정영균 2019.12.04

Oracle Sharding on OCI

데모 Overview

네개의 OCI VM instance로 Sharded Database를 구성.

- inst-sdirector: Shard Director
- inst-scatalog: Shard Catalog
- inst-shard1: 첫번째 Shard
- inst-shard2: 두번째 Shard

Shard Director는 public subnet에 위치, 나머지 Shard Catalog, Shard1, Shard2는 동일 private subnet에 위치 시킴.

19c non-CDB, CREATE SHARD 방식으로 system-managed sharding 구성함. DG, ADG, OGG 설정은 하지 않음.

사전 작업

참조. Oracle Sharding deploy 매뉴얼

https://docs.oracle.com/en/database/oracle/oracle-database/19/shard/sharding-deployment.html#GUID-F99B8742-4089-4E77-87D4-4691EA932207

참조. Master Note for Handling Oracle Sharding - Oracle Database 12.2 Technology (문서 ID 2226341.1)

https://support.oracle.com/epmos/faces/SearchDocDisplay?_adf.ctrl-state=fk02hpwns_4&_afrLoop=354772811691875

Oracle Sharding deploy 순서

- Shard catalog host 에 DB 생성
- Shard node 에 Oracle DB software 설치
- Shard director node 에 Shard director(GSM) software 설치

Shard DB를 deploy 하는 방법은 두가지가 있음

- CREATE SHARD 명령으로 shard를 생성하면서 replication까지 함께 자동으로 생성. 단, PDB가 shard로 사용되는 Multitenant에서는 지원되지 않음
- ADD SHARD 명령으로 미리 생성된 DB를 shard에 추가.

Oracle Cloud Infrastructure에서 19c non-CDB 환경에서 Sharding을 구성할 것이므로 첫번째 CREATE SHARD 방식으로 deploy할 것임.

Oracle Sharding 구성을 위해 필요한 사전 네트웍 설정 – 아래 4,5 번째는 CREATE SHARD 방식에만 해당됨. 모두 설정

- shards > shard director : 1522 (shard director listener), 6123 (shard director local ONS), 6234 (shard director remote ONS)
- shards > shard catalog : 1521 (shard catalog listener)
- shard director, shard catalog > shards : 1521 (shard listener)
- shards > shard catalog : 8080 (agent_port)
- shard catalog > shards : 8080 (scheduler agent port)

■ Compartment

- demo

■ VCN

- demovcn 10.0.0.0/16

■ Internet Gateway

- internetgw

■ NAT Gateway

- natgw

■ Route Table

- routetab-public
  - ✓ Internet Gateway 0.0.0.0/0 internetgw
- routetab-private
  - ✓ NAT Gateway 0.0.0.0/0 natgw

■ Security List

- seclist-sdirector
  - ✓ Ingress Stateful 0.0.0.0/0 22 port
  - ✓ Ingress Stateful 10.0.2.0/24 1521 port
  - ✓ Ingress Stateful 10.0.2.0/24 1522 port
  - ✓ Ingress Stateful 10.0.2.0/24 6123 port
  - ✓ Ingress Stateful 10.0.2.0/24 6234 port
  - ✓ Ingress Stateful 0.0.0.0/0 1521 port
  - ✓ Ingress Stateful 0.0.0.0/0 1522 port

✓	Ingress Stateful	0.0.0.0/0	6123 port
✓	Ingress Stateful	0.0.0.0/0	6234 port
✓	Egress Stateful	0.0.0.0/0	All Protocols

- seclist-shard

✓	Ingress Stateful	10.0.1.0/24	22 port
✓	Ingress Stateful	10.0.1.0/24	1521 port
✓	Ingress Stateful	10.0.2.0/24	All Protocols
✓	Egress Stateful	0.0.0.0/0	All Protocols

■ Subnet

Regional subnet

-	subnet-sdirector	10.0.1.0/24	public	routetab-public	seclist-sdirector
-	subnet-shard	10.0.2.0/24	private	routetab-private	seclist-shard

■ Instance

Oracle Linux 7.7, Oracle Database 19c, VM.Standard2.1

인스턴스 생성 후, “yum update -y”로 패키지 업데이트 수행

-	inst-sdirector	subnet-sdirector	AD3	10.0.1.2	Shard director + bastion 역할.
-	inst-scatalog	subnet-shard	AD3	10.0.2.2	Shard catalog
-	inst-shard1	subnet-shard	AD1	10.0.2.3	Shard#1
-	inst-shard2	subnet-shard	AD2	10.0.2.4	Shard#2
-	inst-shard1stby	subnet-shard	AD2	10.0.2.5	Shard#1의 ADG standby
-	inst-shard2stby	subnet-shard	AD1	10.0.2.6	Shard#2의 ADG standby

호스트에서 방화벽 오픈 – shard director 호스트

```
[root@inst-sdirector opc]# firewall-cmd --permanent --add-port=1521/tcp
success
[root@inst-sdirector opc]# firewall-cmd --permanent --add-port=1522/tcp
success
[root@inst-sdirector opc]# firewall-cmd --permanent --add-port=6123/tcp
success
[root@inst-sdirector opc]# firewall-cmd --permanent --add-port=6234/tcp
success
[root@inst-sdirector opc]# firewall-cmd --reload
success
```

호스트에서 방화벽 오픈 – shard catalog 호스트

```
[root@inst-scatalog opc]# firewall-cmd --permanent --add-port=1521/tcp
success
[root@inst-scatalog opc]# firewall-cmd --permanent --add-port=8080/tcp
success
[root@inst-scatalog opc]# firewall-cmd --reload
success
```

호스트에서 방화벽 오픈 – shar1, shard2 호스트

```
[root@inst-scatalog opc]# firewall-cmd --permanent --add-port=1521/tcp
success
[root@inst-scatalog opc]# firewall-cmd --permanent --add-port=8080/tcp
success
[root@inst-scatalog opc]# firewall-cmd --reload
success
```

이후 작업 편의를 위해 미리 네개 호스트에 동일하게 아래 내용으로 host 파일 설정.

```
[root@inst-sdirector opc]# vi /etc/hosts
...
10.0.1.2 inst-sdirector.subnetsdirector.demovcn.oraclevcn.com inst-sdirector
10.0.2.2 inst-scatalog.subnetshard.demovcn.oraclevcn.com inst-scatalog
10.0.2.3 inst-shard1.subnetshard.demovcn.oraclevcn.com inst-shard1
10.0.2.4 inst-shard2.subnetshard.demovcn.oraclevcn.com inst-shard2
...
```

Oracle Database Software 설치

shard catalog, shard 가 위치할 인스턴스 “inst-scatalog”, “inst-shard1”, “inst-shard2”에 Oracle Database Software 설치

여기서는 rpm으로 Oracle Database Software를 설치. 아래 사이트에서 “Oracle Database 19c (19.3) for Linux x86-64 (RPM)” 다운로드

https://www.oracle.com/database/technologies/oracle19c-linux-downloads.html

각 세 서버 “inst-scatalog”, “inst-shard1”, “inst-shard2”에 아래 rpm을 업로드부터 Database Software 설치까지 수행

```
youjung@YOUJUNG-KR MINGW64 ~/Downloads
$ sftp -o ProxyCommand='ssh -i C:\\Users\\youjung\\.ssh\\id_rsa opc@132.145.146.238 -W %h:%p %r'
-i C:\\Users\\youjung\\.ssh\\id_rsa opc@10.0.2.2
Connected to opc@10.0.2.2.
s     put oracle-database-ee-19c-1.0-1.x86_64.rpm
Uploading oracle-database-ee-19c-1.0-1.x86_64.rpm to /home/opc/oracle-database-ee-19c-1.0-
1.x86_64.rpm
oracle-database-ee-19c-1.0-1.x86_64.rpm
100% 2570MB   2.8MB/s   15:09
```

ssh 접속, 파일 소유권 변경

```
[opc@inst-scatalog ~]$ mv oracle-database-ee-19c-1.0-1.x86_64.rpm /tmp
[opc@inst-scatalog ~]$ sudo -s
[root@inst-scatalog opc]# chown root:root /tmp/oracle-database-ee-19c-1.0-1.x86_64.rpm
```

Preinstallation RPM 설치

```
[root@inst-scatalog opc]# yum install oracle-database-preinstall-19c -y
Loaded plugins: langpacks, ulninfo
Resolving Dependencies
--> Running transaction check
---> Package oracle-database-preinstall-19c.x86_64 0:1.0-1.el7 will be installed
:
:
Complete!
```

Database software 설치

```
[root@inst-scatalog opc]# cd /tmp
[root@inst-scatalog tmp]# yum localinstall oracle-database-ee-19c-1.0-1.x86_64.rpm -y
Loaded plugins: langpacks, ulninfo
Examining oracle-database-ee-19c-1.0-1.x86_64.rpm: oracle-database-ee-19c-1.0-1.x86_64
Marking oracle-database-ee-19c-1.0-1.x86_64.rpm to be installed
Resolving Dependencies
--> Running transaction check
---> Package oracle-database-ee-19c.x86_64 0:1.0-1 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================================
 Package                 Arch      Version    Repository                              Size
================================================================================================
Installing:
 oracle-database-ee-19c    x86_64    1.0-1      /oracle-database-ee-19c-1.0-1.x86_64    6.9 G

Transaction Summary
================================================================================================
Install  1 Package

Total size: 6.9 G
Installed size: 6.9 G
Downloading packages:
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : oracle-database-ee-19c-1.0-1.x86_64                                 1/1
[INFO] Executing post installation scripts...
[INFO] Oracle home installed successfully and ready to be configured.
To configure a sample Oracle Database you can execute the following service configuration script
as root: /etc/init.d/oracledb_ORCLCDB-19c configure
  Verifying  : oracle-database-ee-19c-1.0-1.x86_64                                 1/1

Installed:
  oracle-database-ee-19c.x86_64 0:1.0-1

Complete!
```

Shard Director Software 설치

아래 GSM 설치 문서 참조

https://docs.oracle.com/en/database/oracle/oracle-database/19/gsmug/global-data-services-config.html#GUID-04D33448-2CB4-40C7-9DA0-1CFC6EC5E101

shard director가 위치할 인스턴스 “inst-sdirector”에 shard director software (global service manager-GSM software) 설치

아래 사이트에서 “Oracle Database 19c Global Service Manager (GSM/GDS) (19.3) for Linux x86-64” 다운로드

https://www.oracle.com/database/technologies/oracle19c-linux-downloads.html

“inst-sdirector”에 아래 설치파일을 업로드

```
youjung@YOUJUNG-KR MINGW64 ~/Downloads
```

```
$ sftp -i C:\\Users\\youjung\\.ssh\\id_rsa opc@132.145.146.238
Connected to opc@132.145.146.238.
s     put LINUX.X64_193000_gsm.zip
Uploading LINUX.X64_193000_gsm.zip to /home/opc/LINUX.X64_193000_gsm.zip
LINUX.X64_193000_gsm.zip
100%  915MB   4.9MB/s   03:05
```

GSM 설치 시 14GB 이상의 swap size가 필요함. 이미 14GB 이상의 swap이 확보되어 있다면, 아래 swap size 조정 작업은 하지 않아도 됨. 현재 8GB. 여기서는 8GB swap file을 추가하는 방식으로 swap size를 변경함.

```
[root@inst-sdirector ~]# swapon -s
Filename                                Type            Size    Used    Priority
/dev/sda2                               partition       8388604 0       -2
[root@inst-sdirector ~]# cat /proc/swaps
Filename                                Type            Size    Used    Priority
/dev/sda2                               partition       8388604 0       -2
[root@inst-sdirector ~]# cat /etc/fstab | grep swap
UUID=ffa35d76-0947-49fe-b030-3bf270640b7a swap                    swap    defaults,_netdev,x-
initrd.mount 0 0
```

swap file를 만들고, 이를 활성화하고, 서버 재부팅 시에도 반영되도록 /etc/fstab 파일에 아래 내용 추가

```
[root@inst-sdirector opc]# dd if=/dev/zero of=/root/swapfile count=1024 bs=8388608
1024+0 records in
1024+0 records out
8589934592 bytes (8.6 GB) copied, 117.564 s, 73.1 MB/s
[root@inst-sdirector opc]# mkswap -c /root/swapfile
Setting up swapspace version 1, size = 8388604 KiB
no label, UUID=b9434230-d884-449b-ba37-755ccac401e9
[root@inst-sdirector opc]# swapon /root/swapfile
swapon: /root/swapfile: insecure permissions 0644, 0600 suggested.
[root@inst-sdirector opc]# swapon -s
Filename                                Type            Size    Used    Priority
/dev/sda2                               partition       8388604 0       -2
/root/swapfile                          file    8388604 0       -3
[root@inst-sdirector opc]# vi /etc/fstab
...
/root/swapfile    swap    swap    defaults    0 0
...
[root@inst-sdirector opc]# cat /etc/fstab | grep swap
UUID=ffa35d76-0947-49fe-b030-3bf270640b7a swap                    swap    defaults,_netdev,x-
initrd.mount 0 0
/root/swapfile    swap    swap    defaults    0 0
```

Preinstallation RPM 설치 – 설치 유저가 root는 안되고, 미리 설치되어야 할 패키지가 많으므로 Oracle Database Preinstall 패키지를 설치함. GSM 설치 시 별도 OS 유저로 생성할 수 도 있으나, 여기서는 oracle 유저로 설치함.

```
[root@inst-sdirector opc]# yum install oracle-database-preinstall-19c -y
Loaded plugins: langpacks, ulninfo
Resolving Dependencies
--> Running transaction check
---> Package oracle-database-preinstall-19c.x86_64 0:1.0-1.el7 will be installed
:
:
Complete!
```

파일 소유권을 oracle로 변경 후 압축 해제

```
[root@inst-sdirector opc]# mv LINUX.X64_193000_gsm.zip /tmp
```

```
[root@inst-sdirector opc]# chown oracle:oinstall /tmp/LINUX.X64_193000_gsm.zip
[root@inst-sdirector opc]# su - oracle
[oracle@inst-sdirector ~]$ unzip /tmp/LINUX.X64_193000_gsm.zip
...
inflating: gsm/welcome.html
```

GSM 설치는 Oracle Universal Installer 할 수 있지만, 여기서는 Silent 모드로 수행함. 압축을 푼 GSM 설치 파일에서 response file "/gsm/response/gsm_install.rsp"을 열어서 설치 경로 등을 수정

```
[oracle@inst-sdirector ~]$ vi ~/gsm/response/gsm_install.rsp
...
#------------------------------------------------------------------------------
# Unix group to be set for the inventory directory.
#------------------------------------------------------------------------------
UNIX_GROUP_NAME=oinstall
#------------------------------------------------------------------------------
# Inventory location.
#------------------------------------------------------------------------------
INVENTORY_LOCATION=/u01/app/oraInventory
#------------------------------------------------------------------------------
# Complete path of the Oracle Home
#------------------------------------------------------------------------------
ORACLE_HOME=/u01/app/oracle/product/19c/dbhome_1
#------------------------------------------------------------------------------
# Complete path of the Oracle Base.
#------------------------------------------------------------------------------
ORACLE_BASE=/u01/app/oracle
```

response 파일에 나온 디렉토리 생성 및 소유권 변경, 권한 변경

```
[oracle@inst-sdirector ~]$ exit
logout
[root@inst-sdirector opc]# mkdir -p /u01/app/oraInventory
[root@inst-sdirector opc]# mkdir -p /u01/app/oracle/product/19c/dbhome_1
[root@inst-sdirector opc]# chown -R oracle:oinstall /u01
[root@inst-sdirector opc]# su - oracle
Last login: Thu Nov 21 02:24:04 GMT 2019 on pts/0
[oracle@inst-sdirector ~]$ chmod -R 775 /u01/app/oraInventory
[oracle@inst-sdirector ~]$ chmod -R 775 /u01/app/oracle
```

설치 수행. 매뉴얼과는 다르게 response file의 full path를 기술함.

```
[oracle@inst-sdirector ~]$ cd ~/gsm
[oracle@inst-sdirector gsm]$ ./runInstaller -silent -responseFile
/home/oracle/gsm/response/gsm_install.rsp -showProgress -ignorePrereq
Starting Oracle Universal Installer...

Checking Temp space: must be greater than 551 MB.   Actual 25843 MB    Passed
Preparing to launch Oracle Universal Installer from /tmp/OraInstall2019-12-02_01-21-27AM. Please
wait ...[oracle@inst-sdirector gsm]$ The response file for this session can be found at:
 /u01/app/oracle/product/19c/dbhome_1/install/response/gsm_2019-12-02_01-21-27AM.rsp


Prepare in progress.
.................................................   8% Done.

Prepare successful.

Copy files in progress.
.................................................   13% Done.
.................................................   19% Done.
```

```
.............................................   27% Done.
.............................................   33% Done.
.............................................   38% Done.
.............................................   43% Done.
.............................................   48% Done.
.............................................   53% Done.
.............................................   58% Done.
.............................................   64% Done.
.............................................   69% Done.
.............................................   74% Done.
.............................................   79% Done.

Copy files successful.

Link binaries in progress.

Link binaries successful.

Setup files in progress.
.....................................
Setup files successful.

Setup Inventory in progress.

Setup Inventory successful.
..........
Finish Setup in progress.
.............................................   84% Done.

Finish Setup successful.
The installation of Oracle Distributed Service and Load Management was successful.
Please check '/u01/app/oraInventory/logs/silentInstall2019-12-02_01-21-27AM.log' for more
details.

Setup Oracle Base in progress.

Setup Oracle Base successful.
.............................................   95% Done.

As a root user, execute the following script(s):
       1. /u01/app/oraInventory/orainstRoot.sh
       2. /u01/app/oracle/product/19c/dbhome_1/root.sh



Successfully Setup Software.
.............................................   100% Done.
The log of this install session can be found at:
 /u01/app/oraInventory/logs/installActions2019-12-02_01-21-27AM.log
```

이제 root 유저로 “orainstRoot.sh”과 “root.sh”을 수행한다.

```
[oracle@inst-sdirector gsm]$ exit
logout
[root@inst-sdirector opc]# sh /u01/app/oraInventory/orainstRoot.sh
Changing permissions of /u01/app/oraInventory.
Adding read,write permissions for group.
Removing read,write,execute permissions for world.

Changing groupname of /u01/app/oraInventory to oinstall.
The execution of the script is complete.
[root@inst-sdirector opc]# sh /u01/app/oracle/product/19c/dbhome_1/root.sh
Check /u01/app/oracle/product/19c/dbhome_1/install/root_inst-sdirector_2019-11-21_03-37-56-
440329838.log for the output of root script
```

GSM 설치 유저인 oracle 유저 profile에 GSM 관련 경로 지정

```
[root@inst-sdirector opc]# su - oracle
Last login: Thu Nov 21 03:26:52 GMT 2019 on pts/1
[oracle@inst-sdirector ~]$ vi .bash_profile
...
export TZ=Asia/Seoul
export ORACLE_BASE='/u01/app/oracle'
export ORACLE_HOME='/u01/app/oracle/product/19c/dbhome_1'
export LD_LIBRARY_PATH=$ORACLE_HOME/lib
export PATH=$ORACLE_HOME/bin:$PATH
...
[oracle@inst-sdirector ~]$ source .bash_profile
```

gdsctl 명령이 정상 수행되는 지 확인

```
[oracle@inst-sdirector ~]$ gdsctl
GDSCTL: Version 19.0.0.0.0 - Production on Thu Nov 21 12:44:17 KST 2019

Copyright (c) 2011, 2019, Oracle.  All rights reserved.

Welcome to GDSCTL, type "help" for information.

Warning:  GSM  is not set automatically because gsm.ora does not contain GSM entries. Use "set
gsm" command to set GSM for the session.
Current GSM is set to GSMORA
GDSCTL>
```

Shard Catalog Database 설치

Catalog 설치 호스트 "inst-scatalog"에서 설치.

oracle 유저 환경 변수 설정.

```
[root@inst-scatalog tmp]# su - oracle
Last login: Mon Dec  2 00:51:24 GMT 2019
[oracle@inst-scatalog ~]$ vi .bash_profile
...
export ORACLE_BASE=/opt/oracle
export ORACLE_HOME=/opt/oracle/product/19c/dbhome_1
export ORACLE_SID=SCATALOG
export PATH=$PATH:$ORACLE_HOME/bin
export LD_LIBRARY_PATH=$ORACLE_HOME/lib:/lib:/usr/lib:/usr/local/lib
...
[oracle@inst-scatalog ~]$ source .bash_profile
[oracle@inst-scatalog ~]$ echo $ORACLE_SID
SCATALOG
[oracle@inst-scatalog ~]$ which dbca
/opt/oracle/product/19c/dbhome_1/bin/dbca
```

데이터베이스 디렉토리 생성

```
[oracle@inst-scatalog ~]$ mkdir -p /opt/oracle/oradata/SCATALOG
```

dbca silent 모드로 non-CDB 데이터베이스 생성

```
[oracle@inst-scatalog ~]$ dbca -silent -createDatabase -templateName General_Purpose.dbc -
gdbName SCATALOG -sid SCATALOG -sysPassword Welcome123## -systemPassword Welcome123## -
emConfiguration NONE -datafileDestination /opt/oracle/oradata -storageType FS -characterSet
AL32UTF8
Prepare for db operation
```

```
10% complete
Copying database files
40% complete
Creating and starting Oracle instance
42% complete
46% complete
50% complete
54% complete
60% complete
Completing Database Creation
66% complete
69% complete
70% complete
Executing Post Configuration Actions
100% complete
Database creation complete. For details check the logfiles at:
 /opt/oracle/cfgtoollogs/dbca/SCATALOG.
Database Information:
Global Database Name:SCATALOG
System Identifier(SID):SCATALOG
Look at the log file "/opt/oracle/cfgtoollogs/dbca/SCATALOG/SCATALOG.log" for further details.
```

리스너 파일 생성, 리스너 시작 및 상태 확인

```
[oracle@inst-scatalog ~]$ vi $ORACLE_HOME/network/admin/listener.ora
...
LISTENER =
  (DESCRIPTION_LIST =
    (DESCRIPTION =
      (ADDRESS = (PROTOCOL = TCP)(HOST = inst-scatalog.subnetshard.demovcn.oraclevcn.com)(PORT =
1521))
      (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC1521))
    )
  )

SID_LIST_LISTENER =
  (SID_LIST =
    (SID_DESC =
      (GLOBAL_DBNAME = SCATALOG)
      (ORACLE_HOME = /opt/oracle/product/19c/dbhome_1)
      (SID_NAME = SCATALOG)
    )
  )
...
[oracle@inst-scatalog ~]$ lsnrctl start
...
The command completed successfully
[oracle@inst-scatalog ~]$ lsnrctl status

LSNRCTL for Linux: Version 19.0.0.0.0 - Production on 22-NOV-2019 03:36:29

Copyright (c) 1991, 2019, Oracle.  All rights reserved.

Connecting to (ADDRESS=(PROTOCOL=tcp)(HOST=)(PORT=1521))
STATUS of the LISTENER
------------------------
Alias                     LISTENER
Version                   TNSLSNR for Linux: Version 19.0.0.0.0 - Production
Start Date                22-NOV-2019 03:36:14
Uptime                    0 days 0 hr. 0 min. 15 sec
Trace Level               off
Security                  ON: Local OS Authentication
SNMP                      OFF
Listener Log File         /opt/oracle/diag/tnslsnr/inst-scatalog/listener/alert/log.xml
Listening Endpoints Summary...
```

```
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=inst-
scatalog.subnetshard.demovcn.oraclevcn.com)(PORT=1521)))
Services Summary...
Service "SCATALOG" has 1 instance(s).
  Instance "SCATALOG", status READY, has 1 handler(s) for this service...
Service "SCATALOGXDB" has 1 instance(s).
  Instance "SCATALOG", status READY, has 1 handler(s) for this service...
The command completed successfully
```

TNS 접속 정보 설정 및 연결 확인

```
[oracle@inst-scatalog ~]$ vi $ORACLE_HOME/network/admin/tnsnames.ora
...
SCATALOG =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = inst-scatalog.subnetshard.demovcn.oraclevcn.com)(PORT =
1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SID = SCATALOG)
    )
  )
...
[oracle@inst-scatalog ~]$ tnsping SCATALOG

TNS Ping Utility for Linux: Version 19.0.0.0.0 - Production on 22-NOV-2019 03:44:10

Copyright (c) 1997, 2019, Oracle.  All rights reserved.

Used parameter files:


Used TNSNAMES adapter to resolve the alias
Attempting to contact (DESCRIPTION = (ADDRESS = (PROTOCOL = TCP)(HOST = inst-
scatalog.subnetshard.demovcn.oraclevcn.com)(PORT = 1521)) (CONNECT_DATA = (SERVER = DEDICATED)
(SID = SCATALOG)))
OK (0 msec)
```

데이터베이스에 접속 테스트

```
[oracle@inst-scatalog ~]$ sqlplus sys@SCATALOG as sysdba

SQL*Plus: Release 19.0.0.0.0 - Production on Fri Nov 22 03:45:31 2019
Version 19.3.0.0.0

Copyright (c) 1982, 2019, Oracle.  All rights reserved.

Enter password:

Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.3.0.0.0

SQL> select instance_name,status from v$instance;

INSTANCE_NAME    STATUS
---------------- ------------
SCATALOG         OPEN
```

Sharding Management & Routing Tier 설정

shard catalog, shard director, 각 shard가 서로 통신하도록 설정

Shard Catalog 호스트에 접속해서 db_create_file_dest, open_links, open_links_per_instance 파라미터 설정. open_links, open_links_per_instance은 이후 데모용 어플리케이션에서 사용됨. 16으로 설정

```
[oracle@inst-scatalog ~]$ sqlplus "/as sysdba"
...
SQL> alter system set db_create_file_dest='/opt/oracle/oradata' scope=both;

System altered.

SQL> alter system set open_links=16 scope=spfile;

System altered.

SQL> alter system set open_links_per_instance=16 scope=spfile;

System altered.

SQL> shutdown immediate
...
SQL> startup
ORACLE instance started.
...
SQL> show parameter open_links

NAME                                 TYPE        VALUE
------------------------------------ ----------- ------------------------------
open_links                           integer     16
open_links_per_instance              integer     16
```

Sharding 유저/권한 설정. 신규로 생성하는 mysdbadmin 스키마에는 Sharding 환경 정보 및 관리 작업에 대한 변경 사항이 저장된다. gdsctl 명령이 mysdbadmin 유저를 통해 수행되어 Shading 변경 사항이 전달된다.

```
SQL> set echo on
SQL> set termout on
SQL> spool setup_grants_privs.lst
SQL> alter user gsmcatuser account unlock;

User altered.

SQL> alter user gsmcatuser identified by Welcome123##;

User altered.

SQL> create user mysdbadmin identified by Welcome123##;

User created.

SQL> grant connect, create session, gsmadmin_role to mysdbadmin;

Grant succeeded.

SQL> grant inherit privileges on user SYS to GSMADMIN_INTERNAL;

Grant succeeded.

SQL> spool off
```

Shard Director 호스트에 접속해서 shard catalog 생성. sharding 방식 (system-managed, user-defined, composite)에 따라 shard catalog 생성 옵션이 다름. 매뉴얼 참조. 여기서는 DG, ADG, OGG 등을 이용한

shard replication을 하지 않고, system-managed 방식 Sharding의 shard catalog를 생성

참고. Sharding에서 사용되는 GDSCTL 명령어 reference

https://docs.oracle.com/en/database/oracle/oracle-database/12.2/gsmug/gdsctl-sharding-env.html#GUID-3285C437-78F8-4339-AAAB-33DBB7A9D400

아래에서 "-agent_port"는 XDB가 사용할 포트 번호. 지정하지 않으면 8080으로 세팅

"-agent_password"는 remote scheduler가 Catalog DB에 agent 등록 시 사용하는 패스워드

```
[oracle@inst-sdirector ~]$ gdsctl
GDSCTL: Version 19.0.0.0.0 - Production on Fri Nov 22 13:17:23 KST 2019

Copyright (c) 2011, 2019, Oracle.  All rights reserved.

Welcome to GDSCTL, type "help" for information.

Warning:  GSM  is not set automatically because gsm.ora does not contain GSM entries. Use "set gsm" command to set GSM for the session.
Current GSM is set to GSMORA
GDSCTL> create shardcatalog -database 10.0.2.2:1521:SCATALOG -user mysdbadmin/Welcome123## -sdb demo_sdb -agent_port 8080 -agent_password Welcome123##

Catalog is created
```

shard director 생성 및 시작

```
GDSCTL> add gsm -gsm sdirector -listener 1521 -pwd Welcome123## -catalog 10.0.2.2:1521:SCATALOG
GSM successfully added
GDSCTL> start gsm -gsm sdirector
GSM is started successfully
```

shard director가 각 shard에 접속해서 remote scheduler agent를 수행하여 DBCA, NETCA 등을 이용해서 shard를 셋업하기 위해서는 ①각 shard에서 Oracle 환경설정과, ②Oracle Software 설치 유저 (oracle)의 OS 계정 정보가 필요함. 또한 ③각 shard에서 scheduler agent (schagent) 통신하는 포트를 찾아서 shard catalog가 통신할 수 있도록 개방해 줘야 한다.

Shard1 호스트에 접속해서 1번 shard에서 Oracle 환경설정

```
[root@inst-shard1 tmp]# su - oracle
Last login: Mon Dec  2 00:51:24 GMT 2019
[oracle@inst-shard1 ~]$ vi .bash_profile
...
export ORACLE_BASE=/opt/oracle
export ORACLE_HOME=/opt/oracle/product/19c/dbhome_1
export ORACLE_SID=sh1
export PATH=$PATH:$ORACLE_HOME/bin
export LD_LIBRARY_PATH=$ORACLE_HOME/lib:/lib:/usr/lib:/usr/local/lib
...
[oracle@inst-shard1 ~]$ source .bash_profile
[oracle@inst-shard1 ~]$ exit
logout
```

Shard2 호스트에 접속해서 2번 shard에서 Oracle 환경설정

```
[root@inst-shard2 tmp]# su - oracle
Last login: Mon Dec  2 00:51:22 GMT 2019
[oracle@inst-shard2 ~]$ vi .bash_profile
...
export ORACLE_BASE=/opt/oracle
export ORACLE_HOME=/opt/oracle/product/19c/dbhome_1
export ORACLE_SID=sh2
export PATH=$PATH:$ORACLE_HOME/bin
export LD_LIBRARY_PATH=$ORACLE_HOME/lib:/lib:/usr/lib:/usr/local/lib
...
[oracle@inst-shard2: ~]$ source .bash_profile
[oracle@inst-shard2 ~]$ exit
logout
```

shard1, shard2 호스트에서 패스워드 인증이 가능하도록 sshd 설정을 변경 후, root, oracle 패스워드 설정. shard1, shard2 호스트 모두 설정

```
[root@inst-shard1 opc]# vi /etc/ssh/sshd_config
...
PermitRootLogin yes
...
PasswordAuthentication yes
...
#PasswordAuthentication no
...
[root@inst-shard1 opc]# service sshd reload
Redirecting to /bin/systemctl reload sshd.service
[root@inst-shard1 opc]# passwd root
Changing password for user root.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[root@inst-shard1 opc]# passwd oracle
Changing password for user root.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
```

Shard Director 호스트에 접속해서 shard director가 각 shard에 접속해서 remote scheduler agent를 수행하여 DBCA, NETCA 등을 이용해서 shard를 셋업 때 사용할 credential 생성. 여기서 각 계정은 각 shard의 Oracle Software 설치 OS 유저 (oracle)의 정보임.

```
GDSCTL> add credential -credential oracle_cred -osaccount oracle -ospassword dhfkzmfWkd1!
The operation completed successfully
GDSCTL> exit
```

각 shard에 접속해서 schagent를 시작하고, shard catalog와 연결.

“schagent -registerdatabase” 명령에서 사용하는 포트는 “create shardcatalog” 에서 지정한 포트 번호임. 디폴트는 8080. 패스워드를 입력하라고 하는 부분에서는 마찬가지로 “create shardcatalog” 에서 지정한 패스워드를 입력

또한 schagent 사용 포트를 개방 한다. VCN 레벨에서는 이미 shard와 shard catalog 통신을 가능하도록 했으니, 호스트에서만 개방하면 됨

Shard1 호스트에 접속해서

```
[root@inst-shard1 opc]# su - oracle
Last login: Tue Dec  3 00:56:06 GMT 2019 on pts/0
[oracle@inst-shard1 ~]$ schagent -start

Warning:
The JKS keystore uses a proprietary format. It is recommended to migrate to PKCS12 which is an
industry standard format using "keytool -importkeystore -srckeystore
/opt/oracle/product/19c/dbhome_1/data/wallet/agent.key -destkeystore
/opt/oracle/product/19c/dbhome_1/data/wallet/agent.key -deststoretype pkcs12".

Scheduler agent started using port 17553
[oracle@inst-shard1 ~]$ schagent -status
Agent running with PID 3169

Agent_version:19.3.0.0.0
Running_time:00:00:29
Total_jobs_run:0
Running_jobs:0
Platform:Linux
ORACLE_HOME:/opt/oracle/product/19c/dbhome_1
ORACLE_BASE:/opt/oracle
Port:17553
Host:inst-shard1.subnetshard.demovcn.oraclevcn.com
[oracle@inst-shard1 ~]$ schagent -registerdatabase 10.0.2.2 8080
Agent Registration Password ? *************

Oracle Scheduler Agent Registration for 19.3.0.0.0 Agent
Agent Registration Successful!
[oracle@inst-shard1 ~]$ exit
logout
```

"schagent -status" 명령에서 확인한 schagent 사용 포트를 개방.

```
[root@inst-shard1 opc]# firewall-cmd --permanent --add-port=17553/tcp
success
[root@inst-shard1 opc]# firewall-cmd --reload
success
```

shard 생성 때 사용되는 디렉토리도 미리 생성한다.

```
[root@inst-shard1 opc]# su – oracle
[oracle@inst-shard1 ~]$ mkdir -p $ORACLE_BASE/oradata
[oracle@inst-shard1 ~]$ mkdir -p  $ORACLE_BASE/fast_recovery_area
```

Shard2 호스트에 접속해서

```
[root@inst-shard2 opc]# su - oracle
Last login: Tue Dec  3 00:51:21 GMT 2019 on pts/0
[oracle@inst-shard2 ~]$ schagent -start
[oracle@inst-shard2 ~]$ schagent -start

Warning:
The JKS keystore uses a proprietary format. It is recommended to migrate to PKCS12 which is an
industry standard format using "keytool -importkeystore -srckeystore
/opt/oracle/product/19c/dbhome_1/data/wallet/agent.key -destkeystore
/opt/oracle/product/19c/dbhome_1/data/wallet/agent.key -deststoretype pkcs12".

Scheduler agent started using port 20395
[oracle@inst-shard2 ~]$ schagent -status
Agent running with PID 25053

Agent_version:19.3.0.0.0
```

```
Running_time:00:00:26
Total_jobs_run:0
Running_jobs:0
Platform:Linux
ORACLE_HOME:/opt/oracle/product/19c/dbhome_1
ORACLE_BASE:/opt/oracle
Port:20395
Host:inst-shard2.subnetshard.demovcn.oraclevcn.com

[oracle@inst-shard2 ~]$ schagent -registerdatabase 10.0.2.2 8080
Agent Registration Password ? *************

Oracle Scheduler Agent Registration for 19.3.0.0.0 Agent
Agent Registration Successful!
[oracle@inst-shard2 ~]$ exit
logout
```

"schagent -status" 명령에서 확인한 schagent 사용 포트를 개방.

```
[root@inst-shard2 opc]# firewall-cmd --permanent --add-port=20395/tcp
success
[root@inst-shard2 opc]# firewall-cmd --reload
success
```

shard 생성 때 사용되는 디렉토리도 미리 생성한다.

```
[root@inst-shard2 opc]# su - oracle
[oracle@inst-shard2 ~]$ mkdir -p $ORACLE_BASE/oradata
[oracle@inst-shard2 ~]$ mkdir -p  $ORACLE_BASE/fast_recovery_area
```

System-Managed Sharded Database 생성

이번 테스트에서는 system-managed sharded database를 생성함. 이를 위해서 아래와 같은 작업이 필요함

- shardgroup과 shard 생성
- 각 데이터베이스가 shard로 사용되도록 설정
- DELPOY 명령 수행
- role-based global service를 생성

Shard Director 호스트에 접속해서 현재 세션에 대한 global service manager 설정

```
GDSCTL> set gsm -gsm sdirector
GDSCTL> connect mysdbadmin/Welcome123##
Catalog connection is established
```

"primary_shardgroup" 이라는 이름으로 shardgroup 생성. 이번 테스트에서는 Data Guard, Oracle Golden Gate replication 설정은 하지 않음.

shardgroup을 생성할 때 "-deploy_as primary" 옵션을 줘야 함. 그러지 않을 경우 shard 생성 시점에 "CATALOG:ORA-03783: no new shard to deploy" 발생. 이를 기존 디폴트로 생성되어 있는 primary group을 삭제함.

```
GDSCTL> add shardgroup -shardgroup primary_shardgroup
```

```
The operation completed successfully
```

각 shard 호스트를 catalog의 VNCR(valid node checking for registration) 리스트에 등록 후, 앞서 만든 shard group에 shard를 생성. VNCR은 shard director에서 등록이 허용된 IP주소, 호스트명, 서브넷 등을 설정하고 동적으로 업데이트 하는 역할을 한다.

“add invitenode”, “add shard” 할 때 사용되는 shard 정보는 shard 호스트명이다. shard 호스트 1, 2에 대해 shard를 생성한다.

```
GDSCTL> add invitednode inst-shard1
GDSCTL> create shard -shardgroup primary_shardgroup -destination inst-shard1 -credential
oracle_cred
The operation completed successfully
DB Unique Name: sh1
GDSCTL> add invitednode inst-shard2
GDSCTL> create shard -shardgroup primary_shardgroup -destination inst-shard2 -credential
oracle_cred
The operation completed successfully
DB Unique Name: sh2
```

설정을 확인. shard 이름(sh1, sh2)는 시스템이 생성한 이름임.

```
GDSCTL> config

Regions
------------------------
regionora

GSMs
------------------------
sdirector

Sharded Database
------------------------
demo_sdb

Databases
------------------------
sh1
sh2

Shard Groups
------------------------
primary_shardgroup
shardspaceora_regionora

Shard spaces
------------------------
shardspaceora

Services
------------------------

GDSCTL pending requests
------------------------
Command                    Object                    Status
-------                    ------                    ------
```

```
Global properties
------------------------
Name: oradbcloud
Master GSM: sdirector
DDL sequence #: 0


GDSCTL> config vncr
Name                          Group ID
----                          --------
10.0.2.2
inst-shard1
inst-shard2

GDSCTL> config shardspace
Shard space                   Chunks
-----------                   ------
shardspaceora

GDSCTL> config shardgroup
Shard Group         Chunks Region             Shard space
-----------         ------ ------             -----------
primary_shardgroup         regionora          shardspaceora
shardspaceora_regio        regionora          shardspaceora
nora

GDSCTL> config shard
Name               Shard Group         Status    State       Region    Availability
----               -----------         ------    -----       ------    ------------
sh1                primary_shardgroup  U         none        regionora -
sh2                primary_shardgroup  U         none        regionora -
```

설정한 내용대로 deploy를 한다.

```
GDSCTL> deploy
deploy: examining configuration...
deploy: deploying primary shard 'sh1' ...
deploy: network listener configuration successful at destination 'inst_shard1'
deploy: starting DBCA at destination 'inst_shard1' to create primary shard 'sh1' ...
deploy: deploying primary shard 'sh2' ...
deploy: network listener configuration successful at destination 'inst_shard2'
deploy: starting DBCA at destination 'inst_shard2' to create primary shard 'sh2' ...
deploy: waiting for 2 DBCA primary creation job(s) to complete...
deploy: waiting for 2 DBCA primary creation job(s) to complete...
deploy: waiting for 2 DBCA primary creation job(s) to complete...
deploy: waiting for 2 DBCA primary creation job(s) to complete...
deploy: waiting for 2 DBCA primary creation job(s) to complete...
deploy: waiting for 2 DBCA primary creation job(s) to complete...
deploy: waiting for 2 DBCA primary creation job(s) to complete...
deploy: waiting for 2 DBCA primary creation job(s) to complete...
deploy: waiting for 2 DBCA primary creation job(s) to complete...
deploy: waiting for 2 DBCA primary creation job(s) to complete...
deploy: waiting for 2 DBCA primary creation job(s) to complete...
deploy: waiting for 2 DBCA primary creation job(s) to complete...
deploy: waiting for 2 DBCA primary creation job(s) to complete...
deploy: waiting for 2 DBCA primary creation job(s) to complete...
```

```
deploy: waiting for 2 DBCA primary creation job(s) to complete...
deploy: DBCA primary creation job succeeded at destination 'inst_shard1' for shard 'sh1'
deploy: waiting for 1 DBCA primary creation job(s) to complete...
deploy: DBCA primary creation job succeeded at destination 'inst_shard2' for shard 'sh2'
deploy: requesting Data Guard configuration on shards via GSM
deploy: shards configured successfully
The operation completed successfully
```

deploy 결과를 확인. shard가 모두 등록되었는 지 확인. shard 상태 확인.

```
GDSCTL> config shard
Name                Shard Group         Status    State       Region    Availability
----                -----------         ------    -----       ------    ------------
sh3                 primary_shardgroup  Ok        Deployed    regionora ONLINE
sh4                 primary_shardgroup  Ok        Deployed    regionora ONLINE
GDSCTL> databases
Database: "sh1" Registered: Y State: Ok ONS: N. Role: PRIMARY Instances: 1 Region: regionora
   Registered instances:
     demo_sdb%1
Database: "sh1" Registered: Y State: Ok ONS: N. Role: PRIMARY Instances: 1 Region: regionora
   Registered instances:
     demo_sdb%11
GDSCTL> config shard -shard sh1
Name: sh1
Shard Group: primary_shardgroup
Status: Ok
State: Deployed
Region: regionora
Connection string: inst-
shard1.subnetshard.demovcn.oraclevcn.com:1521/sh1.subnetshard.demovcn.oraclevcn.com:dedicated
SCAN address:
ONS remote port: 0
Disk Threshold, ms: 20
CPU Threshold, %: 75
Version: 19.0.0.0
Failed DDL:
DDL Error: ---
Failed DDL id:
Availability: ONLINE
Rack:


Supported services
------------------------
Name                                                    Preferred Status
----                                                    --------- ------

GDSCTL> config shard -shard sh2
Name: sh2
Shard Group: primary_shardgroup
Status: Ok
State: Deployed
Region: regionora
Connection string: inst-
shard2.subnetshard.demovcn.oraclevcn.com:1521/sh2.subnetshard.demovcn.oraclevcn.com:dedicated
SCAN address:
ONS remote port: 0
```

```
Disk Threshold, ms: 20
CPU Threshold, %: 75
Version: 19.0.0.0
Failed DDL:
DDL Error: ---
Failed DDL id:
Availability: ONLINE
Rack:


Supported services
------------------------
Name                                                    Preferred Status
----                                                    --------- ------
```

primary shard에서 수행될 global service 생성

```
GDSCTL> add service -service oltp_rw_srvc -role primary
The operation completed successfully
GDSCTL> config service


Name          Network name                Pool          Started Preferred all
----          ------------                ----          ------- -------------
oltp_rw_srvc   oltp_rw_srvc.demo_sdb.oradbcl demo_sdb       No      Yes
              oud
```

global service 시작

```
GDSCTL> start service -service oltp_rw_srvc
The operation completed successfully
GDSCTL> status service
Service "oltp_rw_srvc.demo_sdb.oradbcloud" has 2 instance(s). Affinity: ANYWHERE
   Instance "demo_sdb%1", name: "sh1", db: "sh1", region: "regionora", status: ready.
   Instance "demo_sdb%11", name: "sh2", db: "sh2", region: "regionora", status: ready.
```

System-Managed Sharded Database를 위한 스키마 생성

Shard Catalog DB에 접속해서 수행. 스키마 유저, 테이블스페이스 (sharded 테이블용, duplicate 테이블용) 생성

```
[oracle@inst-scatalog ~]$ sqlplus "/as sysdba"

SQL*Plus: Release 19.0.0.0.0 - Production on Wed Dec 4 08:48:21 2019
Version 19.3.0.0.0
...
SQL> alter session enable shard ddl;

Session altered.

SQL> create user app_schema identified by Welcome123##;

User created.

SQL> grant all privileges to app_schema;

Grant succeeded.
```

```
SQL> grant gsmadmin_role to app_schema;

Grant succeeded.

SQL> grant select_catalog_role to app_schema;

Grant succeeded.

SQL> grant connect, resource to app_schema;

Grant succeeded.

SQL> grant dba to app_schema;

Grant succeeded.

SQL> grant execute on dbms_crypto to app_schema;

Grant succeeded.

SQL> CREATE TABLESPACE SET TSP_SET_1 using template (datafile size 100m autoextend on next 10M
maxsize unlimited extent management local segment space management auto);

Tablespace created.

SQL> CREATE TABLESPACE products_tsp datafile size 100m autoextend on next 10M maxsize unlimited
extent management local uniform size 1m;

Tablespace created.
```

테스트 유저(app_schema)에 sharded 테이블, duplicated 테이블 및 시퀀스 생성,

```
SQL> connect app_schema/Welcome123##
Connected.
SQL> ALTER SESSION ENABLE SHARD DDL;

Session altered.

SQL> CREATE SHARDED TABLE Customers
  (
    CustId      VARCHAR2(60) NOT NULL,
    FirstName   VARCHAR2(60),
    LastName    VARCHAR2(60),
    Class       VARCHAR2(10),
    Geo         VARCHAR2(8),
    CustProfile VARCHAR2(4000),
    Passwd      RAW(60),
    CONSTRAINT pk_customers PRIMARY KEY (CustId),
    CONSTRAINT json_customers CHECK (CustProfile IS JSON)
  ) TABLESPACE SET TSP_SET_1
  PARTITION BY CONSISTENT HASH (CustId) PARTITIONS AUTO;

Table created.

SQL> CREATE SHARDED TABLE Orders
  (
```

```
    OrderId     INTEGER NOT NULL,
    CustId      VARCHAR2(60) NOT NULL,
    OrderDate   TIMESTAMP NOT NULL,
    SumTotal    NUMBER(19,4),
    Status      CHAR(4),
    CONSTRAINT  pk_orders PRIMARY KEY (CustId, OrderId),
    CONSTRAINT  fk_orders_parent FOREIGN KEY (CustId)
    REFERENCES Customers ON DELETE CASCADE
  ) PARTITION BY REFERENCE (fk_orders_parent);

Table created.

SQL> CREATE SHARDED TABLE LineItems
  (
    OrderId     INTEGER NOT NULL,
    CustId      VARCHAR2(60) NOT NULL,
    ProductId   INTEGER NOT NULL,
    Price       NUMBER(19,4),
    Qty         NUMBER,
    CONSTRAINT  pk_items PRIMARY KEY (CustId, OrderId, ProductId),
    CONSTRAINT  fk_items_parent FOREIGN KEY (CustId, OrderId)
    REFERENCES Orders ON DELETE CASCADE
  ) PARTITION BY REFERENCE (fk_items_parent);

Table created.

SQL> CREATE DUPLICATED TABLE Products
  (
    ProductId  INTEGER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,
    Name       VARCHAR2(128),
    DescrUri   VARCHAR2(128),
    LastPrice  NUMBER(19,4)
  ) TABLESPACE products_tsp;

Table created.

SQL> CREATE SEQUENCE Orders_Seq;

Sequence created.
```

Shard Director에 접속해서 ddl 과정에서 에러가 없었는 지 확인

```
GDSCTL> show ddl
id      DDL Text                                 Failed shards
--      --------                                 -------------
5       grant connect, resource to app_schema
6       grant dba to app_schema
7       grant execute on dbms_crypto to app_s...
8       CREATE TABLESPACE SET TSP_SET_1 using...
9       CREATE TABLESPACE products_tsp datafi...
10      CREATE SHARDED TABLE Customers   (   ...
11      CREATE SHARDED TABLE Orders   (    O...
12      CREATE SEQUENCE Orders_Seq
13      CREATE SHARDED TABLE LineItems   (   ...
14      CREATE MATERIALIZED VIEW "APP_SCHEMA"...

GDSCTL> config shard -shard sh1
```

```
Name: sh1
Shard Group: primary_shardgroup
Status: Ok
State: Deployed
Region: regionora
Connection string: inst-
shard1.subnetshard.demovcn.oraclevcn.com:1521/sh1.subnetshard.demovcn.oraclevcn.com:dedicated
SCAN address:
ONS remote port: 0
Disk Threshold, ms: 20
CPU Threshold, %: 75
Version: 19.0.0.0
Failed DDL:
DDL Error: ---
Failed DDL id:
Availability: ONLINE
Rack:


Supported services
------------------------
Name                                                    Preferred Status
----                                                    --------- ------
oltp_rw_srvc                                             Yes       Enabled

GDSCTL> config shard -shard sh2
Name: sh2
Shard Group: primary_shardgroup
Status: Ok
State: Deployed
Region: regionora
Connection string: inst-
shard2.subnetshard.demovcn.oraclevcn.com:1521/sh2.subnetshard.demovcn.oraclevcn.com:dedicated
SCAN address:
ONS remote port: 0
Disk Threshold, ms: 20
CPU Threshold, %: 75
Version: 19.0.0.0
Failed DDL:
DDL Error: ---
Failed DDL id:
Availability: ONLINE
Rack:


Supported services
------------------------
Name                                                    Preferred Status
----                                                    --------- ------
oltp_rw_srvc                                             Yes       Enabled
```

각 Shard에 접속해서 테이블스페이스와 테이블이 생성되었는지 확인. 아래는 shard1의 결과임. shard2에서도 동일하게 확인

```
[oracle@inst-shard1 ~]$ sqlplus "/as sysdba"
...
SQL> set lines 100
```

```
SQL> col tablespace_name for a30
SQL> select TABLESPACE_NAME, BYTES/1024/1024 MB from sys.dba_data_files order by
tablespace_name;

TABLESPACE_NAME                     MB
------------------------------ ----------
C001TSP_SET_1                      100
C002TSP_SET_1                      100
C003TSP_SET_1                      100
C004TSP_SET_1                      100
...
SYSAUX                             550
SYSTEM                             900
TSP_SET_1                          100
UNDOTBS1                           340
USERS                                5

126 rows selected.

SQL> set linesize 140
SQL> column table_name format a20
SQL> column tablespace_name format a20
SQL> column partition_name format a20
SQL> show parameter db_unique_name

NAME                                 TYPE        VALUE
------------------------------------ ----------- ------------------------------
_gwm_db_unique_name                   string      sh1
db_unique_name                       string      sh1
SQL> select table_name, partition_name, tablespace_name from dba_tab_partitions where
tablespace_name like 'C%TSP_SET_1' order by tablespace_name;

TABLE_NAME           PARTITION_NAME       TABLESPACE_NAME
-------------------- -------------------- --------------------
LINEITEMS            CUSTOMERS_P1         C001TSP_SET_1
ORDERS               CUSTOMERS_P1         C001TSP_SET_1
CUSTOMERS            CUSTOMERS_P1         C001TSP_SET_1
ORDERS               CUSTOMERS_P2         C002TSP_SET_1
CUSTOMERS            CUSTOMERS_P2         C002TSP_SET_1
...
LINEITEMS            CUSTOMERS_P120       C03CTSP_SET_1
ORDERS               CUSTOMERS_P120       C03CTSP_SET_1

360 rows selected.
```

Shard Catalog DB에 접속해서 chunk가 균일하게 분산되었는지 확인.

```
SQL> col shard for a20
SQL> SELECT a.name Shard, COUNT(b.chunk_number) Number_of_Chunks FROM gsmadmin_internal.database
a, gsmadmin_internal.chunk_loc b WHERE a.database_num=b.database_num GROUP BY a.name ORDER BY
a.name;

SHARD                NUMBER_OF_CHUNKS
-------------------- ----------------
sh3                               120
sh4                               120
```

샘플 테이블이 생성되었는지 Shard Catalog와 Shard1, Shard2에서 확인. duplicate 테이블은 mview 형태로 구현된 것을 알 수 있다.

```
-- Shard Catalog
SQL> select table_name from dba_tables where owner='APP_SCHEMA';

TABLE_NAME
--------------------------------------------------------------------------------
CUSTOMERS
ORDERS
LINEITEMS
RUPD$_PRODUCTS
PRODUCTS
MLOG$_PRODUCTS

6 rows selected.

-- Shard1
SQL> select table_name from dba_tables where owner='APP_SCHEMA';

TABLE_NAME
--------------------
CUSTOMERS
ORDERS
LINEITEMS
USLOG$_PRODUCTS
PRODUCTS

SQL> select table_name from dba_tables where owner='APP_SCHEMA';

-- Shard2
TABLE_NAME
--------------------------------------------------------------------------------
CUSTOMERS
ORDERS
LINEITEMS
USLOG$_PRODUCTS
PRODUCTS
```

샘플 데이터 입력. 여기서는 Shrard Director에서 서비명으로 easy connect로 접속해서 확인함.

```
[oracle@inst-sdirector ~]$ lsnrctl status

LSNRCTL for Linux: Version 19.0.0.0.0 - Production on 04-DEC-2019 11:47:49

Copyright (c) 1991, 2019, Oracle.  All rights reserved.

Connecting to (ADDRESS=(PROTOCOL=tcp)(HOST=)(PORT=1521))
STATUS of the LISTENER
------------------------
Alias                     SDIRECTOR
Version                   TNSLSNR for Linux: Version 19.0.0.0.0 - Production
Start Date                04-DEC-2019 13:06:59
Uptime                    0 days 7 hr. 40 min. 49 sec
Trace Level               off
Security                  ON: Local OS Authentication
SNMP                      OFF
Listener Parameter File   /u01/app/oracle/product/19c/dbhome_1/network/admin/gsm.ora
```

```
Listener Log File        /u01/app/oracle/diag/gsm/inst-sdirector/sdirector/alert/log.xml
Listening Endpoints Summary...
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=inst-
sdirector.subnetsdirector.demovcn.oraclevcn.com)(PORT=1521)))
Services Summary...
Service "GDS$CATALOG.oradbcloud" has 1 instance(s).
  Instance "SCATALOG", status READY, has 1 handler(s) for this service...
Service "GDS$COORDINATOR.oradbcloud" has 1 instance(s).
  Instance "SCATALOG", status READY, has 1 handler(s) for this service...
Service "_MONITOR" has 1 instance(s).
  Instance "SDIRECTOR", status READY, has 1 handler(s) for this service...
Service "_PINGER" has 1 instance(s).
  Instance "SDIRECTOR", status READY, has 1 handler(s) for this service...
Service "oltp_rw_srvc.demo_sdb.oradbcloud" has 2 instance(s).
  Instance "demo_sdb%1", status READY, has 1 handler(s) for this service...
  Instance "demo_sdb%11", status READY, has 1 handler(s) for this service...
The command completed successfully
[oracle@inst-sdirector ~]$ sqlplus
app_schema/Welcome123##@132.145.146.238:1521/oltp_rw_srvc.demo_sdb.oradbcloud

SQL*Plus: Release 19.0.0.0.0 - Production on Wed Dec 4 11:48:04 2019
Version 19.3.0.0.0

Copyright (c) 1982, 2019, Oracle.  All rights reserved.

Last Successful login time: Wed Dec 04 2019 11:29:51 +00:00

Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.3.0.0.0

SQL> insert into app_schema.customers values (1,'Snow','John','King','North',null, null);

1 row created.

SQL> commit;

Commit complete.
```

여기에서는 데이터가 shard1로 들어갔음을 알 수 있음.

```
-- 1번 shard
SQL> select current_scn from v$database;

CURRENT_SCN
-----------
    2088182

SQL> select firstname from app_schema.customers;

FIRSTNAME
------------------------------------------------------------
Snow

SQL> select current_scn from v$database;

CURRENT_SCN
```

```
-----------
    2088585

-- 2번 shard
SQL> select current_scn from v$database;

CURRENT_SCN
-----------
    2088250

SQL> select firstname from app_schema.customers;

no rows selected

SQL> select current_scn from v$database;

CURRENT_SCN
-----------
    2088572
```