의 DB 서버만으로 테스트할 수도 있습니다. 이 경우 하나의 호스트 변수 파일만 구성하면 됩니다.



이 플레이북은 구축하는 Oracle 호스트 및 데이터베이스의 수에 관계없이 동일한 방식으로 실행됩니다.

host_name.yml 파일을 host_vars에서 편집합니다

각 Oracle 호스트에는 호스트별 변수를 포함하는 호스트 이름으로 식별되는 호스트 변수 파일이 있습니다. 호스트 이름을 지정할 수 있습니다. Host VAR Config 섹션에서 host_vars를 편집 및 복사하고 원하는 host_name.yml 파일에 붙여 넣습니다.



파란색 항목은 환경에 맞게 변경해야 합니다.

호스트 VAR 구성

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

```

# Add your Oracle Host
ansible_host: "10.61.180.15"

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

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

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

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

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

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

```

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

VAR.yml 파일을 편집합니다

VAR.yml 파일은 Oracle 구축을 위해 모든 환경 관련 변수(ONTAP, Linux 또는 Oracle)를 통합합니다.

1. VAR 섹션에서 변수를 편집 및 복사하고 해당 변수를 'VAR.yml' 파일에 붙여 넣습니다.

```
#####
##### 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"

```

플레이북을 실행합니다

필요한 환경 전제 조건을 완료하고 변수를 VAR.yml과 your_host.yml에 복사하면 이제 Playbook을 배포할 준비가 된 것입니다.



사용자 환경과 일치하도록 <username>을(를) 변경해야 합니다.

- 올바른 태그와 ONTAP 클러스터 사용자 이름을 전달하여 ONTAP 플레이북을 실행하십시오. 메시지가 표시되면 ONTAP Cluster 및 vsadmin의 암호를 입력합니다.

```

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

```

- Linux 플레이북을 실행하여 Linux 구축 부분을 실행합니다. 관리자 ssh 암호와 sudo 암호를 입력합니다.

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

3. Oracle Playbook을 실행하여 Oracle 구축 부분을 실행합니다. 관리자 ssh 암호와 sudo 암호를 입력합니다.

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

동일한 **Oracle** 호스트에 추가 데이터베이스를 구축합니다

플레이북의 Oracle 부분은 실행 당 Oracle 서버에 단일 Oracle 컨테이너 데이터베이스를 생성합니다. 동일한 서버에 추가 컨테이너 데이터베이스를 만들려면 다음 단계를 수행하십시오.

1. host_vars 변수를 수정합니다.
 - a. 3단계로 돌아가기 - host_vars에서 host_name.yml 파일을 편집합니다.
 - b. Oracle SID를 다른 명명 문자열로 변경합니다.
 - c. 수신기 포트를 다른 번호로 변경합니다.
 - d. EM Express를 설치한 경우 EM Express 포트를 다른 번호로 변경하십시오.
 - e. 수정된 호스트 변수를 복사하여 'host_vars' 아래의 Oracle 호스트 변수 파일에 붙여넣습니다.
2. 위의 에 나와 있는 것처럼 "ORACLE_config" 태그를 사용하여 플레이북을 실행합니다 [플레이북을 실행합니다](#).

Oracle 설치를 검증합니다

1. Oracle 서버에 Oracle 사용자로 로그인하고 다음 명령을 실행합니다.

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



설치가 예상대로 완료되고 Oracle DB가 시작되면 Oracle 프로세스가 나열됩니다

2. 데이터베이스에 로그인하여 다음 명령 세트로 생성된 DB 구성 설정과 PDB를 확인합니다.

```

[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

```

그러면 dNFS가 제대로 작동하고 있음을 확인할 수 있습니다.

3. 다음 명령을 사용하여 listener를 통해 데이터베이스에 연결하여 hte oracle listener 구성을 확인합니다. 적절한 수신기 포트 및 데이터베이스 서비스 이름으로 변경합니다.

```

[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

```

이를 통해 Oracle Listener가 제대로 작동하고 있음을 확인할 수 있습니다.

도움을 받을 수 있는 곳

툴킷에 대한 도움이 필요한 경우 에 가입하십시오 ["NetApp 솔루션 자동화 커뮤니티는 여유 채널을 지원합니다"](#) 질문 또는 질문을 게시할 수 있는 솔루션 자동화 채널을 찾아보십시오.

솔루션 개요

Oracle 데이터베이스용 자동화된 데이터 보호

조직은 효율성을 높이고, 구현을 앞당기고, 수동 작업을 줄이기 위해 환경을 자동화하고 있습니다. Ansible과 같은 구성 관리 툴을 사용하여 엔터프라이즈 데이터베이스 운영을 간소화하고 있습니다. 이 솔루션에서는 Ansible을 사용하여 NetApp ONTAP을 통해 Oracle의 데이터 보호를 자동화하는 방법을 보여줍니다. 스토리지 관리자, 시스템 관리자 및 DBA가 오프사이트 데이터 센터 또는 퍼블릭 클라우드에 대한 데이터 복제를 일관되고 신속하게 설정할 수 있도록 함으로써 다음과 같은 이점을 얻을 수 있습니다.

- 설계 복잡성과 인적 오류를 제거하고 반복 가능한 일관된 구축 및 모범 사례를 구현합니다
- 클러스터 간 복제, CVO 인스턴스화 및 Oracle 데이터베이스 복구 구성 시간 단축
- 데이터베이스 관리자, 시스템 및 스토리지 관리자의 생산성 향상
- DR 시나리오의 테스트를 쉽게 수행할 수 있는 데이터베이스 복구 워크플로우를 제공합니다.

NetApp은 고객에게 검증된 Ansible 모듈과 역할을 제공하여 Oracle 데이터베이스 환경의 구축, 구성, 라이프사이클 관리를 가속합니다. 이 솔루션은 다음을 지원하기 위한 지침 및 Ansible 플레이북 코드를 제공합니다.

온프레미스에서 온프레미스 복제까지

- 소스와 대상에 대한 인터클러스터 LIF를 만듭니다
- 클러스터 및 SVM 피어링 설정
- Oracle 볼륨의 SnapMirror를 생성하고 초기화합니다
- Oracle 바이너리, 데이터베이스 및 로그용 AWX/Tower를 통해 복제 일정을 생성합니다
- 대상에서 Oracle DB를 복원하고 데이터베이스를 온라인 상태로 전환합니다

온프레미스에서 **AWS**의 **CVO**로

- AWS 커넥터를 생성합니다
- AWS에서 CVO 인스턴스를 생성합니다
- Cloud Manager에 온프레미스 클러스터 추가
- 소스에 대한 인터클러스터 LIF를 만듭니다
- 클러스터 및 SVM 피어링 설정
- Oracle 볼륨의 SnapMirror를 생성하고 초기화합니다
- Oracle 바이너리, 데이터베이스 및 로그용 AWX/Tower를 통해 복제 일정을 생성합니다
- 대상에서 Oracle DB를 복원하고 데이터베이스를 온라인 상태로 전환합니다

준비가 되면 를 클릭합니다 "[여기 에서 솔루션 시작 을 확인하십시오](#)".

시작하기

이 솔루션은 AWX/Tower 환경에서 실행되도록 설계되었습니다.

AWX/타워

AWX/Tower 환경의 경우 ONTAP 클러스터 관리 및 Oracle 서버(IP 및 호스트 이름)의 인벤토리 생성, 자격 증명 생성, NetApp Automation GitHub에서 Ansible 코드를 가져오는 프로젝트 구성, 자동화를 시작하는 작업 템플릿 등이 있습니다.

1. 이 솔루션은 프라이빗 클라우드 시나리오(사내-사내), 하이브리드 클라우드(사내-퍼블릭 클라우드 간 Cloud Volumes ONTAP[CVO])에서 실행되도록 설계되었습니다.
2. 사용자 환경에 맞는 변수를 입력하고 이를 복사하여 작업 템플릿의 추가 VAR 필드에 붙여 넣습니다.
3. 추가 VAR이 작업 템플릿에 추가되면 자동화를 시작할 수 있습니다.
4. 자동화는 3단계(설정, Oracle 바이너리, 데이터베이스, 로그 및 로그용 복제 일정)와 DR 사이트에서 데이터베이스를 복구하기 위한 4단계로 구성됩니다.
5. CVO 데이터 보호에 필요한 키 및 토큰을 얻기 위한 자세한 지침은 을 참조하십시오 "[CVO 및 Connector 구축을 위한 사전 요구 사항을 수집합니다](#)"

요구 사항

strong class="big">온프레미스|

방법입니다	요구 사항
* Ansible 환경 *	AWX/타워
	Ansible v. 2.10 이상
	Python 3
	Python 라이브러리 - NetApp-lib-xmltodict-jmespath
* ONTAP *	ONTAP 버전 9.8 이상
	두 개의 데이터 애그리게이트
	NFS VLAN 및 ifgrp가 생성되었습니다
* Oracle 서버 *	RHEL 7/8
	Oracle Linux 7/8
	NFS, 공용 및 선택적 관리를 위한 네트워크 인터페이스
	소스의 기존 Oracle 환경과 대상(DR 사이트 또는 퍼블릭 클라우드)의 동급 Linux 운영 체제

strong class="big">CVO

방법입니다	요구 사항
* Ansible 환경 *	AWX/타워
	Ansible v. 2.10 이상
	Python 3
	Python 라이브러리 - NetApp-lib-xmltodict-jmespath
* ONTAP *	ONTAP 버전 9.8 이상
	두 개의 데이터 애그리게이트
	NFS VLAN 및 ifgrp가 생성되었습니다
* Oracle 서버 *	RHEL 7/8
	Oracle Linux 7/8
	NFS, 공용 및 선택적 관리를 위한 네트워크 인터페이스
	소스의 기존 Oracle 환경과 대상(DR 사이트 또는 퍼블릭 클라우드)의 동급 Linux 운영 체제
	Oracle EC2 인스턴스에 적절한 스왑 공간을 설정합니다. 기본적으로 일부 EC2 인스턴스는 0 스왑으로 구축됩니다
* Cloud Manager/AWS *	AWS 액세스/비밀 키
	NetApp Cloud Manager 계정
	NetApp Cloud Manager 업데이트 토큰

온프레미스

이 자동 배포는 3개의 개별 역할로 구성된 단일 Ansible 플레이북을 통해 설계되었습니다. 역할은 ONTAP, Linux 및 Oracle 구성을 위한 것입니다. 다음 표에서는 자동화되고 있는 작업을 설명합니다.

플레이북	작업
* ONTAP_설정 *	ONTAP 환경 사전 점검
	소스 클러스터에서 Intercluster LIF 생성(선택 사항)
	대상 클러스터에 Intercluster LIF 생성(선택 사항)
	클러스터 및 SVM 피어링 생성
	대상 SnapMirror 생성 및 지정된 Oracle 볼륨의 초기화
* ora_replication_cg *	/etc/oratab의 각 데이터베이스에 대해 백업 모드를 활성화합니다
	Oracle 바이너리 및 데이터베이스 볼륨의 스냅샷을 생성합니다
	SnapMirror가 업데이트되었습니다
	/etc/oratab에서 각 데이터베이스의 백업 모드를 해제합니다
* ora_replication_log *	/etc/oratab에 있는 각 데이터베이스의 현재 로그를 전환합니다
	Oracle Log 볼륨의 스냅샷입니다
	SnapMirror가 업데이트되었습니다
* ora_RECOVERY *	SnapMirror를 꺾습니다
	NFS를 사용하도록 설정하고 타겟에서 Oracle 볼륨에 대한 접합 경로를 생성합니다
	DR Oracle 호스트를 구성합니다
	Oracle 볼륨을 마운트하고 확인합니다
	Oracle 데이터베이스 복구 및 시작

CVO

이 자동 배포는 3개의 개별 역할로 구성된 단일 Ansible 플레이북을 통해 설계되었습니다. 역할은 ONTAP, Linux 및 Oracle 구성을 위한 것입니다. 다음 표에서는 자동화되고 있는 작업을 설명합니다.

플레이북	작업
* cvo_setup *	환경 사전 점검
	AWS 구성/AWS 액세스 키 ID/비밀 키/기본 지역
	AWS 역할 생성
	AWS에서 NetApp Cloud Manager Connector 인스턴스 생성
	AWS에서 CVO(Cloud Volumes ONTAP) 인스턴스 생성
	NetApp Cloud Manager에 온프레미스 소스 ONTAP 클러스터를 추가하십시오
	대상 SnapMirror 생성 및 지정된 Oracle 볼륨의 초기화

플레이북	작업
* ora_replication_cg *	/etc/oratab의 각 데이터베이스에 대해 백업 모드를 활성화합니다 Oracle 바이너리 및 데이터베이스 볼륨의 스냅샷을 생성합니다 SnapMirror가 업데이트되었습니다 /etc/oratab에서 각 데이터베이스의 백업 모드를 해제합니다
* ora_replication_log *	/etc/oratab에 있는 각 데이터베이스의 현재 로그를 전환합니다 Oracle Log 볼륨의 스냅샷입니다 SnapMirror가 업데이트되었습니다
* ora_RECOVERY *	SnapMirror를 꺾습니다 NFS를 사용하도록 설정하고 타겟 CVO에서 Oracle 볼륨의 접합 경로를 생성합니다 DR Oracle 호스트를 구성합니다 Oracle 볼륨을 마운트하고 확인합니다 Oracle 데이터베이스 복구 및 시작

기본 매개변수

자동화를 간소화하기 위해 필요한 많은 Oracle 매개 변수를 기본값으로 사전 설정하였습니다. 일반적으로 대부분의 배포에서 기본 매개 변수를 변경할 필요는 없습니다. 고급 사용자는 기본 매개 변수를 주의 깊게 변경할 수 있습니다. 기본 매개 변수는 각 역할 폴더의 defaults 디렉토리에 있습니다.

라이선스

GitHub 리포지토리에 설명된 대로 라이선스 정보를 읽어야 합니다. 이 리포지토리의 콘텐츠에 액세스, 다운로드, 설치 또는 사용하면 라이선스 조항에 동의하는 것입니다 ["여기"](#).

이 저장소의 콘텐츠에서 파생 저작물을 생성 및/또는 공유하는 데는 특정 제한이 있습니다. 의 약관을 읽었는지 확인하십시오 ["라이선스"](#) 콘텐츠를 사용하기 전에. 모든 약관에 동의하지 않는 경우 이 리포지토리에서 콘텐츠를 액세스, 다운로드 또는 사용하지 마십시오.

준비가 되면 를 클릭합니다 ["자세한 AWX/Tower 절차를 보려면 여기를 클릭하십시오"](#).

단계별 배포 절차

AWX/Tower Oracle 데이터 보호

환경에 대한 인벤토리, 그룹, 호스트 및 자격 증명을 생성합니다

이 섹션에서는 NetApp 자동화 솔루션을 사용하는 환경을 준비하기 위해 AWX/Ansible 타워에서 인벤토리, 그룹, 호스트, 액세스 자격 증명을 설정하는 방법을 설명합니다.

1. 인벤토리를 구성합니다.

- a. Resources(리소스) → Inventory(인벤토리) → Add(추가) 로 이동하고 Add Inventory(재고 추가) 를 클릭합니다.

- b. 이름 및 조직 세부 정보를 입력하고 저장 을 클릭합니다.
- c. 재고 페이지에서 생성된 재고를 클릭합니다.
- d. Groups 하위 메뉴로 이동하여 Add를 클릭합니다.
- e. 첫 번째 그룹에 대해 Oracle 이름을 입력하고 저장 을 클릭합니다.
- f. DR_ORACLE이라는 두 번째 그룹에 대해 이 프로세스를 반복합니다.
- g. 생성된 Oracle 그룹을 선택하고 Hosts 하위 메뉴로 이동한 다음 Add New Host를 클릭합니다.
- h. 소스 Oracle 호스트 관리 IP의 IP 주소를 제공하고 Save를 클릭합니다.
- i. DR_Oracle 그룹에 대해 이 프로세스를 반복하고 DR/대상 Oracle 호스트의 관리 IP/호스트 이름을 추가해야 합니다.



다음은 ONTAP 온프레미스 또는 AWS의 CVO에 대한 자격 증명 유형 및 자격 증명을 생성하는 지침입니다.

사내

1. 자격 증명을 구성합니다.
2. 자격 증명 형식을 만듭니다. ONTAP와 관련된 솔루션의 경우 자격 증명 유형을 사용자 이름 및 암호 항목과 일치하도록 구성해야 합니다.
 - a. Administration → Credential Types로 이동한 후 Add를 클릭합니다.
 - b. 이름과 설명을 입력합니다.
 - c. 입력 구성에 다음 내용을 붙여 넣습니다.

```
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. Injector Configuration(주입기 구성)에 다음 내용을 붙여넣고 Save(저장)를 클릭합니다.

```
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. ONTAP에 대한 자격 증명을 생성합니다
 - a. Resources → Credentials로 이동한 후 Add를 클릭합니다.
 - b. ONTAP 자격 증명에 대한 이름과 조직 세부 정보를 입력합니다
 - c. 이전 단계에서 만든 자격 증명 유형을 선택합니다.
 - d. 유형 세부 정보 에서 소스 및 대상 클러스터에 대한 사용자 이름 및 암호를 입력합니다.
 - e. 저장 을 클릭합니다
4. Oracle에 대한 자격 증명을 생성합니다

- a. Resources → Credentials로 이동한 후 Add를 클릭합니다.
- b. Oracle의 이름 및 조직 세부 정보를 입력합니다
- c. 시스템 자격 증명 유형을 선택합니다.
- d. 유형 세부 정보 에서 Oracle 호스트의 사용자 이름 및 암호를 입력합니다.
- e. 올바른 권한 에스컬레이션 방법을 선택하고 사용자 이름과 암호를 입력합니다.
- f. 저장 을 클릭합니다
- g. DR_Oracle 호스트에 대해 다른 자격 증명에 대해 필요한 경우 프로세스를 반복합니다.

CVO

1. 자격 증명을 구성합니다.
2. 자격 증명 유형을 만듭니다. ONTAP와 관련된 솔루션의 경우 자격 증명 유형을 사용자 이름 및 암호 항목과 일치하도록 구성해야 합니다. 또한 Cloud Central 및 AWS에 대한 항목을 추가합니다.
 - a. Administration → Credential Types로 이동한 후 Add를 클릭합니다.
 - b. 이름과 설명을 입력합니다.
 - c. 입력 구성에 다음 내용을 붙여 넣습니다.


```
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. Injector Configuration(주입기 구성)에 다음 내용을 붙여넣고 Save(저장)를 클릭합니다.

```

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. ONTAP/CVO/AWS에 대한 자격 증명을 생성합니다

- a. Resources → Credentials로 이동한 후 Add를 클릭합니다.
- b. ONTAP 자격 증명에 대한 이름과 조직 세부 정보를 입력합니다
- c. 이전 단계에서 만든 자격 증명 유형을 선택합니다.
- d. 유형 세부 정보 아래에 소스 및 CVO 클러스터, Cloud Central/Manager, AWS 액세스/비밀 키 및 Cloud Central 업데이트 토큰의 사용자 이름 및 암호를 입력합니다.
- e. 저장 을 클릭합니다

4. Oracle에 대한 자격 증명 생성(소스)

- a. Resources → Credentials로 이동한 후 Add를 클릭합니다.
- b. Oracle 호스트의 이름 및 조직 세부 정보를 입력합니다
- c. 시스템 자격 증명 유형을 선택합니다.
- d. 유형 세부 정보 에서 Oracle 호스트의 사용자 이름 및 암호를 입력합니다.
- e. 올바른 권한 에스컬레이션 방법을 선택하고 사용자 이름과 암호를 입력합니다.
- f. 저장 을 클릭합니다

5. Oracle Destination에 대한 자격 증명을 생성합니다

- a. Resources → Credentials로 이동한 후 Add를 클릭합니다.
- b. DR Oracle 호스트의 이름 및 조직 세부 정보를 입력합니다
- c. 시스템 자격 증명 유형을 선택합니다.
- d. 세부 정보 유형 에 사용자 이름(EC2-USER 또는 기본값에서 변경한 경우 해당 입력) 및 SSH 개인 키를 입력합니다
- e. 올바른 권한 에스컬레이션 방법(sudo)을 선택하고 필요한 경우 사용자 이름과 암호를 입력합니다.
- f. 저장 을 클릭합니다

프로젝트를 만듭니다

1. Resources → Projects로 이동하여 Add를 클릭합니다.
 - a. 이름 및 조직 세부 정보를 입력합니다.
 - b. 소스 제어 자격 증명 유형 필드에서 Git 를 선택합니다.
 - c. 를 입력합니다 <https://github.com/NetApp-Automation/na_oracle19c_data_protection.git> 소스 제어 URL입니다.
 - d. 저장 을 클릭합니다.
 - e. 소스 코드가 변경되면 프로젝트를 가끔 동기화해야 할 수 있습니다.

글로벌 변수를 설정합니다

이 섹션에 정의된 변수는 모든 Oracle 호스트, 데이터베이스 및 ONTAP 클러스터에 적용됩니다.

1. 다음 임베디드 글로벌 변수 또는 VAR 양식에 환경별 매개 변수를 입력합니다.



파란색 항목은 환경에 맞게 변경해야 합니다.

사내

```
# 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"
```

자동화 플레이북

4개의 개별 플레이북을 실행해야 합니다.

1. 온프레미스 또는 CVO 환경 설정을 위한 플레이북
2. Oracle 바이너리 및 데이터베이스 복제를 위한 일정 계획
3. 일정에 따라 Oracle 로그를 복제하는 데 필요한 Playbook
4. 타겟 호스트에서 데이터베이스를 복구하는 플레이북입니다

ONTAP/CVO 설정

ONTAP 및 CVO 설정

- 작업 템플릿을 구성하고 시작합니다. *

1. 작업 템플릿을 작성합니다.

- a. Resources → Templates → Add로 이동하여 Add Job Template을 클릭합니다.
- b. ONTAP/CVO 설정의 이름을 입력합니다
- c. 작업 유형을 선택합니다. 실행 은 Playbook을 기반으로 시스템을 구성합니다.
- d. Playbook의 해당 인벤토리, 프로젝트, 플레이북 및 자격 증명을 선택합니다.
- e. 사내 환경의 경우 ONTAP_setup.yml 플레이북을 선택하고 CVO 인스턴스로 복제할 때 cvo_setup.yml을 선택합니다.
- f. 4단계에서 복사한 글로벌 변수를 YAML 탭의 템플릿 변수 필드에 붙여 넣습니다.
- g. 저장 을 클릭합니다.

2. 작업 템플릿을 시작합니다.

- a. 리소스 → 템플릿 으로 이동합니다.
- b. 원하는 템플릿을 클릭한 다음 실행을 클릭합니다.



이 템플릿을 사용하여 다른 Playbook에 복사할 것입니다.

바이너리 및 데이터베이스 볼륨의 복제입니다

바이너리 및 데이터베이스 복제 플레이북 예약

- 작업 템플릿을 구성하고 시작합니다. *

1. 이전에 생성한 작업 템플릿을 복사합니다.

- a. 리소스 → 템플릿 으로 이동합니다.
- b. ONTAP/CVO 설정 템플릿을 찾은 후 Copy Template을 마우스 오른쪽 버튼으로 클릭합니다
- c. 복사된 템플릿에서 템플릿 편집 을 클릭하고 이름을 바이너리 및 데이터베이스 복제 플레이북으로 변경합니다.
- d. 템플릿에 대해 동일한 재고, 프로젝트, 자격 증명을 유지합니다.
- e. 실행할 플레이북으로 ora_replication_cg.yml을 선택합니다.
- f. 변수는 동일하게 유지되지만 CVO 클러스터 IP는 dst_cluster_ip 변수에 설정되어야 합니다.
- g. 저장 을 클릭합니다.

2. 작업 템플릿을 예약합니다.

- a. 리소스 → 템플릿 으로 이동합니다.
- b. 바이너리 및 데이터베이스 복제 플레이북 템플릿을 클릭한 다음, 최상위 옵션 세트에서 일정을 클릭합니다.
- c. 추가 를 클릭하고 바이너리 및 데이터베이스 복제에 대한 이름 일정 추가 를 클릭한 다음 시간 시작 시 시작 날짜/시간을 선택하고 로컬 표준 시간대를 선택한 다음 실행 빈도 를 선택합니다. 실행 빈도는

대개 SnapMirror 복제가 업데이트됩니다.



로그 볼륨 복제에 대해 별도의 일정이 생성되므로 보다 빈번한 케이던스로 복제할 수 있습니다.

로그 볼륨의 복제입니다

로그 복제 플레이북 예약

- 작업 템플릿을 구성하고 시작합니다. *

1. 이전에 생성한 작업 템플릿을 복사합니다.

- a. 리소스 → 템플릿 으로 이동합니다.
- b. ONTAP/CVO 설정 템플릿을 찾은 후 Copy Template을 마우스 오른쪽 버튼으로 클릭합니다
- c. 복사된 템플릿에서 템플릿 편집 을 클릭하고 이름을 로그 복제 플레이북 으로 변경합니다.
- d. 템플릿에 대해 동일한 재고, 프로젝트, 자격 증명을 유지합니다.
- e. 실행할 플레이북으로 ora_replication_logs.yml을 선택합니다.
- f. 변수는 동일하게 유지되지만 CVO 클러스터 IP는 dst_cluster_ip 변수에 설정되어야 합니다.
- g. 저장 을 클릭합니다.

2. 작업 템플릿을 예약합니다.

- a. 리소스 → 템플릿 으로 이동합니다.
- b. Log Replication Playbook 템플릿을 클릭한 다음 최상위 옵션 집합에서 Schedules를 클릭합니다.
- c. 추가, 로그 복제에 대한 이름 스케줄 추가, 시간 시작 시 시작 날짜/시간 선택, 로컬 시간대 선택 및 실행 빈도 를 차례로 클릭합니다. 실행 빈도는 대개 SnapMirror 복제가 업데이트됩니다.



마지막 1시간 단위 업데이트까지 복구할 수 있도록 매 시간마다 업데이트되도록 로그 스케줄을 설정하는 것이 좋습니다.

데이터베이스 복원 및 복구

로그 복제 플레이북 예약

- 작업 템플릿을 구성하고 시작합니다. *

1. 이전에 생성한 작업 템플릿을 복사합니다.

- a. 리소스 → 템플릿 으로 이동합니다.
- b. ONTAP/CVO 설정 템플릿을 찾은 후 Copy Template을 마우스 오른쪽 버튼으로 클릭합니다
- c. 복사된 템플릿에서 템플릿 편집 을 클릭하고 이름을 복원 및 복구 Playbook 으로 변경합니다.
- d. 템플릿에 대해 동일한 재고, 프로젝트, 자격 증명을 유지합니다.
- e. 실행할 플레이북으로 ora_recovery.yml을 선택합니다.
- f. 변수는 동일하게 유지되지만 CVO 클러스터 IP는 dst_cluster_ip 변수에 설정되어야 합니다.
- g. 저장 을 클릭합니다.



이 플레이북은 원격 사이트에서 데이터베이스를 복원할 준비가 될 때까지 실행할 수 없습니다.

Oracle 데이터베이스 복구 중

1. 사내 운영 Oracle 데이터베이스 데이터 볼륨은 NetApp SnapMirror 복제를 통해 2차 데이터 센터의 이중 ONTAP 클러스터나 퍼블릭 클라우드의 Cloud Volume ONTAP로 보호됩니다. 완전히 구성된 재해 복구 환경에서는 2차 데이터 센터 또는 퍼블릭 클라우드의 복구 컴퓨팅 인스턴스가 대기 상태이며 재해 발생 시 운영 데이터베이스를 복구할 수 있는 준비가 되어 있습니다. 대기 컴퓨팅 인스턴스는 OS 커널 패치에서 parallel 업데이트를 실행하거나 잠금 단계에서 업그레이드를 실행하여 온프레미스 인스턴스와 동기화된 상태를 유지합니다.
2. 이 솔루션에서 Oracle 바이너리 볼륨은 타겟 인스턴스에 복제되어 타겟 인스턴스에 마운트하여 Oracle 소프트웨어 스택을 실행하는 것으로 나타났습니다. Oracle을 복구하는 이러한 접근 방식은 재해가 발생한 마지막 순간에 Oracle을 새로 설치하는 데 비해 많은 이점을 제공합니다. 이 제품은 Oracle 설치가 현재 온프레미스 프로덕션 소프트웨어 설치 및 패치 수준 등과 완벽하게 동기화되도록 보장합니다. 그러나 소프트웨어 라이선스가 Oracle과 어떻게 구성되어 있는지에 따라 복구 사이트에서 복제된 Oracle 바이너리 볼륨에 대한 소프트웨어 라이선스가 추가로 부여되거나 적용되지 않을 수 있습니다. 사용자는 동일한 접근 방식을 사용하기 전에 소프트웨어 라이선스 담당자에게 잠재적인 Oracle 라이선스 요구 사항을 평가하는 것이 좋습니다.
3. 대상의 대기 Oracle 호스트는 Oracle 필수 구성 요소 구성으로 구성됩니다.
4. SnapMirror가 손상되고 볼륨이 쓰기 가능으로 만들어져 대기 Oracle 호스트에 마운트됩니다.
5. Oracle 복구 모듈은 모든 DB 볼륨이 대기 컴퓨팅 인스턴스에 마운트된 후 복구 사이트에서 Oracle을 복구 및 시작하는 다음과 같은 작업을 수행합니다.
 - a. 제어 파일 동기화: 중요한 데이터베이스 제어 파일을 보호하기 위해 서로 다른 데이터베이스 볼륨에 중복 Oracle 제어 파일을 구축했습니다. 하나는 데이터 볼륨에 있고 다른 하나는 로그 볼륨에 있습니다. 데이터 및 로그 볼륨은 서로 다른 빈도로 복제되므로 복구 시 동기화되지 않습니다.
 - b. Oracle 바이너리 다시 연결: Oracle 바이너리가 새 호스트로 재배포되므로 재링크가 필요합니다.
 - c. Oracle 데이터베이스 복구: 복구 메커니즘은 Oracle 로그 볼륨에서 마지막으로 사용 가능한 아카이브 로그의 마지막 시스템 변경 번호를 제어 파일에서 검색하고 Oracle 데이터베이스를 복구하여 장애 발생 시 DR 사이트에 복제할 수 있는 모든 비즈니스 트랜잭션을 복구합니다. 그런 다음 복구 사이트에서 사용자 연결 및 비즈니스 트랜잭션을 수행할 수 있도록 데이터베이스가 새로 도입되었습니다.



복구 플레이북을 실행하기 전에 /etc/oratab 및 /etc/orainst.loc을 소스 Oracle 호스트에서 대상 호스트로 복제해야 합니다

TR-4794: NetApp EF-Series 기반의 Oracle 데이터베이스

Mitch Blackburn, NetApp Ebin Kadavy

TR-4794는 스토리지 관리자와 데이터베이스 관리자가 NetApp EF-Series 스토리지에 Oracle을 성공적으로 구축할 수 있도록 설계되었습니다.

["TR-4794: NetApp EF-Series 기반의 Oracle 데이터베이스"](#)

저작권 정보

Copyright © 2024 NetApp, Inc. All Rights Reserved. 미국에서 인쇄된 본 문서의 어떠한 부분도 저작권 소유자의 사전 서면 승인 없이는 어떠한 형식이나 수단(복사, 녹음, 녹화 또는 전자 검색 시스템에 저장하는 것을 비롯한 그래픽, 전자적 또는 기계적 방법)으로도 복제될 수 없습니다.

NetApp이 저작권을 가진 자료에 있는 소프트웨어에는 아래의 라이선스와 고지사항이 적용됩니다.

본 소프트웨어는 NetApp에 의해 '있는 그대로' 제공되며 상품성 및 특정 목적에의 적합성에 대한 명시적 또는 묵시적 보증을 포함하여(이에 제한되지 않음) 어떠한 보증도 하지 않습니다. NetApp은 대체품 또는 대체 서비스의 조달, 사용 불능, 데이터 손실, 이익 손실, 영업 중단을 포함하여(이에 국한되지 않음), 이 소프트웨어의 사용으로 인해 발생하는 모든 직접 및 간접 손해, 우발적 손해, 특별 손해, 징벌적 손해, 결과적 손해의 발생에 대하여 그 발생 이유, 책임론, 계약 여부, 엄격한 책임, 불법 행위(과실 또는 그렇지 않은 경우)와 관계없이 어떠한 책임도 지지 않으며, 이와 같은 손실의 발생 가능성이 통지되었다 하더라도 마찬가지입니다.

NetApp은 본 문서에 설명된 제품을 언제든지 예고 없이 변경할 권리를 보유합니다. NetApp은 NetApp의 명시적인 서면 동의를 받은 경우를 제외하고 본 문서에 설명된 제품을 사용하여 발생하는 어떠한 문제에도 책임을 지지 않습니다. 본 제품의 사용 또는 구매의 경우 NetApp에서는 어떠한 특허권, 상표권 또는 기타 지적 재산권이 적용되는 라이선스도 제공하지 않습니다.

본 설명서에 설명된 제품은 하나 이상의 미국 특허, 해외 특허 또는 출원 중인 특허로 보호됩니다.

제한적 권리 표시: 정부에 의한 사용, 복제 또는 공개에는 DFARS 252.227-7013(2014년 2월) 및 FAR 52.227-19(2007년 12월)의 기술 데이터-비상업적 품목에 대한 권리(Rights in Technical Data -Noncommercial Items) 조항의 하위 조항 (b)(3)에 설명된 제한사항이 적용됩니다.

여기에 포함된 데이터는 상업용 제품 및/또는 상업용 서비스(FAR 2.101에 정의)에 해당하며 NetApp, Inc.의 독점 자산입니다. 본 계약에 따라 제공되는 모든 NetApp 기술 데이터 및 컴퓨터 소프트웨어는 본질적으로 상업용이며 개인 비용만으로 개발되었습니다. 미국 정부는 데이터가 제공된 미국 계약과 관련하여 해당 계약을 지원하는 데에만 데이터에 대한 전 세계적으로 비독점적이고 양도할 수 없으며 재사용이 불가능하며 취소 불가능한 라이선스를 제한적으로 가집니다. 여기에 제공된 경우를 제외하고 NetApp, Inc.의 사전 서면 승인 없이는 이 데이터를 사용, 공개, 재생산, 수정, 수행 또는 표시할 수 없습니다. 미국 국방부에 대한 정부 라이선스는 DFARS 조항 252.227-7015(b)(2014년 2월)에 명시된 권한으로 제한됩니다.

상표 정보

NETAPP, NETAPP 로고 및 <http://www.netapp.com/TM>에 나열된 마크는 NetApp, Inc.의 상표입니다. 기타 회사 및 제품 이름은 해당 소유자의 상표일 수 있습니다.