



Implementazione di Oracle Database

NetApp Solutions

NetApp
September 10, 2024

This PDF was generated from <https://docs.netapp.com/it-it/netapp-solutions/../../../../ontap-apps-dbs/oracle/oracle-overview.html> on September 10, 2024. Always check docs.netapp.com for the latest.

Sommario

Implementazione di Oracle Database	1
Panoramica della soluzione	1

Implementazione di Oracle Database

Panoramica della soluzione

In questa pagina viene descritto il metodo automatizzato per l'implementazione di Oracle19c sullo storage NetApp ONTAP.

Implementazione automatica di Oracle19c per ONTAP su NFS

Le organizzazioni stanno automatizzando i propri ambienti per ottenere efficienze, accelerare le implementazioni e ridurre l'impegno manuale. I tool di gestione della configurazione come Ansible vengono utilizzati per ottimizzare le operazioni dei database aziendali. In questa soluzione, dimostreremo come utilizzare Ansible per automatizzare il provisioning e la configurazione di Oracle 19c con NetApp ONTAP. Consentendo agli amministratori dello storage, agli amministratori di sistema e ai DBA di implementare in modo coerente e rapido nuovo storage, configurare server di database e installare il software Oracle 19c, otterrete i seguenti vantaggi:

- Elimina le complessità di progettazione e gli errori umani e implementa un'implementazione coerente e ripetibile e Best practice
- Riduzione dei tempi di provisioning dello storage, configurazione degli host DB e installazione di Oracle
- Aumenta la produttività di amministratori di database, sistemi e amministratori dello storage
- Scalabilità di storage e database con facilità

NetApp offre ai clienti i moduli e i ruoli Ansible validati per accelerare l'implementazione, la configurazione e la gestione del ciclo di vita del tuo ambiente di database Oracle. Questa soluzione fornisce istruzioni e codice del playbook Ansible per aiutarti a:

- Creare e configurare lo storage NFS ONTAP per il database Oracle
- Installare Oracle 19c su RedHat Enterprise Linux 7/8 o Oracle Linux 7/8
- Configurare Oracle 19c sullo storage NFS ONTAP

Per ulteriori dettagli o per iniziare, consulta i video di panoramica riportati di seguito.

Implementazioni AWX/tower

Parte 1: Introduzione, requisiti, dettagli di automazione e configurazione iniziale AWX/Tower

[Implementazione AWX](#)

Parte 2: Variabili ed esecuzione del Playbook

[AWX Playbook Run](#)

Implementazione della CLI

Parte 1: Guida introduttiva, requisiti, dettagli di automazione e configurazione host di Ansible Control

[Implementazione della CLI](#)

Parte 2: Variabili ed esecuzione del Playbook

Per iniziare

Questa soluzione è stata progettata per essere eseguita in un ambiente AWX/Tower o da CLI su un host di controllo Ansible.

AWX/Tower

Per gli ambienti AWX/tower, viene fornita una guida alla creazione di un inventario della gestione del cluster ONTAP e del server Oracle (IP e nomi host), alla creazione di credenziali, alla configurazione di un progetto che estrae il codice Ansible da NetApp Automation Github e al modello di lavoro che avvia l'automazione.

1. Compilare le variabili specifiche del proprio ambiente, quindi copiarle e incollarle nei campi Extra Vars del modello di lavoro.
2. Una volta aggiunti i var aggiuntivi al modello di lavoro, è possibile avviare l'automazione.
3. Il modello di lavoro viene eseguito in tre fasi specificando i tag per ontap_CONFIG, linux_CONFIG e oracle_CONFIG.

CLI tramite l'host di controllo Ansible

1. Per configurare l'host Linux in modo che sia possibile utilizzarlo come host di controllo Ansible [fare clic qui per istruzioni dettagliate](#)
2. Una volta configurato l'host di controllo Ansible, è possibile clonare il repository Ansible Automation.
3. Modificare il file hosts con gli IP e/o i nomi host della gestione del cluster ONTAP e degli IP di gestione del server Oracle.
4. Compilare le variabili specifiche dell'ambiente, quindi copiarle e incollarle in `vars.yml` file.
5. Ogni host Oracle dispone di un file variabile identificato dal relativo nome host che contiene variabili specifiche dell'host.
6. Una volta completati tutti i file variabili, è possibile eseguire il playbook in tre fasi specificando i tag per `ontap_config`, `linux_config`, e `oracle_config`.

Requisiti

Ambiente	Requisiti
Ambiente Ansible	Host AWX/Tower o Linux come host di controllo Ansible
	Ansible v.2.10 e versioni successive
	Python 3
	Librerie Python - netapp-lib - xmltodict - jmespath
ONTAP	ONTAP versione 9.3 - 9.7
	Due aggregati di dati
	VLAN NFS e ifgrp create

Ambiente	Requisiti
Server Oracle	RHEL 7/8
	Oracle Linux 7/8
	Interfacce di rete per NFS, gestione pubblica e opzionale
	File di installazione Oracle sui server Oracle

Dettagli sull'automazione

Questa implementazione automatica è progettata con un singolo playbook Ansible che consiste di tre ruoli separati. I ruoli sono per le configurazioni ONTAP, Linux e Oracle. La seguente tabella descrive le attività automatizzate.

Ruolo	Attività
ontap_config	Verifica preliminare dell'ambiente ONTAP
	Creazione di SVM basate su NFS per Oracle
	Creazione di policy di esportazione
	Creazione di volumi per Oracle
	Creazione di LIF NFS
linux_config	Creare punti di montaggio e montare volumi NFS
	Verificare i montaggi NFS
	Configurazione specifica del sistema operativo
	Creare directory Oracle
	Configurare gli hugepage
	Disattiva SELinux e il daemon del firewall
	Attivare e avviare il servizio chronyd
	aumentare il limite massimo del descrittore di file
	Creare il file di sessione pam.d.
oracle_config	Installazione del software Oracle
	Creare un listener Oracle
	Creare database Oracle
	Configurazione dell'ambiente Oracle
	Salva stato PDB
	Attivare la modalità di archiviazione delle istanze
	Abilitare il client DNFS
	Abilitare l'avvio e lo spegnimento automatici del database tra i riavvii del sistema operativo

Parametri predefiniti

Per semplificare l'automazione, abbiamo preimpostato molti parametri di implementazione Oracle richiesti con valori predefiniti. In genere non è necessario modificare i parametri predefiniti per la maggior parte delle implementazioni. Un utente più avanzato può apportare modifiche ai parametri predefiniti con cautela. I parametri predefiniti si trovano in ogni cartella di ruoli nella directory dei valori predefiniti.

Istruzioni per l'implementazione

Prima di iniziare, scaricare i seguenti file di installazione e patch Oracle e inserirli in `/tmp/archive` Directory con accesso in lettura, scrittura ed esecuzione per tutti gli utenti su ciascun server DB da implementare. Le attività di automazione cercano i file di installazione denominati in quella particolare directory per l'installazione e la configurazione di Oracle.

```
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
```

Licenza

Leggere le informazioni sulla licenza come indicato nel repository Github. Accedendo, scaricando, installando o utilizzando il contenuto di questo repository, l'utente accetta i termini della licenza stabilita ["qui"](#).

Si noti che esistono alcune limitazioni relative alla produzione e/o alla condivisione di qualsiasi opera derivata con il contenuto di questo repository. Leggere attentamente i termini del ["Licenza"](#) prima di utilizzare il contenuto. Se non si accettano tutti i termini, non accedere, scaricare o utilizzare il contenuto di questo repository.

Una volta pronti, fare clic su ["Qui per le procedure di implementazione AWX/Tower dettagliate"](#) oppure ["Qui per l'implementazione della CLI"](#).

Procedura di implementazione passo-passo

In questa pagina viene descritto il metodo automatizzato per l'implementazione di Oracle19c sullo storage NetApp ONTAP.

Implementazione AWX/Tower Database Oracle 19c

1. Creare l'inventario, il gruppo, gli host e le credenziali per il proprio ambiente

Questa sezione descrive la configurazione di inventario, gruppi, host e credenziali di accesso in AWX/Ansible Tower che preparano l'ambiente per l'utilizzo delle soluzioni automatizzate di NetApp.

1. Configurare l'inventario.

- Accedere a Resources → Inventories → Add e fare clic su Add Inventory (Aggiungi inventario).
- Fornire il nome e i dettagli dell'organizzazione, quindi fare clic su Save (Salva).
- Nella pagina Inventories (inventari), fare clic sull'inventario creato.
- Se sono presenti variabili di inventario, incollarle nel campo variabili.
- Accedere al sottomenu Groups (gruppi) e fare clic su Add (Aggiungi).

- f. Fornire il nome del gruppo per ONTAP, incollare le variabili di gruppo (se presenti) e fare clic su Salva.
 - g. Ripetere la procedura per un altro gruppo per Oracle.
 - h. Selezionare il gruppo ONTAP creato, accedere al sottomenu hosts e fare clic su Aggiungi nuovo host.
 - i. Fornire l'indirizzo IP dell'IP di gestione del cluster ONTAP, incollare le variabili host (se presenti) e fare clic su Salva.
 - j. Questo processo deve essere ripetuto per l'IP/nome host di gestione del gruppo Oracle e degli host Oracle.
2. Creare tipi di credenziale. Per le soluzioni che utilizzano ONTAP, è necessario configurare il tipo di credenziale in modo che corrisponda alle voci di nome utente e password.
 - a. Accedere a Administration → Credential Types (Amministrazione tipi di credenziali) e fare clic su Add (Aggiungi).
 - b. Fornire il nome e la descrizione.
 - c. Incollare il seguente contenuto in Input Configuration (Configurazione input):

```
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. Incollare il seguente contenuto nella configurazione dell'iniettore:

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

1. Configurare le credenziali.
 - a. Accedere a Resources → Credentials (risorse credenziali) e fare clic su Add (Aggiungi).
 - b. Immettere il nome e i dettagli dell'organizzazione per ONTAP.
 - c. Selezionare il tipo di credenziale personalizzato creato per ONTAP.
 - d. In Dettagli tipo, immettere il nome utente, la password e la password vsadmin_password.
 - e. Fare clic su Torna alla credenziale e fare clic su Aggiungi.
 - f. Immettere il nome e i dettagli dell'organizzazione per Oracle.

- g. Selezionare il tipo di credenziale Machine.
- h. In Dettagli tipo, immettere il nome utente e la password per gli host Oracle.
- i. Selezionare il metodo corretto di escalation dei privilegi e immettere il nome utente e la password.

2. Creare un progetto

1. Accedere a risorse → progetti e fare clic su Aggiungi.
 - a. Inserire il nome e i dettagli dell'organizzazione.
 - b. Selezionare Git nel campo Source Control Credential Type (tipo credenziale controllo origine).
 - c. invio <https://github.com/NetApp-Automation/na_oracle19c_deploy.git> Come URL del controllo di origine.
 - d. Fare clic su Salva.
 - e. Potrebbe essere necessario sincronizzare il progetto occasionalmente quando il codice sorgente cambia.

3. Configurare Oracle host_vars

Le variabili definite in questa sezione vengono applicate a ogni singolo server e database Oracle.

1. Inserire i parametri specifici dell'ambiente nel seguente modulo host Oracle incorporato o host_vars.



Gli elementi in blu devono essere modificati in base all'ambiente in uso.

Config. VAR host

```
#####
#####          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. Inserire tutte le variabili nei campi blu.
2. Una volta completata l'immissione delle variabili, fare clic sul pulsante Copy (Copia) del modulo per copiare tutte le variabili da trasferire su AWX o Tower.
3. Tornare a AWX o Tower e andare a Resources → hosts, quindi selezionare e aprire la pagina di configurazione del server Oracle.
4. Nella scheda Dettagli, fare clic su Modifica e incollare le variabili copiate dal punto 1 nel campo variabili sotto la scheda YAML.
5. Fare clic su Salva.
6. Ripetere questa procedura per tutti i server Oracle aggiuntivi nel sistema.

4. Configurare le variabili globali

Le variabili definite in questa sezione si applicano a tutti gli host Oracle, ai database e al cluster ONTAP.

1. Inserire i parametri specifici dell'ambiente nel seguente formato vars o variabili globali incorporate.



Gli elementi in blu devono essere modificati in base all'ambiente in uso.

```
#####
##### 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. Inserire tutte le variabili nei campi blu.
2. Una volta completata l'immissione delle variabili, fare clic sul pulsante Copy (Copia) del modulo per copiare tutte le variabili da trasferire a AWX o Tower nel seguente modello di lavoro.

5. Configurare e avviare il modello di lavoro.

1. Creare il modello di lavoro.
 - a. Accedere a risorse → modelli → Aggiungi e fare clic su Aggiungi modello di processo.
 - b. Immettere il nome e la descrizione
 - c. Selezionare il tipo di lavoro; Esegui consente di configurare il sistema in base a un playbook e Check esegue un'esecuzione a secco di un playbook senza configurare effettivamente il sistema.
 - d. Seleziona l'inventario, il progetto, il playbook e le credenziali corrispondenti per il playbook.
 - e. Selezionare all_playbook.yml come playbook predefinito da eseguire.
 - f. Incollare le variabili globali copiate dal passaggio 4 nel campo Template Variables (variabili modello) nella scheda YAML.
 - g. Selezionare la casella prompt all'avvio nel campo Job Tags.
 - h. Fare clic su Salva.
2. Avviare il modello di lavoro.
 - a. Accedere a risorse → modelli.

- b. Fare clic sul modello desiderato, quindi fare clic su Launch (Avvia).
- c. Quando richiesto all'avvio per Job Tags, digitare requirements_config. Potrebbe essere necessario fare clic sulla riga Create Job Tag sotto requirements_config per inserire il tag del processo.



requirements_config garantisce di disporre delle librerie corrette per eseguire gli altri ruoli.

1. Fare clic su Avanti, quindi su Avvia per avviare il processo.
2. Fare clic su View → Jobs (Visualizza lavori) per monitorare l'output e l'avanzamento del lavoro.
3. Quando richiesto all'avvio per Job Tags, digitare ontap_config. Potrebbe essere necessario fare clic sulla riga Create "Job Tag" (Crea tag lavoro) sotto ontap_config per inserire il tag del lavoro.
4. Fare clic su Avanti, quindi su Avvia per avviare il processo.
5. Fare clic su View → Jobs (Visualizza lavori) per monitorare l'output e l'avanzamento del lavoro
6. Una volta completato il ruolo ontap_CONFIG, eseguire nuovamente il processo per linux_CONFIG.
7. Accedere a risorse → modelli.
8. Selezionare il modello desiderato, quindi fare clic su Launch (Avvia).
9. Quando richiesto all'avvio per il tipo di tag del processo in linux_config, potrebbe essere necessario selezionare la riga Create "job tag" (Crea tag del processo) sotto linux_config per inserire il tag del processo.
10. Fare clic su Avanti, quindi su Avvia per avviare il processo.
11. Selezionare Visualizza → lavori per monitorare l'output e l'avanzamento del lavoro.
12. Una volta completato il ruolo linux_config, eseguire nuovamente il processo per oracle_config.
13. Accedere a risorse → modelli.
14. Selezionare il modello desiderato, quindi fare clic su Launch (Avvia).
15. Quando richiesto all'avvio per Job Tags, digitare oracle_config. Potrebbe essere necessario selezionare la riga Create "Job Tag" (Crea tag lavoro) sotto oracle_config per inserire il tag lavoro.
16. Fare clic su Avanti, quindi su Avvia per avviare il processo.
17. Selezionare Visualizza → lavori per monitorare l'output e l'avanzamento del lavoro.

6. Implementare database aggiuntivi sullo stesso host Oracle

La parte Oracle del playbook crea un singolo database container Oracle su un server Oracle per ogni esecuzione. Per creare ulteriori database container sullo stesso server, attenersi alla seguente procedura.

1. Rivedere le variabili host_vars.
 - a. Tornare al passaggio 2 - Configurazione di Oracle host_vars.
 - b. Modificare il SID Oracle con una stringa di denominazione diversa.
 - c. Modificare la porta del listener con un numero diverso.
 - d. Modificare la porta EM Express con un numero diverso se si installa EM Express.
 - e. Copiare e incollare le variabili host riviste nel campo Oracle host Variables (variabili host Oracle) nella scheda host Configuration Detail (Dettagli configurazione host).
2. Avviare il modello di processo di implementazione con solo il tag oracle_config.
3. Accedere al server Oracle come utente oracle ed eseguire i seguenti comandi:

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



In questo modo verranno elencati i processi oracle se l'installazione è stata completata come previsto e oracle DB è stato avviato

4. Accedere al database per controllare le impostazioni di configurazione del db e i PDB creati con i seguenti set di comandi.

```
[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 pdba
```

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

Ciò conferma che DNFS funziona correttamente.

5. Connettersi al database tramite listener per controllare la configurazione del listener Oracle con il seguente comando. Passare alla porta del listener e al nome del servizio database appropriati.

```
[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
```

Ciò conferma che Oracle listener funziona correttamente.

Dove cercare aiuto?

Se hai bisogno di aiuto con il toolkit, iscriviti a. ["La community di NetApp Solution Automation supporta il canale slack"](#) e cerca il canale di automazione della soluzione per inviare domande o domande.

Procedura di implementazione passo-passo

Il presente documento descrive in dettaglio l'implementazione di Oracle 19c utilizzando l'interfaccia a riga di comando (cli) di automazione.

Implementazione CLI Database Oracle 19c

Questa sezione descrive i passaggi necessari per preparare e implementare il database Oracle19c con la CLI. Assicurarsi di aver esaminato il ["Guida introduttiva e sezione sui requisiti"](#) e preparato il tuo ambiente di conseguenza.

Scarica Oracle19c repo

1. Dal controller ansible, esegui il seguente comando:

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

2. Dopo aver scaricato il repository, modificare le directory in na_oracle19c_deploy <cd na_oracle19c_deploy>.

Modificare il file hosts

Prima dell'implementazione, completare le seguenti operazioni:

1. Modificare la directory `na_oracle19c_deploy` del file `hosts`.
2. In `[ONTAP]` (indirizzo IP), modificare l'indirizzo IP in base all'IP di gestione del cluster.
3. Nel gruppo `[oracle]`, aggiungere i nomi degli host oracle. Il nome host deve essere risolto nel relativo indirizzo IP tramite DNS o il file `hosts`, oppure deve essere specificato nell'host.
4. Una volta completata questa procedura, salvare le modifiche.

Il seguente esempio illustra un file host:

```
#ONTAP Host

[ontap]

"10.61.184.183"

#Oracle hosts

[oracle]

"rtpora01"

"rtpora02"
```

Questo esempio esegue il playbook e implementa oracle 19c su due server oracle DB contemporaneamente. È inoltre possibile eseguire il test con un solo server DB. In tal caso, è necessario configurare un solo file di variabili host.



Il playbook viene eseguito allo stesso modo, indipendentemente dal numero di host e database Oracle implementati.

Modificare il file `host_name.yml` in `host_vars`

Ciascun host Oracle dispone di un file di variabili host identificato dal nome host che contiene variabili specifiche dell'host. È possibile specificare qualsiasi nome per l'host. Modificare e copiare `host_vars` Dalla sezione host VARS Config (Configurazione VAR host) e incollarla nel file desiderato `host_name.yml` file.



Gli elementi in blu devono essere modificati in base all'ambiente in uso.

Config. VAR host

```
#####
#####          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"}
```

Modificare il file vars.yml

Il vars.yml File consolida tutte le variabili specifiche dell'ambiente (ONTAP, Linux o Oracle) per l'implementazione Oracle.

1. Modificare e copiare le variabili dalla sezione VAR e incollarle nel vars.yml file.

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

Eseguire il manuale

Dopo aver completato i prerequisiti di ambiente richiesti e aver copiato le variabili in `vars.yml` e `your_host.yml`, ora sei pronto per implementare i playbook.



<username> deve essere modificato in base all'ambiente in uso.

1. Avvia la guida ONTAP inserendo i tag corretti e il nome utente del cluster ONTAP. Immettere la password per il cluster ONTAP e vsadmin quando richiesto.

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

2. Eseguire il playbook Linux per eseguire la parte di distribuzione di Linux. Immettere la password admin ssh e sudo.

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

3. Esegui il manuale Oracle per eseguire la parte relativa all'implementazione di Oracle. Immettere la password admin ssh e sudo.

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

Implementare database aggiuntivi sullo stesso host Oracle

La parte Oracle del playbook crea un singolo database container Oracle su un server Oracle per ogni esecuzione. Per creare un database container aggiuntivo sullo stesso server, attenersi alla seguente procedura:

1. Rivedere le variabili host_vars.
 - a. Tornare al passaggio 3 - modificare `host_name.yml` file sotto `host_vars`.
 - b. Modificare il SID Oracle con una stringa di denominazione diversa.
 - c. Modificare la porta del listener con un numero diverso.
 - d. Modificare la porta EM Express con un numero diverso se è stato installato EM Express.
 - e. Copiare e incollare le variabili host riviste nel file delle variabili host Oracle in `host_vars`.
2. Eseguire il manuale con `oracle_config` contrassegnare come illustrato nella [Eseguire il manuale](#).

Convalidare l'installazione di Oracle

1. Accedere al server Oracle come utente oracle ed eseguire i seguenti comandi:

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



In questo modo verranno elencati i processi oracle se l'installazione è stata completata come previsto e oracle DB è stato avviato

2. Accedere al database per controllare le impostazioni di configurazione del db e i PDB creati con i seguenti set di comandi.

```
[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

Ciò conferma che DNFS funziona correttamente.

3. Connettersi al database tramite listener per controllare la configurazione del listener Oracle con il seguente comando. Passare alla porta del listener e al nome del servizio database appropriati.

```
[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
```

Ciò conferma che Oracle listener funziona correttamente.

Dove cercare aiuto?

Se hai bisogno di aiuto con il toolkit, iscriviti a. ["La community di NetApp Solution Automation supporta il canale slack"](#) e cerca il canale di automazione della soluzione per inviare domande o domande.

Informazioni sul copyright

Copyright © 2024 NetApp, Inc. Tutti i diritti riservati. Stampato negli Stati Uniti d'America. Nessuna porzione di questo documento soggetta a copyright può essere riprodotta in qualsiasi formato o mezzo (grafico, elettronico o meccanico, inclusi fotocopie, registrazione, nastri o storage in un sistema elettronico) senza previo consenso scritto da parte del detentore del copyright.

Il software derivato dal materiale sottoposto a copyright di NetApp è soggetto alla seguente licenza e dichiarazione di non responsabilità:

IL PRESENTE SOFTWARE VIENE FORNITO DA NETAPP "COSÌ COM'È" E SENZA QUALSIVOGLIA TIPO DI GARANZIA IMPLICITA O ESPRESSA FRA CUI, A TITOLO ESEMPLIFICATIVO E NON ESAUSTIVO, GARANZIE IMPLICITE DI COMMERCIALIZZABILITÀ E IDONEITÀ PER UNO SCOPO SPECIFICO, CHE VENGONO DECLINATE DAL PRESENTE DOCUMENTO. NETAPP NON VERRÀ CONSIDERATA RESPONSABILE IN ALCUN CASO PER QUALSIVOGLIA DANNO DIRETTO, INDIRETTO, ACCIDENTALE, SPECIALE, ESEMPLARE E CONSEGUENZIALE (COMPRESI, A TITOLO ESEMPLIFICATIVO E NON ESAUSTIVO, PROCUREMENT O SOSTITUZIONE DI MERCI O SERVIZI, IMPOSSIBILITÀ DI UTILIZZO O PERDITA DI DATI O PROFITTI OPPURE INTERRUZIONE DELL'ATTIVITÀ AZIENDALE) CAUSATO IN QUALSIVOGLIA MODO O IN RELAZIONE A QUALUNQUE TEORIA DI RESPONSABILITÀ, SIA ESSA CONTRATTUALE, RIGOROSA O DOVUTA A INSOLVENZA (COMPRESA LA NEGLIGENZA O ALTRO) INSORTA IN QUALSIASI MODO ATTRAVERSO L'UTILIZZO DEL PRESENTE SOFTWARE ANCHE IN PRESENZA DI UN PREAVVISO CIRCA L'EVENTUALITÀ DI QUESTO TIPO DI DANNI.

NetApp si riserva il diritto di modificare in qualsiasi momento qualunque prodotto descritto nel presente documento senza fornire alcun preavviso. NetApp non si assume alcuna responsabilità circa l'utilizzo dei prodotti o materiali descritti nel presente documento, con l'eccezione di quanto concordato espressamente e per iscritto da NetApp. L'utilizzo o l'acquisto del presente prodotto non comporta il rilascio di una licenza nell'ambito di un qualche diritto di brevetto, marchio commerciale o altro diritto di proprietà intellettuale di NetApp.

Il prodotto descritto in questa guida può essere protetto da uno o più brevetti degli Stati Uniti, esteri o in attesa di approvazione.

LEGENDA PER I DIRITTI SOTTOPOSTI A LIMITAZIONE: l'utilizzo, la duplicazione o la divulgazione da parte degli enti governativi sono soggetti alle limitazioni indicate nel sottoparagrafo (b)(3) della clausola Rights in Technical Data and Computer Software del DFARS 252.227-7013 (FEB 2014) e FAR 52.227-19 (DIC 2007).

I dati contenuti nel presente documento riguardano un articolo commerciale (secondo la definizione data in FAR 2.101) e sono di proprietà di NetApp, Inc. Tutti i dati tecnici e il software NetApp forniti secondo i termini del presente Contratto sono articoli aventi natura commerciale, sviluppati con finanziamenti esclusivamente privati. Il governo statunitense ha una licenza irrevocabile limitata, non esclusiva, non trasferibile, non cedibile, mondiale, per l'utilizzo dei Dati esclusivamente in connessione con e a supporto di un contratto governativo statunitense in base al quale i Dati sono distribuiti. Con la sola esclusione di quanto indicato nel presente documento, i Dati non possono essere utilizzati, divulgati, riprodotti, modificati, visualizzati o mostrati senza la previa approvazione scritta di NetApp, Inc. I diritti di licenza del governo degli Stati Uniti per il Dipartimento della Difesa sono limitati ai diritti identificati nella clausola DFARS 252.227-7015(b) (FEB 2014).

Informazioni sul marchio commerciale

NETAPP, il logo NETAPP e i marchi elencati alla pagina <http://www.netapp.com/TM> sono marchi di NetApp, Inc. Gli altri nomi di aziende e prodotti potrebbero essere marchi dei rispettivi proprietari.