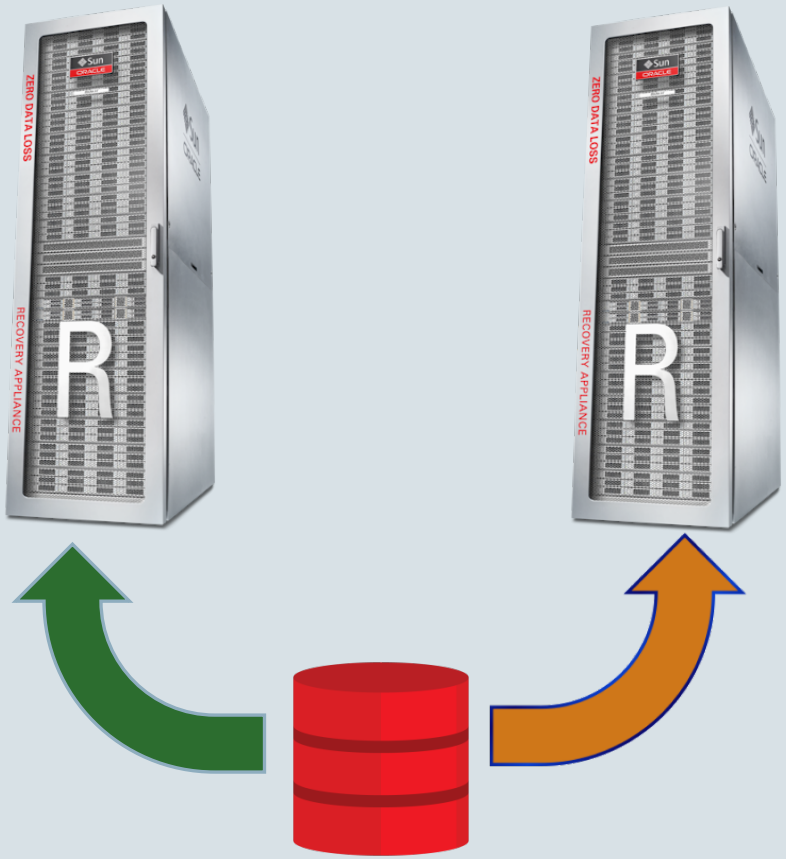

ORACLE®

ZDLRA High Availability for Backup and Recovery

Oracle Server Technology
High Availability Systems Development
Maximum Availability Architecture
September 2018



Safe Harbor Statement

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

Program Agenda

1. Solution overview
2. Requirements
3. Setup and Configuration
4. Operations

Zero Data Loss Recovery Appliance

Engineered Data Protection For Complete Recoverability of the Oracle Database

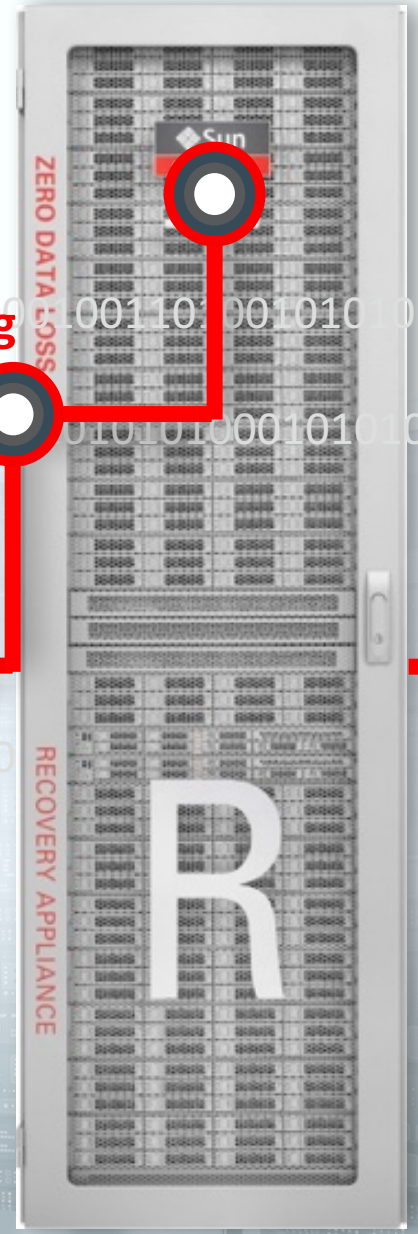
Enterprise Databases & Applications

End-to-End Data Protection Control & Monitoring

Eliminate Long Backup Windows

Continually Validates Recovery Status

Reliable & Complete Recovery



Scale-Out Hardware & Storage

Archive Backup to Cloud Storage

ORACLE

ZDLRA HA for Backup & Recovery MAA Configuration

- The Oracle Maximum Availability Architecture (MAA) best practice to protect the ZDLRA against site disasters and system outages is to have a ZDLRA replica.
- With a ZDLRA replica, protected database backup, redo, and restore operations continue, preserving complete data protection.
- This presentation will address how to configure protected database failover to a downstream ZDLRA when the upstream Recovery Appliance is unavailable due to unplanned or planned outages.
- The focus is on RMAN backup and redo operations. After configuration, no client-side changes are required for the daily RMAN backup operations when the upstream Recovery Appliance has planned or unplanned outages.
- For the detailed configuration steps, refer to My Oracle Support note:
  Configuring High Availability ZDLRA Client for Backup and Restore (Doc ID 2432144.1)

Solution Overview

This presentation provides the steps required to configure a protected database for transparent failover of backup operations and – when used - redo transport to a downstream Recovery Appliance. The following is assumed:

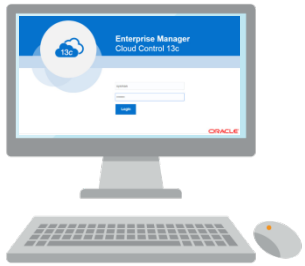
1. Real time redo transport is enabled. During the outage of the upstream ZDLRA, the redo transport will fail over to downstream ZDLRA
2. The protected databases used in the subsequent examples are 12.1 (CDB121DR) and 12.2 (CDB122DR) container databases with one pluggable database each.
3. An upstream Recovery Appliance (RAHADR1)
4. A downstream Recovery Appliance (RAHADR2)
5. A common VPC user called HADR_COMMON_VPCUSER was created on both Recovery Appliances and **must** use the same password on both.
6. A local VPC user called HADR_LOCAL_VPCUSER created on both Recovery Appliances but the password can be different between the two.
7. A replication server between RAHADR1 and RAHADR2 is using the VPC user REPUSER_FROM_HADR1

Program Agenda

1. Solution overview
2. Requirements
3. Setup and Configuration
4. Operations

Requirements

Enterprise Manager 13 Release 2
(EM 13.2.2.0.180430) or later



Upstream Recovery Appliance
Release: 12.2.1.1.1.201805-RELEASE
or later

Downstream Recovery Appliance
Release: 12.2.1.1.1.201805-RELEASE
or later

RDBMS 12.1.0.2.171017 or later +
One-Off Patch 25926338

- OR -

RDBMS 12.2.0.1.180116 or later +
One-Off Patch 27757888

Program Agenda

1. Solution overview
2. Requirements
3. Setup and Configuration
4. Operations

Setup and Configuration (ZDLRA) - VPC Users Creation

RAHADR1

On **each** ZDLRA (RAHADR1 and RAHADR2),
create two VPC users for the protected database

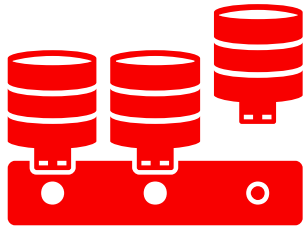
```
# racli add vpc_user --user_name HADR_LOCAL_VPCUSER
[HADR_LOCAL_VPCUSER] New Password: hadr1_L0cal_Pa55w0rd
Sun Mar 25 08:27:53 2018: Start: Add vpc user HADR_LOCAL_VPCUSER.
Sun Mar 25 08:27:53 2018: Add vpc user HADR_LOCAL_VPCUSER successfully.
Sun Mar 25 08:27:53 2018: End: Add vpc user HADR_LOCAL_VPCUSER.

# racli add vpc_user --user_name HADR_COMMON_VPCUSER
[HADR_COMMON_VPCUSER] New Password: c0mm0n_Pa55w0rd
Sun Mar 25 08:27:53 2018: Start: Add vpc user HADR_COMMON_VPCUSER.
Sun Mar 25 08:27:53 2018: Add vpc user HADR_COMMON_VPCUSER successfully.
Sun Mar 25 08:27:53 2018: End: Add vpc user HADR_COMMON_VPCUSER.
```

RAHADR2

On **the downstream** ZDLRA (RAHADR2),
create the VPC user to be used by the replication server

```
# racli add vpc_user --user_name REPUSER_FROM_HADR1
[REPUSER_FROM_HADR1] New Password: v3ry_c0mplex_pa55w0rd
Sun Mar 25 08:35:01 2018: Start: Add vpc user REPUSER_FROM_HADR1.
Sun Mar 25 08:35:01 2018: Add vpc user REPUSER_FROM_HADR1 successfully.
Sun Mar 25 08:35:01 2018: End: Add vpc user REPUSER_FROM_HADR1.
```

Setup and Configuration (Protected Database) - tnsnames.ora

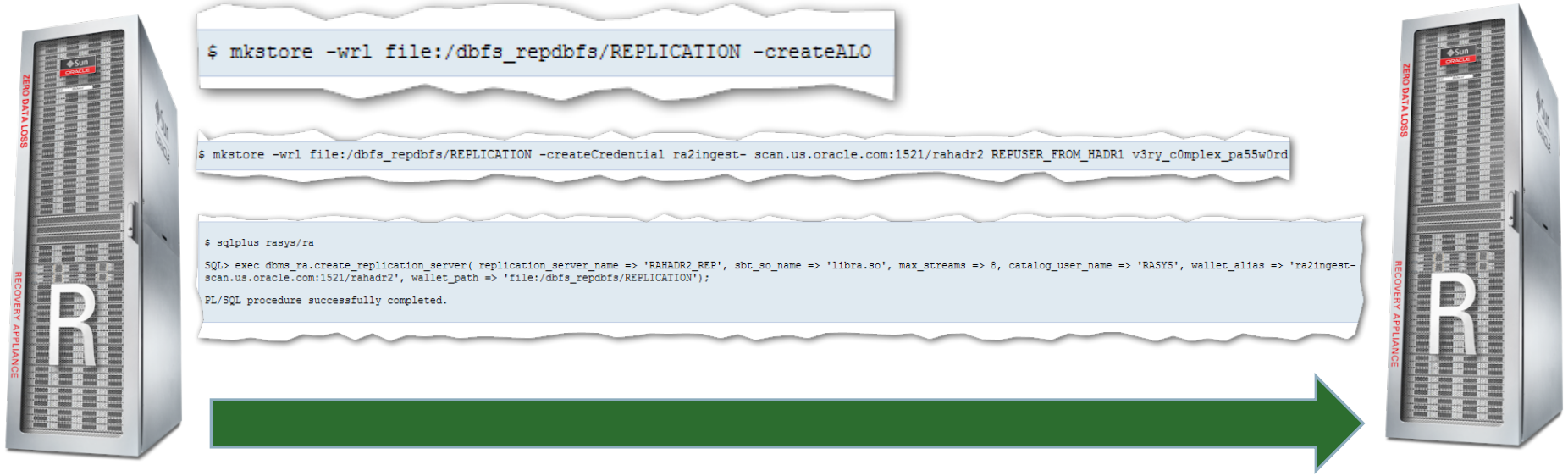


On the protected database and on each database server (if it is RAC configured) update the tnsnames.ora under the oracle database home.

IMPORTANT: If ra_install.jar was used on the protected database, ensure that there are no ${ORACLE_HOME}/dbs/ra${ORACLE_SID}.ora files on any of the database servers. The presence of this file has the effect of overriding all the configuration parameters defined further.

```
DR_RAHADR =
  (DESCRIPTION_LIST =
    (LOAD_BALANCE = off)
    (FAILOVER = on)
    (DESCRIPTION =
      (CONNECT_TIMEOUT = 5)
      (TRANSPORT_CONNECT_TIMEOUT = 3)
      (RETRY_COUNT = 3)
      (ADDRESS_LIST =
        (ADDRESS = (PROTOCOL = TCP)(HOST = ra1ingest-scan)(PORT = 1521))
      )
      (CONNECT_DATA =
        (SERVICE_NAME = rahadr1)
      )
    )
    (DESCRIPTION =
      (CONNECT_TIMEOUT = 5)
      (TRANSPORT_CONNECT_TIMEOUT = 3)
      (RETRY_COUNT = 3)
      (ADDRESS_LIST =
        (ADDRESS = (PROTOCOL = TCP)(HOST = ra2ingest-scan)(PORT = 1521))
      )
      (CONNECT_DATA =
        (SERVICE_NAME = rahadr2)
      )
    )
  )

DR_RAHADR1 =
  (DESCRIPTION_LIST =
    (DESCRIPTION =
      (CONNECT_TIMEOUT = 5)
      (TRANSPORT_CONNECT_TIMEOUT = 3)
      (RETRY_COUNT = 3)
      (ADDRESS_LIST =
        (ADDRESS = (PROTOCOL = TCP)(HOST = ra1ingest-scan)(PORT = 1521))
      )
      (CONNECT_DATA =
        (SERVICE_NAME = rahadr1)
      )
    )
  )

DR_RAHADR2 =
  (DESCRIPTION_LIST =
    (DESCRIPTION =
      (CONNECT_TIMEOUT = 5)
      (TRANSPORT_CONNECT_TIMEOUT = 3)
      (RETRY_COUNT = 3)
      (ADDRESS_LIST =
        (ADDRESS = (PROTOCOL = TCP)(HOST = ra2ingest-scan)(PORT = 1521))
      )
      (CONNECT_DATA =
        (SERVICE_NAME = rahadr2)
      )
    )
  )
```

Setup and Configuration (ZDLRA) - Replication Server creation



RAHADR1

RAHADR2

On the Upstream ZDLRA:

1. Create a replication wallet if it does not exist
2. Add the VPC replication credentials to the replication wallet
3. Create the replication server

Setup and Configuration (ZDLRA) - Protected Database Enrollment


```
$ sqlplus rasys/ra

SQL> exec dbms_ra.create_protection_policy( protection_policy_name => 'cdb122dr_PP' numtodsinterval(3,'DAY') numtodsinterval(1.25,'DAY'), storage_location_name => 'DELTA' , recovery_window_goal => ,
unprotected_window => , allow_backup_deletion => 'NO');

PL/SQL procedure successfully completed.

SQL> exec dbms_ra.add_db(db_unique_name => 'cdb122dr', protection_policy_name => 'cdb122dr_PP', reserved_space => '1T');

PL/SQL procedure successfully completed.

SQL> exec dbms_ra.grant_db_access(username => 'REPUSER_FROM_HADR1', db_unique_name => 'cdb122dr');

PL/SQL procedure successfully completed.
```


```
$ sqlplus rasys/ra

SQL> exec dbms_ra.create_protection_policy( protection_policy_name => 'cdb122dr_PP' , storage_location_name => 'DELTA' , recovery_window_goal => numtodsinterval(3,'DAY') , unprotected_window =>
numtodsinterval(5,'MINUTE') , allow_backup_deletion => 'NO');

PL/SQL procedure successfully completed.

SQL> exec dbms_ra.add_db(db_unique_name => 'cdb122dr', protection_policy_name => 'cdb122dr_PP', reserved_space => '1T');

PL/SQL procedure successfully completed.

SQL> exec dbms_ra.grant_db_access(username => 'HADR_LOCAL_VPCUSER', db_unique_name => 'cdb122dr');

PL/SQL procedure successfully completed.
```

```
$ sqlplus rasys/ra

SQL> exec dbms_ra.add_replication_server( replication_server_name => 'RAHADR2_REP', protection_policy_name => 'cdb122dr_PP');

PL/SQL procedure successfully completed.
```

RAHADR1

RAHADR2

On the Upstream ZDLRA:

1. Create a protection policy to be used by the protected database
2. Add the protected database to the ZDLRA
3. Grant the database access to the VPC user

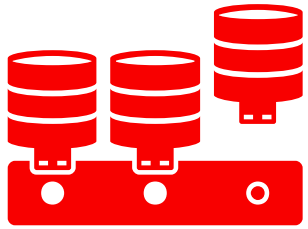
On the Upstream ZDLRA:

1. Add the protection policy to the replication server

On the Downstream ZDLRA:

1. Create a protection policy to be used by the protected database
2. Add the protected database to the ZDLRA
3. Grant the database access to the VPC user

Setup and Configuration (Protected Database) - Registration



On the Protected Database:

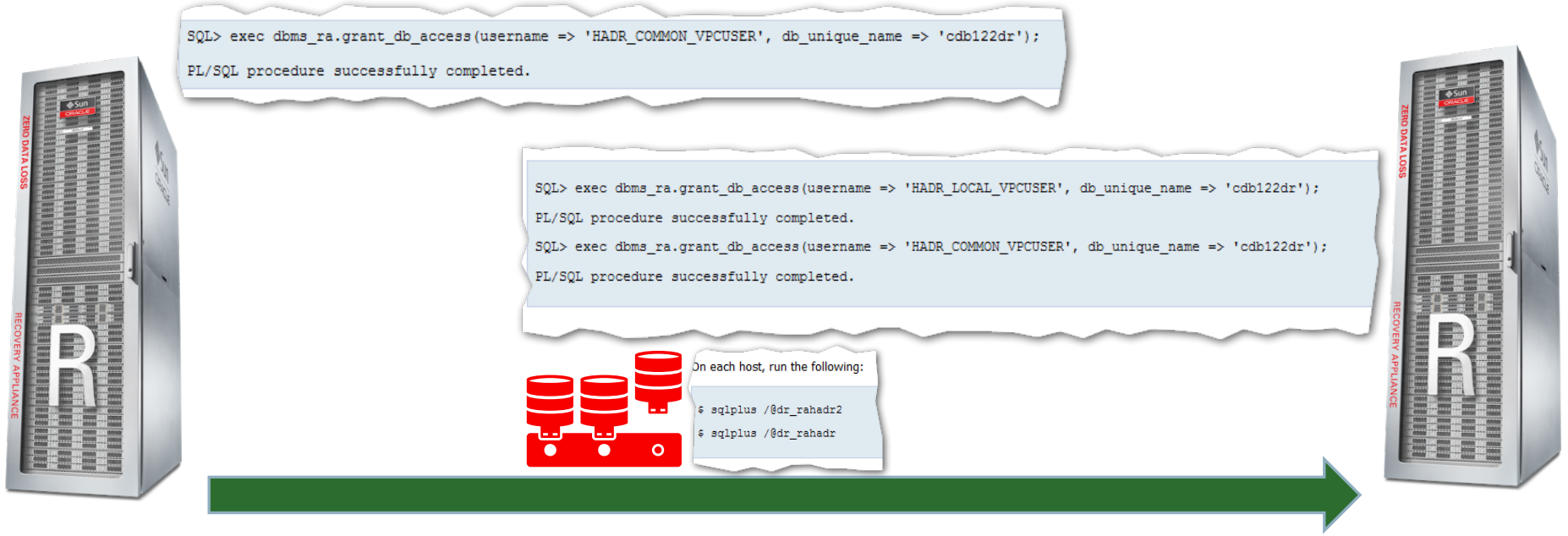
1. Configure the sqlnet.ora file that will be used by RMAN to connect to the correct Recovery Appliance – if the protected database is RAC, this should be done on all nodes.
2. Create auto-login wallet to store the the VPC users credentials.
3. Create credential aliases for each of the three credentials that will be used by RMAN.
4. Verify that the credentials are working correctly
5. Register the protected database with RMAN in the Upstream ZDLRA
6. Perform a test backup of the current controlfile.
7. List the backup you just created to verify it is available on both Upstream and Downstream ZDLRA.

Verify there are two copies of the control file, one on Recovery Appliance hadr1 (RAHADR1) and the other on Recovery Appliance hadr2 (RAHADR2).

```
RMAN> list backupset tag CONTROLTEST;

List of Backup Sets
===================

BS Key Type LV Size
------- ---- -- ----------

220 Full 138.75M
Control File Included: Ckp SCN: 9076177 Ckp time: 05-JUN-18
Backup Set Copy #1 of backup set 220
Device Type Elapsed Time Completion Time Compressed Tag
----------- ------------ --------------- ---------- ---
SBT_TAPE 07:00:21 05-JUN-18 NO CONTROLTEST
List of Backup Pieces for backup set 220 Copy #1
BP Key Pc# Status Media Piece Name
------- --- ----------- ---------------------- ----------
221 1 AVAILABLE Recovery Appliance (RAHADR1) CDB122DR_2kt4m80u_1_1
Backup Set Copy #2 of backup set 220
Device Type Elapsed Time Completion Time Compressed Tag
----------- ------------ --------------- ---------- ---
SBT_TAPE 07:00:21 05-JUN-18 NO CONTROLTEST
List of Backup Pieces for backup set 220 Copy #2
BP Key Pc# Status Media Piece Name
------- --- ----------- ---------------------- ----------
246 1 AVAILABLE Recovery Appliance (RAHADR2) RA_SBT_CDB122DR_3244939197_230_2kt4m80u_1_2_220
```

Note that each ZDLRA holds a copy of the backup piece

Setup and Configuration (ZDLRA) – Add Remaining Grants



RAHADR1

On the Upstream ZDLRA (RAHADR1):

1. Add the grant access to the remaining VPC user (HADR_COMMON_VPCUSER).

RAHADR2

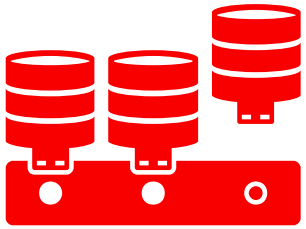
On the Downstream ZDLRA (RAHADR2):

1. Add the grant access to the two remaining VPC users (HADR_LOCAL_VPCUSER and HADR_COMMON_VPCUSER)

From the protected database :

1. Verify that the credentials are working correctly.

Setup and Configuration (Protected Database)
RMAN configuration



On one of the protected database hosts, run:

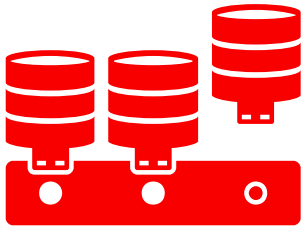
```
$ rman target / catalog /@dr_rahadr1

RMAN> CONFIGURE CHANNEL DEVICE TYPE 'SBT_TAPE' FORMAT '%d_%U' PARMS "SBT_LIBRARY=/u01/app/oracle/product/12.2.0.1/dbhome_1/lib/libra.so, ENV=(RA_WALLET='location=file:/u01/app/oracle/product/12.2.0.1
/dbhome_1/dbs/z dlra credential_alias=dr_rahadr')";
```

On the protected database:

1. Configure the SBT_TAPE channel device parameters for use with the DR_RAHADR alias.
2. Optionally configure additional parameters, which are best practice recommendations.

```
RMAN> CONFIGURE BACKUP OPTIMIZATION on;
RMAN> CONFIGURE CONTROLFILE AUTOBACKUP on;
RMAN> CONFIGURE DEFAULT DEVICE TYPE TO sbt;
RMAN> CONFIGURE DEVICE TYPE SBT_TAPE PARALLELISM 2 BACKUP TYPE TO BACKUPSET;
RMAN> CONFIGURE SNAPSHOT CONTROLFILE NAME TO '+RECOC1/cdb122dr/snapcf.f';
RMAN> CONFIGURE ARCHIVELOG DELETION POLICY TO backed up 1 times to device type sbt;
```

Setup and Configuration (Protected Database)
Real-Time Redo Transport configuration



Real-Time Redo Transport for protected databases can be configured to use the upstream RA **when available, and if not, failover to use the downstream** RA. When the upstream RA **becomes available again, redo transport will automatically use the upstream** RA.

Start by configuring the redo_transport_user and then choose either of the options below.

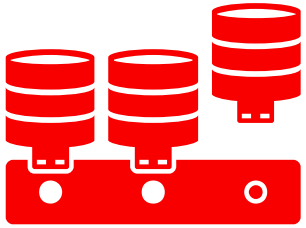
1. Data Guard Broker can be used to manage the settings
2. Init.ora log_archive_dest parameters

```
$ sqlplus / as sysdba

SQL> alter system set redo_transport_user=hadr_local_vpcuser;

System altered.
```

Setup and Configuration (Protected Database)

Real-Time Redo Transport configuration (Option 1)

Configuring Real-Time Redo Transport using Data Guard Broker



Step 1 : Enable the DG_BROKER parameters.*

```
$ sqlplus / as sysdba
SQL> alter system set dg_broker_config_file1='+DATAC1/cdb122dr/dr1cdb122dr.dat';
System altered.
SQL> alter system set dg_broker_config_file2='+DATAC1/cdb122dr/dr2cdb122dr.dat';
System altered.
SQL> alter system set dg_broker_start=true;
System altered.
```

For RDBMS 12.1.0.2

```
$ dgmgrl sys/welcome1
DGMGRL for Linux: Version 12.1.0.2.0 - 64bit Production
Copyright (c) 2000, 2013, Oracle. All rights reserved.
Welcome to DGMGRL, type "help" for information.
```

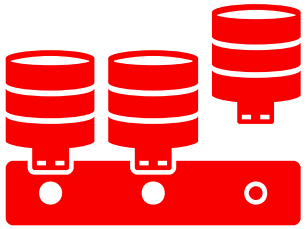
For RDBMS 12.2.0.1

```
$ dgmgrl sys/welcome1
DGMGRL for Linux: Release 12.2.0.1.0 - Production on Tue Jun 5 11:37:44 2018
Copyright (c) 1982, 2017, Oracle and/or its affiliates. All rights reserved.
Welcome to DGMGRL, type "help" for information.
Connected to "cdb122dr"
Connected as SYSDG.
DGMGRL> create configuration cdb122dr as primary database is cdb122dr connect identifier is '//scam06-scan3/cdb122dr';
Configuration "cdb122dr" created with primary database "cdb122dr"
DGMGRL> add recovery_appliance rahadr1 as connect identifier is 'dr_rahadr1';
Recovery Appliance "rahadr1" added
DGMGRL> add recovery_appliance rahadr2 as connect identifier is 'dr_rahadr2';
Recovery Appliance "rahadr2" added
DGMGRL> edit recovery_appliance rahadr1 set property MaxFailure=1;
Property "maxfailure" updated
DGMGRL> edit recovery_appliance rahadr1 set property ReopenSecs=10;
Property "reopensecs" updated
DGMGRL> edit recovery_appliance rahadr1 set property NetTimeout=8;
Property "nettimeout" updated
DGMGRL> edit recovery_appliance rahadr2 set property MaxFailure=1;
Property "maxfailure" updated
DGMGRL> edit recovery_appliance rahadr2 set property NetTimeout=8;
Property "nettimeout" updated
DGMGRL> edit database cdb122dr set property RedoRoutes = '(LOCAL : (rahadr1 async priority=1, rahadr2 async priority=2))';
Warning: ORA-16677: Standby database has the same or higher priority than other members specified in the RedoRoutes group.
Property "redoroutes" updated
DGMGRL> enable configuration;
Enabled.
```

Important: Important: If Redo Transport does not start, attempt to restart database instance to register and enable new wallet configuration.
For a RAC configuration, you restart one RAC instance at a time in a rolling fashion to maintain database service uptime.

Setup and Configuration (Protected Database)
Real-Time Redo Transport configuration (Option 2)

Configuring Real-Time Redo Transport manually



For DBMS 12.2.0.1:

```
$ sqlplus rasys/ra

SQL> alter system set log_archive_config = 'dg_config=(cdb122dr,rahadr1,rahadr2)';

SQL> alter system set log_archive_dest_2='service=dr_rahadr1 ASYNC NOAFFIRM delay=0 optional compression=disable max_failure=1 max_connections=1 reopen=10 db_unique_name=rahadr1 net_timeout=8 group=1
priority=1 valid_for=(online_logfile,all_roles)';

SQL> alter system set log_archive_dest_3='service=dr_rahadr2 ASYNC NOAFFIRM delay=0 optional compression=disable max_failure=1 max_connections=1 reopen=300 db_unique_name=rahadr2 net_timeout=8 group=1
priority=2 valid_for=(online_logfile,all_roles)';

SQL> alter system set log_archive_dest_state_2=enable;

SQL> alter system set log_archive_dest_state_3=enable;
```

For DBMS 12.1.0.2:

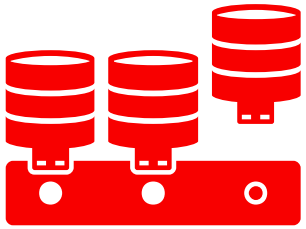
```
$ sqlplus rasys/ra

SQL> alter system set log_archive_config = 'dg_config=(cdb122dr,rahadr1,rahadr2)';

SQL> alter system set log_archive_dest_state_2=reset;

SQL> alter system set log_archive_dest_state_3=reset;

SQL> alter system set log_archive_dest_2='service=dr_rahadr1 ASYNC NOAFFIRM delay=0 optional compression=disable max_failure=1 max_connections=1 reopen=10 db_unique_name=rahadr1 net_timeout=8
alternate=LOG_ARCHIVE_DEST_3 valid_for=(online_logfile,all_roles)';

SQL> alter system set log_archive_dest_3='service=dr_rahadr2 ASYNC NOAFFIRM delay=0 optional compression=disable max_failure=1 max_connections=1 reopen=300 db_unique_name=rahadr2 net_timeout=8
alternate=LOG_ARCHIVE_DEST_2 valid_for=(online_logfile,all_roles)';

SQL> alter system set log_archive_dest_state_2=alternate;

SQL> alter system set log_archive_dest_state_3=alternate;

SQL> alter system set log_archive_dest_state_2=enable;
```

Important: If Redo Shipping does not start, then you may need to bounce the protected database. For a RAC database, ***this can be done in a rolling fashion****.*

Program Agenda

1. Solution overview
2. Requirements
3. Setup and Configuration
4. Operations

Operations (Protected Database) – Create RMAN backup scripts



```
{
allocate channel rahadr1_sbt_1 device type sbt format '%d_%U' PARMS="SBT_LIBRARY=/u01/app/oracle/product/12.2.0.1/dbhome_1/lib/libra.so, ENV=(RA_WALLET='location=file:/u01/app/oracle/product/12.2.0.1/dbhom
e_1/dbs/zdlra credential_alias=dr_rahadr1')";
allocate channel rahadr1_sbt_2 device type sbt format '%d_%U' PARMS="SBT_LIBRARY=/u01/app/oracle/product/12.2.0.1/dbhome_1/lib/libra.so, ENV=(RA_WALLET='location=file:/u01/app/oracle/product/12.2.0.1/dbhom
e_1/dbs/zdlra credential_alias=dr_rahadr1')";
backup tag '&1' cumulative incremental level 1 filesperset 1 section size 64g database plus archivelog not backed up filesperset 32 delete input;
}
```

```
{
allocate channel rahadr2_sbt_1 device type sbt format '%d_%U' PARMS="SBT_LIBRARY=/u01/app/oracle/product/12.2.0.1/dbhome_1/lib/libra.so, ENV=(RA_WALLET='location=file:/u01/app/oracle/product/12.2.0.1/dbhom
e_1/dbs/zdlra credential_alias=dr_rahadr2')";
allocate channel rahadr2_sbt_2 device type sbt format '%d_%U' PARMS="SBT_LIBRARY=/u01/app/oracle/product/12.2.0.1/dbhome_1/lib/libra.so, ENV=(RA_WALLET='location=file:/u01/app/oracle/product/12.2.0.1/dbhom
e_1/dbs/zdlra credential_alias=dr_rahadr2')";
backup tag '&1' cumulative incremental level 1 filesperset 1 section size 64g database plus archivelog not backed up filesperset 32 delete input;
}
```

On the protected database:

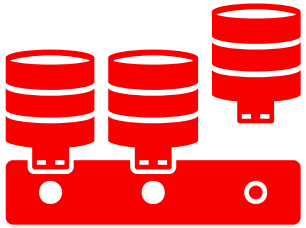
1. On one of the hosts, create the **backup_database_rahadr1.rman** text file.
2. On one of the hosts, create the **backup_database_rahadr2.rman** text file.
3. Load the HADR1 script into the RAHADR1 Recovery Appliance.
4. Load the HADR2 script into the RAHADR2 Recovery Appliance.
5. Verify the RMAN access to the script using the VPC credentials.

```
$ rman target / catalog /@dr_rahadr1
RMAN> delete script backup_database;
RMAN> create script backup_database from file '/home/oracle/backup_database_rahadr1.rman';
```

```
$ rman target / catalog /@dr_rahadr2
RMAN> delete script backup_database;
RMAN> create script backup_database from file '/home/oracle/backup_database_rahadr2.rman';
```

```
$ rman target / catalog /@dr_rahadr
RMAN> print script backup_database;
printing stored script: backup_database
{
allocate channel rahadr1_sbt_1 device type sbt format '%d_%U' PARMS="SBT_LIBRARY=/u01/app/oracle/product/12.2.0.1/dbhome_1/lib/libra.so, ENV=(RA_WALLET='location=file:/u01/app/oracle/product/12.2.0.1/dbhom
e_1/dbs/zdlra credential_alias=dr_rahadr1')";
allocate channel rahadr1_sbt_2 device type sbt format '%d_%U' PARMS="SBT_LIBRARY=/u01/app/oracle/product/12.2.0.1/dbhome_1/lib/libra.so, ENV=(RA_WALLET='location=file:/u01/app/oracle/product/12.2.0.1/dbhom
e_1/dbs/zdlra credential_alias=dr_rahadr1')";
backup tag '&1' cumulative incremental level 1 filesperset 1 section size 64g database plus archivelog not backed up filesperset 32 delete input;
}
```

Operations (Protected Database) – Performing Backups

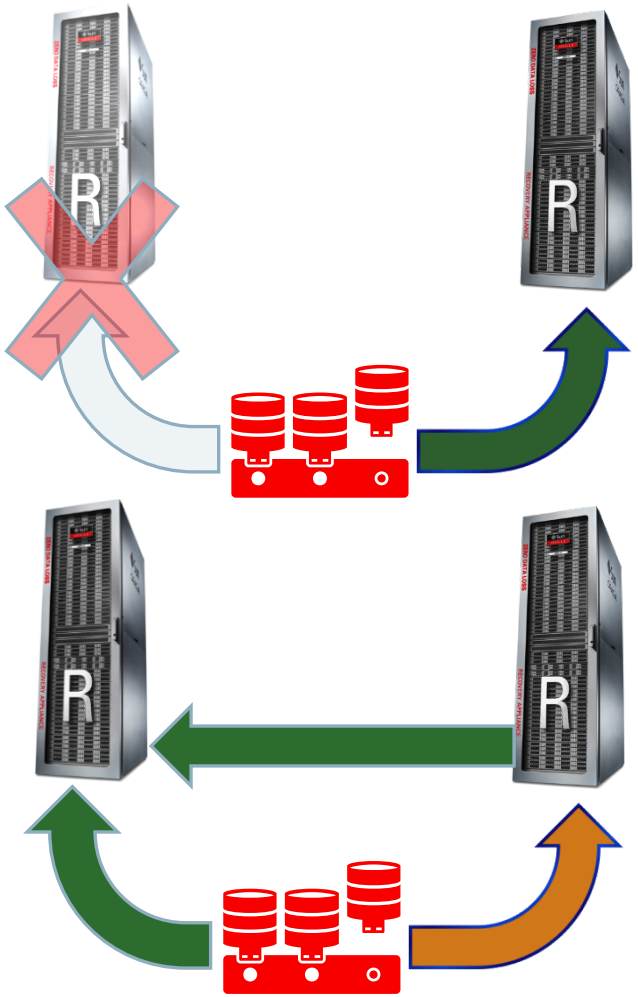


On the protected database:

1. The following RMAN command should be used for all RMAN backup operations

```
rman target / catalog /@dr_rahadr

The script will login to RAHADR1 if it is running, otherwise it will login to RAHADR2.

RMAN> run { execute script backup_database using 'Level1'; }

executing script: backup_database

allocated channel: rahadr1_sbt_1

channel rahadr1_sbt_1: SID=1936 instance=cdb122dr1 device type=SBT_TAPE

channel rahadr1_sbt_1: RA Library (RAHADR1) SID=6DEA2A958DFBE0CFE05311F3850AB3AB

allocated channel: rahadr1_sbt_2

channel rahadr1_sbt_2: SID=394 instance=cdb122dr1 device type=SBT_TAPE

channel rahadr1_sbt_2: RA Library (RAHADR1) SID=6DEA2A9CC2BBE0D0E05311F3850AC634

Notice that the channel allocations state if we are logging into RAHADR1 (rahadr1_sbt_1 & rahadr1_sbt_2) as well as specifying the RA database name (RAHADR1).
```

Operations (ZDLRA) – Backup Piece Gap Resolution



- In case of a failover, the backup process is transparent to the protected database.
- When the upstream (RAHADR1) becomes available again, the backups that had failed over to downstream (RAHADR2) need to be transferred back to RAHADR1 to resolve the gap of virtual full backups.
- **The gap to be resolved will show as INDEX_BACKUP tasks in ORDERING_WAIT state on RAHADR1**, as the virtual full backup metadata is present (via normal catalog reconcile with RAHADR2 configured as downstream) but are **not yet physically** present on the appliance.
- To do this, a PL/SQL script was created to look for INDEX_BACKUP tasks that are in an ORDERING_WAIT state and will then determine which backup pieces need to be transferred to RAHADR1 from RAHADR2. It will then transfer these pieces in parallel (if possible) back to RAHADR1 using the DBMS_RA.POPULATE_BACKUP_PIECE API.
- The script first loads the RA_POPULATE_BACKUP_PIECE procedure into the database, and then creates a DBMS_SCHEDULER_JOB that runs every 15 minutes to see if there are any pieces that need to be transferred. The initial query is very quick but if pieces are found, then the job will run for an extended period of time due to the INDEX_BACKUP tasks that are created on RAHADR1 as a result of the DBMS_RA.POPULATE_BACKUP_PIECE calls. The script to be used is tkrmrshadr.sql.

```
Installing tkrmrshadr.sql
- Install the script into all RAs participating in
HADR.
- The script only needs to be installed once:
    - As rasys:  sqlplus rasys/<rasyspwd>
@<full_dir_location_of_script>/tkrmrshadr.sql
```

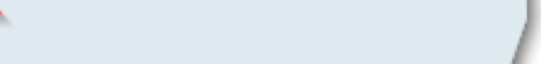
Operations (ZDLRA) – Backup Piece Gap Resolution (Cont.)

Note: For customers on versions earlier than ZDLRA 12.2.1.1.2-201810-RELEASE, multi-section backups are not supported by DBMS_RA.POPULATE_BACKUP_PIECE.

*For multi-section backups, the DB trace file with a pattern of *pdb_<timestamp>.trc will contain a list of BACKUP FROM SCN commands to be* ***executed on the protected database*** *(CDB122DR) to the upstream RA (RAHADR1) to resolve the backup gap.*

An Example of the *pbp_<timestamp>.trc file showing 6 different files that are multi-section backups is as follows:

```
*** MODULE NAME:(RA_POPULATE_BACKUP_PIECE##) 2018-09-19T22:36:51.521022-07:00
*** ACTION NAME:(Dumping_MultiSection_Backups) 2018-09-19T22:36:51.521038-07:00

==================
The following RMAN backup commands will resolve any ORDERING_WAIT tasks as a result of multi-section backups
==================
For Database IOPDB (db_key = 4989718)
backup incremental from scn 77361811 datafile 11;
backup incremental from scn 77358907 datafile 12;
backup incremental from scn 77358906 datafile 86;
backup incremental from scn 77362092 datafile 87;
backup incremental from scn 77362337 datafile 90;
backup incremental from scn 77362132 datafile 95;
==================
Populate_backup_piece : End of run at 19-SEP-18 10.36.51.540609000 PM -07:00
```



Note: For customers on ZDLRA Version 12.2.1.1.2-201810-RELEASE or later, the above restriction does not apply, i.e. POPULATE_BACKUP_PIECE supports multi-section backups. Ensure that the latest tkrmrshadr.sql script is downloaded from MOS Note 2432144.1 and executed on all Recovery Appliances.

Integrated Cloud

Applications & Platform Services


ORACLE®