## 准备工作

### 停止db和service

|  |
| --- |
| SQL> shutdown immediate |

### 删除旧库

1. 信息确认

|  |
| --- |
| SQL> select status from gv$instance;  STATUS  ------------  OPEN  OPEN  SQL> select open\_mode,database\_role from gv$database;  OPEN\_MODE DATABASE\_ROLE  -------------------------------- -------------------------  READ ONLY WITH APPLY PHYSICAL STANDBY |

1. dbca删库

略。

1. 清空/oradata,/oraarch,/oraredo磁盘组

使用rm指令删除/oradata,/oraarch,/oraredo下文件夹以及文件。

rm -rf /oradata/\* /oraarch/\* /oraredo/\* /app/oracle/admin/sgetradbdg/adump

mkdir /oraarch/sgetradbdg

mkdir /oradata/sgetradbdg

mkdir /oraredo/sgetradbdg

mkdir -p /app/oracle/admin/sgetradbdg/adump

### 拷贝密码文件

1. 在备库端将主库传输过来的口令文件拷贝到$ORACLE\_HOME/dbs下并重命名：

|  |
| --- |
| [oracle@traracdb1 ~]$ cp /backup/rmanfromsh/traracdb1\_20190606/orapwsgetradb1 /app/oracle/product/11.2.0/db\_1/dbs/orapwsgetradbdg |

## 修改参数文件

### 修改备库参数

1. 修改备库参数文件

vim /app/oracle/product/11.2.0/db\_1/dbs/pfile`date +%Y%m%d`.ora

(需要调整操作如下)

|  |
| --- |
| \*.\_cleanup\_rollback\_entries=2000  \*.\_datafile\_write\_errors\_crash\_instance=FALSE  \*.\_disable\_streams\_pool\_auto\_tuning=TRUE  \*.\_optimizer\_mjc\_enabled=FALSE  \*.\_optimizer\_use\_feedback=FALSE  \*.\_PX\_use\_large\_pool=TRUE  \*.archive\_lag\_target=3600  \*.audit\_file\_dest='/app/oracle/admin/sgetradbdg/adump'  \*.audit\_sys\_operations=TRUE  \*.audit\_trail='XML','EXTENDED'  \*.compatible='11.2.0.4.0'  \*.control\_file\_record\_keep\_time=30  \*.control\_files='/oradata/sgetradbdg/control01.ctl','/oraredo/sgetradbdg/control02.ctl','/oraarch/sgetradbdg/control03.ctl'  \*.db\_block\_checking='FULL'  \*.db\_block\_size=8192  \*.db\_cache\_size=25769803776  \*.db\_create\_file\_dest='/oradata'  \*.db\_domain=''  \*.db\_file\_name\_convert='+data/sgetradb','/oradata/sgetradbdg','+arch/sgetradb','/oraarch/sgetradbdg'  \*.db\_files=2048  \*.db\_flashback\_retention\_target=240  \*.db\_name='sgetradb'  \*.db\_recovery\_file\_dest='/oraarch'  \*.db\_recovery\_file\_dest\_size=53687091200  \*.db\_unique\_name='sgetradbdg'  \*.db\_writer\_processes=2  \*.deferred\_segment\_creation=FALSE  \*.diagnostic\_dest='/app/oracle'  \*.dispatchers='(PROTOCOL=TCP) (SERVICE=sgetradbdgXDB)'  \*.enable\_ddl\_logging=TRUE  \*.event='28401 TRACE NAME CONTEXT FOREVER, LEVEL 1'  \*.fal\_client='sgetradbdg'  \*.fal\_server='sgetradb'  \*.filesystemio\_options='SETALL'  \*.job\_queue\_processes=10  \*.large\_pool\_size=268435456  \*.log\_archive\_config='DG\_CONFIG=(sgetradb,sgetradbdg)'  \*.log\_archive\_dest\_1='LOCATION=/oraarch'  \*.log\_archive\_dest\_3='SERVICE=sgetradb ASYNC compression=enable valid\_for=(ONLINE\_LOGFILE,PRIMARY\_ROLE) DB\_UNIQUE\_NAME=sgetradb'  \*.log\_archive\_dest\_state\_3='enable'  \*.log\_archive\_format='%t\_%s\_%r.dbf'  \*.log\_archive\_max\_processes=4  \*.log\_file\_name\_convert='+data/sgetradb','/oraredo/sgetradbdg','+arch/sgetradb','/oraarch/sgetradbdg' #redo log  \*.open\_cursors=1000  \*.parallel\_max\_servers=12  \*.pga\_aggregate\_target=25769803776  \*.processes=2000  \*.remote\_login\_passwordfile='EXCLUSIVE'  \*.resource\_limit=TRUE  \*.resource\_manager\_plan=''  \*.service\_names='sgetradb,sgetra'  \*.session\_cached\_cursors=100  \*.sga\_target=85899345920  \*.standby\_file\_management='AUTO'  \*.undo\_tablespace='UNDOTBS1'  ##按照灾备环境的进行配置，可直接复制以下内容替换到文件中  #\*.log\_file\_name\_convert='/oraredo/sgetradb','/oraredo/sgetradbdg','/oraarch/sge#tradb','/oraarch/sgetradbdg'  #\*.db\_create\_online\_log\_dest\_1='/oraredo'  #\*.db\_create\_online\_log\_dest\_2='/oraarch' |

#注：如果log\_file\_name\_convert里面有logfile,需要提前在/oraarch和/oradata添加目录。

1. 创建文件夹(所有节点都要创建)

|  |
| --- |
| mkdir -p /app/oracle/admin/sgetradbdg/adump |

1. 注意参数文件修改与核对
2. 注意两个节点的口令文件的权限，位置和命名orapwsgetradbdg
3. tnsping sgetradb检查网络服务名是否ping通

## 创建并恢复备库实例

### 恢复参数文件

|  |
| --- |
| SQL> create spfile from pfile='/app/oracle/product/11.2.0/db\_1/dbs/pfile`date +%Y%m%d`.ora';  SQL> startup nomount |

### 恢复控制文件

|  |
| --- |
| [oracle@traracdb1 20190607]$ rman target /  RMAN> restore standby controlfile from '/backup/rmanfromsh/traracdb1\_20190606/ctl\_stand\_con.ctl';  RMAN> alter database mount; |

### 恢复数据库

1. restore database前其他节点保持关闭状态

|  |
| --- |
| RMAN> catalog start with '/backup/rmanfromsh/traracdb1\_20190606/';  RMAN> crosscheck backupset;  RMAN> delete noprompt expired backup;  RMAN> restore database; |

1. 启动数据库到mount状态

|  |
| --- |
| SQL> shutdown immediate  SQL> startup mount |

## 创建standby redo

### 查看主库日志文件大小

备注：主/备库端操作

|  |
| --- |
| col member for a60  col bytes for 99999999  col status for a10  col type for a20  set line 180 pages 999  SELECT L.GROUP#, L.THREAD#, LF.MEMBER, L.BYTES/1024/1024, L.STATUS, LF.TYPE  FROM V$LOG L, V$LOGFILE LF  WHERE L.GROUP# = LF.GROUP#; |

### 创建standby日志组

以下为主备库均无standby log情况下操作。

|  |
| --- |
| select group#,status,type,member from v$logfile;  Alter database drop standby logfile group N;  … |

1. 添加**主备库**端两节点的standby日志组

添加规则：

standby redo log组数公式 >= (每个instance日志组个数+1)\*instance个数

|  |
| --- |
| alter database add standby logfile thread 1 group 10 ('/oraredo/sgetradbdg/redo10a.log','/oraarch/sgetradbdg/redo10b.log') size 1024M;  alter database add standby logfile thread 1 group 11 ('/oraredo/sgetradbdg/redo11a.log','/oraarch/sgetradbdg/redo11b.log') size 1024M;  alter database add standby logfile thread 1 group 12 ('/oraredo/sgetradbdg/redo12a.log','/oraarch/sgetradbdg/redo12b.log') size 1024M;  alter database add standby logfile thread 1 group 13 ('/oraredo/sgetradbdg/redo13a.log','/oraarch/sgetradbdg/redo13b.log') size 1024M;  alter database add standby logfile thread 1 group 14 ('/oraredo/sgetradbdg/redo14a.log','/oraarch/sgetradbdg/redo14b.log') size 1024M;  alter database add standby logfile thread 1 group 15 ('/oraredo/sgetradbdg/redo15a.log','/oraarch/sgetradbdg/redo15b.log') size 1024M;  alter database add standby logfile thread 2 group 16 ('/oraredo/sgetradbdg/redo16a.log','/oraarch/sgetradbdg/redo16b.log') size 1024M;  alter database add standby logfile thread 2 group 17 ('/oraredo/sgetradbdg/redo17a.log','/oraarch/sgetradbdg/redo17b.log') size 1024M;  alter database add standby logfile thread 2 group 18 ('/oraredo/sgetradbdg/redo18a.log','/oraarch/sgetradbdg/redo18b.log') size 1024M;  alter database add standby logfile thread 2 group 19 ('/oraredo/sgetradbdg/redo19a.log','/oraarch/sgetradbdg/redo19b.log') size 1024M;  alter database add standby logfile thread 2 group 20 ('/oraredo/sgetradbdg/redo20a.log','/oraarch/sgetradbdg/redo20b.log') size 1024M;  alter database add standby logfile thread 2 group 21 ('/oraredo/sgetradbdg/redo21a.log','/oraarch/sgetradbdg/redo21b.log') size 1024M; |

### 备库介质恢复

备注：使用节点1开启mrp

|  |
| --- |
| alter database recover managed standby database using current logfile  disconnect from session nodelay; |

### 开启日志投递

备注：主库端操作

|  |
| --- |
| alter system set log\_archive\_dest\_state\_3='enable' scope=both sid='\*';  alter system switch logfile; |

## 同步状态检查

### 查看备库警告日志

略。

### mrp进程状态检查

|  |
| --- |
| select inst\_id,process,status,thread#,sequence#,block# from gv$managed\_standby where PROCESS like 'MRP%'; |

### 外部归档文件处理

注意：外部存档日志文件空间，如空间不足请及时删除(**文档 ID** **1617965.1**)

|  |
| --- |
| SQL> select \* from V$FLASH\_RECOVERY\_AREA\_USAGE;  RMAN> delete foreign archivelog all; |

### 日志接收与同步检查

|  |
| --- |
| 备库：  -- 检查备库已经接收到的 sequence# 号  select thread#,max(sequence#) "Last Standby Seq Received" from v$archived\_log val,v$database vdb where val.resetlogs\_change#=vdb.resetlogs\_change# group by thread# order by 1;  -- 检查备库已经应用到的 sequence# 号  select thread#,max(sequence#) "Last Standby Seq Applied" from v$archived\_log val,v$database vdb where val.resetlogs\_change#=vdb.resetlogs\_change# and val.applied in ('YES','IN-MEMORY') group by thread# order by 1; |

### 延迟查询

|  |
| --- |
| select name,value from v$dataguard\_stats; |

#日志应用完成

alter database recover managed standby database cancel;

alter database open;

SQL> alter database recover managed standby database using current logfile

disconnect from session nodelay;

## SNAPSHOT STANDBY相关配置

|  |
| --- |
| SQL> show parameter recovery  SQL> alter system set DB\_RECOVERY\_FILE\_DEST\_SIZE=20G scope=both sid='\*';  System altered.  SQL> alter system set db\_recovery\_file\_dest='+DATA' scope=both sid='\*';  System altered. |

## 检查监听服务状态

|  |
| --- |
| lsnrctl status  ###lsnrctl status LISTENER\_SCAN1 |

## 验证备库状态

### 进行snapshot standby切换测试

|  |
| --- |
| SQL> select name,value from v$dataguard\_stats;  SQL> alter database recover managed standby database cancel;  SQL>shutdown immediate  SQL>startup mount  SQL> alter database convert to snapshot standby;  SQL>shutdown immediate  SQL>startup  SQL> select database\_role,open\_mode from v$database;  SQL>shutdown immediate  SQL>startup mount  SQL> alter database convert to physical standby;  SQL>shutdown immediate  SQL>startup  SQL> alter database recover managed standby database using current logfile disconnect from session nodelay;  SQL> select database\_role,open\_mode from v$database; |

## 验证备库状态

### 进行snapshot standby切换测试

|  |
| --- |
| SQL> select name,value from v$dataguard\_stats;  SQL> alter database recover managed standby database cancel;  SQL>shutdown immediate  SQL>startup mount  SQL> alter database convert to snapshot standby;  SQL>shutdown immediate  SQL>startup  SQL> select database\_role,open\_mode from v$database;  SQL>shutdown immediate  SQL>startup mount  SQL> alter database convert to physical standby;  SQL>shutdown immediate  SQL>startup  SQL> alter database recover managed standby database using current logfile disconnect from session nodelay;  SQL> select database\_role,open\_mode from v$database; |

# 附件-主库操作步骤

备注：主库端操作

### RMAN备份主库

备注：主库端操作

1. 查看数据库大小

|  |
| --- |
| SQL> SELECT SUM(DS.BYTES)/1024/1024/1024 "SIZE(G)" FROM DBA\_SEGMENTS ds; |

1. 编写脚本进行全库备份

|  |
| --- |
| [oracle@rac1 bak]$ vi fullbak.sh  export ORACLE\_SID= sgetradb1  export ORACLE\_BASE=/app/oracle  export ORACLE\_HOME=$ORACLE\_BASE/product/11.2.0/db\_1  export LD\_LIBRARY\_PATH=$ORACLE\_HOME/lib  export NLS\_DATE\_FORMAT="yyyy-mm-dd hh24:mi:ss"  export NLS\_LANG="AMERICAN\_AMERICA.ZHS16GBK"  export PATH=$PATH:$HOME/bin:$ORACLE\_HOME/bin  bak\_date=$(date '+%Y%m%d')  bak\_dir=/bak/$bak\_date  mkdir -p $bak\_dir  chmod -R 755 /bak  rman target / <<EOF  run{  CONFIGURE DEVICE TYPE DISK PARALLELISM 4 BACKUP TYPE TO COMPRESSED BACKUPSET;  configure channel device type disk maxpiecesize 4G;  BACKUP  FORMAT='$bak\_dir/data\_%U\_%T.dbf'  DATABASE;  BACKUP SPFILE FORMAT '$bak\_dir/spfile\_%U\_%T.ora';  backup current controlfile for standby format'$bak\_dir/ctl\_stand\_con.ctl';  }  EXIT;  EOF |

|  |
| --- |
| [oracle@rac1 bak]$ chmod 755 fullbak.sh  [oracle@rac1 bak]$ nohup sh fullbak.sh > fullbak.log & |

说明：做以上备份操作后，参考2.4.1节将主库的参数文件进行备份

### 是否安装相关组件

|  |
| --- |
| SQL> SELECT \* FROM V$OPTION WHERE PARAMETER in ('Oracle Data Guard', 'Advanced Compression');  PARAMETER VALUE  ------------------------------- ----------  Oracle Data Guard TRUE  Advanced Compression TRUE |

### FORCE\_LOGGING模式

|  |
| --- |
| select force\_logging from v$database;  (开启：alter database force logging;) |

### 最小附件日志

|  |
| --- |
| SQL> SELECT INST\_ID,SUPPLEMENTAL\_LOG\_DATA\_MIN FROM GV$DATABASE;  INST\_ID SUPPLEME  ---------- --------------  1 YES  2 YES |

### 归档模式

|  |
| --- |
| SQL> archive log list;  Database log mode Archive Mode  Automatic archival Enabled  Archive destination +DATA  Oldest online log sequence 63  Next log sequence to archive 64  Current log sequence 64 |

### remote\_login\_passwordfile配置

|  |
| --- |
| show parameter remote\_login\_passwordfile  (remote\_login\_passwordfile应为EXCLUSIVE) |

### 修改主库参数

备注：主库端操作

1. 将主库备份的参数文件传送至备库进行修改

|  |
| --- |
| su - oracle  sqlplus / as sysdba  create pfile='/tmp/oracle/initsgetradb1.ora.bak' from spfile; |

1. 修改主库参数

|  |
| --- |
| alter system set log\_archive\_config='dg\_config=(sgetradb,sgetradg,sgetradbdg)' scope=both sid='\*';  alter system set log\_archive\_dest\_state\_3='defer' scope=both sid='\*';  #等备库实例启动，再开启enabled  alter system set log\_archive\_dest\_3=  'service=sgetradbdg ASYNC compression=enable valid\_for=(ONLINE\_LOGFILE,PRIMARY\_ROLE) DB\_UNIQUE\_NAME='sgetradbdg' scope=both sid='\*';  alter system set fal\_client=' sgetradb' scope=both sid='\*';  alter system set fal\_server='sgetradg' ,'sgetradbdg' scope=both sid='\*';  alter system set standby\_file\_management=AUTO scope=both sid='\*';  alter system set log\_archive\_max\_processes = 8 scope=both sid='\*'; |

### 备份文件传输至备库

备注：主库端操作

将2.2.7节的全库备份通过灾备复制网络传输至深圳traracdb1服务器的/backup目录中。

# 附件

## 附件一：备库搭建重要参数检查

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 检查项 | 参考值 | 检查命令 |
| 1 | 密码文件名称 | 节点1:traracdb1:orapwsgetradbdg1  节点2:traracdb2:orapwsgetradbdg2 | ls $ORACLE\_HOME/dbs/orapw\* |
| 2 | pfile | +DATA/sgetradbdg/spfilesgetradbdg.ora | show parameter spfile(两个节点都检查) |
| 3 | db\_file\_name\_convert | sgetradb, sgetradbdg | show parameter convert |
| 4 | log\_file\_name\_convert | sgetradb, sgetradbdg | show parameter convert |
| 5 | log\_archive\_config | dg\_config=(sgetradb,sgetradbdg) | show parameter log\_archive\_config |
| 6 | fal\_client | sgetradbdg | show parameter fal |
| 7 | fal\_server | sgetradb | show parameter fal |
| 8 | listener\_networks | ((NAME=network1)(LOCAL\_LISTENER=listener\_net1)(REMOTE\_LISTENER=traracdb-scan:1521))','((NAME=network2)(LOCAL\_LISTENER=listener\_net2)(REMOTE\_LISTENER=remote\_net2)) | show parameter networks |
| 9 | log\_archive\_dest\_1 | LOCATION=+ARCH VALID\_FOR=(ALL\_LOGFILES,ALL\_ROLES) DB\_UNIQUE\_NAME=sgetradbdg | show parameter log\_archive\_dest\_1 |
| 10 | db\_unique\_name | sgetradbdg | show parameter db\_uniq |