NFS 기반 ONTAP용 Oracle 19c의 자동 구축

NetApp Solutions

NetApp
April 20, 2024

This PDF was generated from https://docs.netapp.com/ko-kr/netapp-solutions/databases/getting_started_requirements.html on April 20, 2024. Always check docs.netapp.com for the latest.

목차

솔루션 개요 1
NFS에서 ONTAP용 Oracle19c의 자동 배포 1
시작하기 1
단계별 배포 절차 4
단계별 배포 절차 13

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-xmltodictt-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
* 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 사용자에 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
```

라이센스

GitHub 리포지토리에 설명된 대로 라이센스 정보를 읽어야 합니다. 이 리포지토리의 콘텐츠에 액세스, 다운로드, 설치 또는 사용하면 라이선스 조항에 동의하는 것입니다 "여기".

이 저장소의 컨텐츠에서 파생 저작물을 생성 및/또는 공유하는 데는 특정 제한이 있습니다. 의 약관을 읽었는지 확인하십시오 "라이센스" 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 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"}
```

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 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. 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_vars**에서 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.

1. 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
```

2. 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.

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 라이센스와 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 라이센스도 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 라이센스를 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 라이센스는 DFARS 조항 252.227-7015(b)(2014년 2월)에 명시된 권한으로 제한됩니다.

상표 정보

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