ORACLE DATABASE 12c



Deploy Automated Site Failover, Automated Database Re-Instatement and Maximum Primary Instance Offloading using Oracle Active Data Guard

Powered by Oracle Data Guard Observer

@Customer Technical Hands-On Workshops
Developed by:
Fawzi. A. Alswaimil
Integration Solution Specialist
Saudi Arabia | Kuwait | Bahrain | Egypt
fawzi.alswaimil@oracle.com | +966 50 345 5094

Deploying Automated Managed Site Failover, Automated Database Re-Instatement and Maximum Primary Instance Offloading

Using Oracle Active Data Guard

Version 1.0
November, 2017

Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:
U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle America, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party contents, products and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

How is This Manual Organized?

The manual is divided into three sections:

1. Deploying Automated Managed Site Failover, Automated Database Re-Instatement and Maximum Primary Instance Offloading

2. APPENDIX A: Oracle Data Guard FAST-START Failover Workshop, Activity Guide

3. APPENDIX B: Bi-Directional Topology for Business Continuity

Deploying Automated Managed Site Failover, Automated Database Re-Instatement and Maximum Primary Instance Offloading

This is the session slide deck. It composed of five sections:

- Deploy Oracle Data Guard
- Enabling FAST-START Failover and Automatic Re-Instatement
- Enabling Active Data Guard
- Implementing Backup and Recovery for Oracle Active Data Guard
- Enabling Snapshot Standby Database
- Monitoring Active Data Guard

APPENDIX A: Oracle Data Guard FAST-START Failover Workshop, Activity Guide

The activity guide is hands-on guide to complete the following:

- Deploy Oracle Data Guard
- Enabling FAST-START Failover and Automatic Re-Instatement
- Enabling Active Data Guard
- Implementing Backup and Recovery for Oracle Active Data Guard
- Enabling Snapshot Standby Database
- Monitoring Active Data Guard

The following diagram shows the sequence of performing the workshop. The sequence is strict and must be executed as shown in Figure 1.

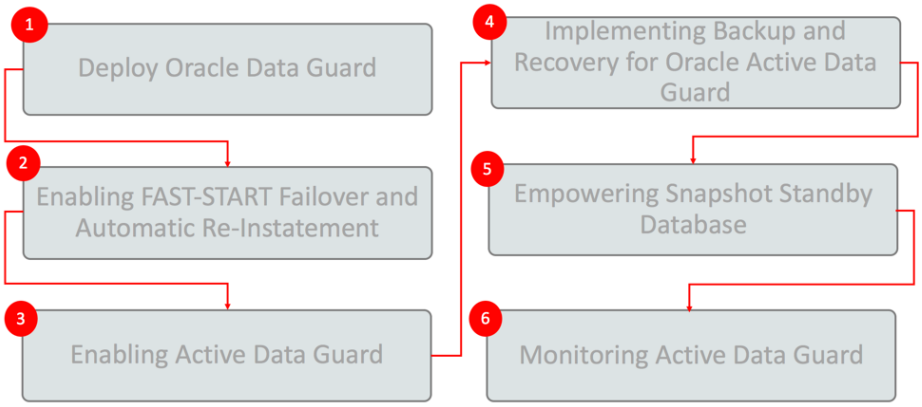


Figure 1: Workshop Sequencing

APPENDIX B: Bi-Directional Topology for Business Continuity

This section explains the deployment of Oracle GoldenGate to deliver a Disaster Recovery Solution on Oracle Database Standard Edition.

When using Oracle GoldenGate to implement a disaster recovery (DR) site, it is developed as active-passive, bi-directional topology architecture. The source database is designated as the primary system and the target database is designated as the live-standby system. The live-standby system database remains on read-write mode. Applications users are connected to either the primary or the live-standby system, but not both. The live-standby system remains ready for the failover and switchover situations.

Deploying Automated Managed Site Failover, Automated Database Re-Instatement and Maximum Primary Instance Offloading

Using Oracle Active Data Guard

Fawzi A Alswaimil
Integration Solution Specialist
eMail: Fawzi.alswaimil@oracle.com
Cell: +966 50 345-5094

Date: 30-Nov-2017
Location: Oracle Khobar
Time: 9am-4pm GMT+3
https://myoracle.webex.com/meet/fawzi.alswaimil
Intercall Code: 1399662
Call Password: 1234

@Customer Technical Hands-On Workshop Program v1.0

ORACLE

Copyright © 2015 Oracle and/or its affiliates. All rights reserved.



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.

ORACLE

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Fawzi Alswaimil

Integration Cloud Solution Team

Three decades of Oracle technology experience

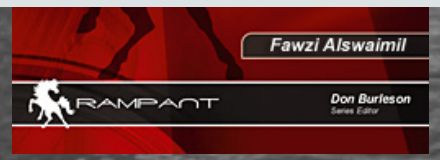
Develop, Position, Architect and deploy GG Solutions

Database PL/SQL API, Perl and Oracle for .NET Developer

Worked with Oracle 5, 6, 7, 8i, 9i, 10g, 11g, 12c

An Author, trainer and Oracle workshops developer







Solution Specialist V

Email: Fawzi.alswaimil@oracle.com

Mobile: +966 50 345 5094

Agenda

	Session	Timing
Day 1	An Introduction	09:00am – 12:00 noon
	Architecture in-depth	
	Data Guard Broker	
	Protection modes and role Transition	
	Deploy Auto-Managed Standby using RMAN	
	Flashback database for Fast-Start Failover	
	Backup and Recovery on Data Guard Environment	
	Monitoring Oracle Data Guard Gap	
	Demonstration	
	Data Guard deployment technique	
	Lunch Break	12:00pm - 1:00pm
	Hands-On Lab	1:00pm – 4:00pm

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Workshop Package

Slide Deck



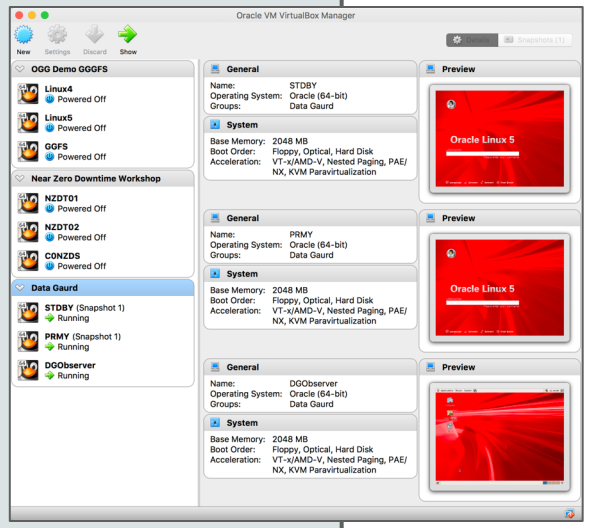
Activity Guide

APPENDIX A

Oracle GoldenGate

APPENDIX B

3 Linux Virtual Machines



Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Oracle Software Download Terms and Conditions

Visit Oracle Delivery Software Cloud

edelivery.oracle.com

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

First-Class 24/7 Building Blocks

Real Application Cluster

RAC

Delivers unlimited instance high-availability.
Transparent Application Fail-Over (TAF) support
Speed-Up and Scale-Up Database Application
Cache Fusion across instances (Up to 100 active instances)

High Availability

Automatic Storage Management

ASM

Vertical Volume Manager
Designed for Real application Cluster
On-line files maintenance operations
Striping and mirroring on low commodity hardware

On-Line Management

Active Data Guard

ADG

A physical Standby Database residing remotely.
Block-to-Block Sync, Physically
Logical Standby Data Guard and snapshot standby database
Shipped with Oracle GoldenGate

Disaster Recovery

Oracle GoldenGate

OGG

Advanced Database Replication (DML and DDL).
Online Operational Business Intelligence
Business Continuity (Zero Downtime Upgrade and Migration)
Spin-off read-write from standby database

Business Continuity

Real Application Testing

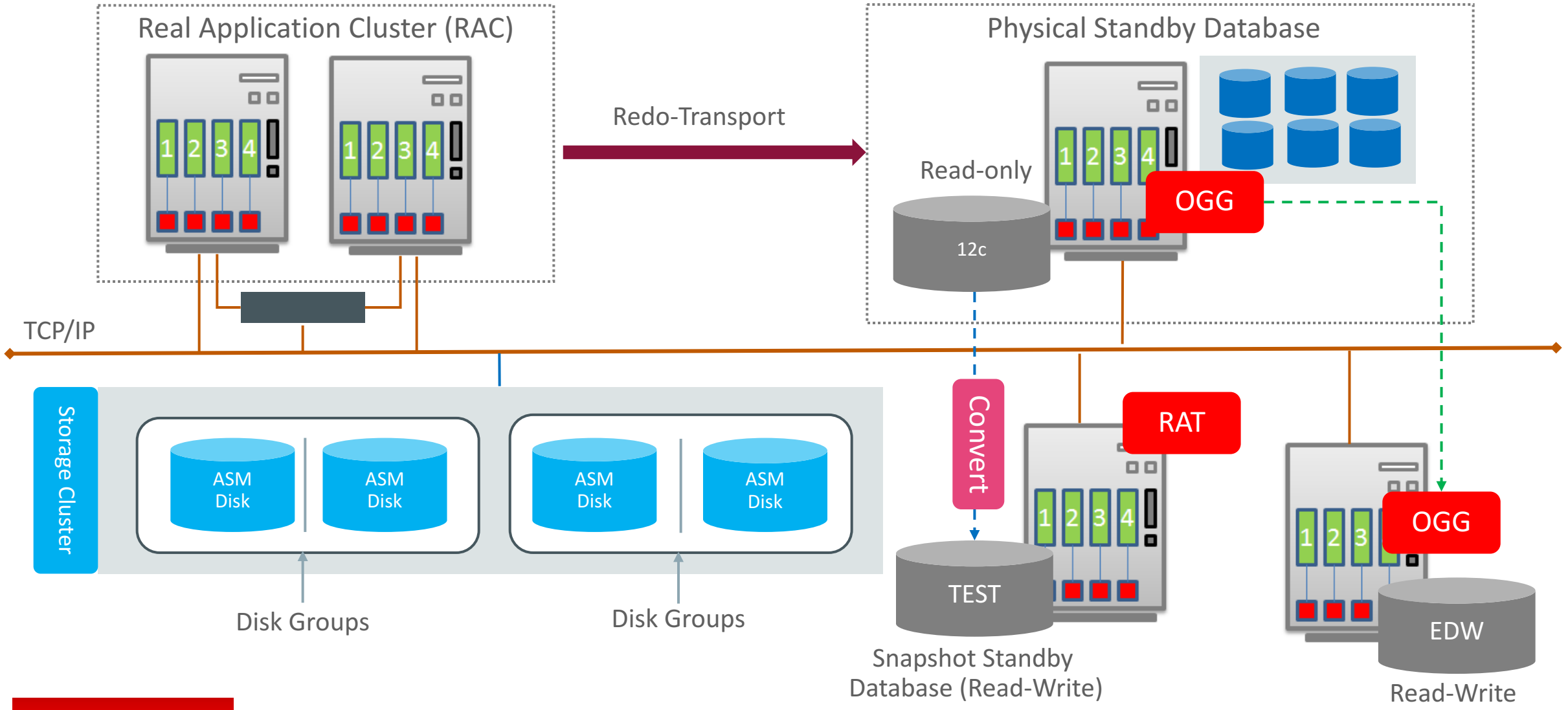
RAT

Change Management system
Hardware and Software Upgrade and maintenance
SQL Performance Analyzer (SPA)
Database Replay on a snapshot standby database

Database Change Management

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Connecting the Building Blocks



Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Data Guard Functionalities

- Continuous service during disasters or crippling data failures
- Complete data protection against corruption and data loss
- Elimination of idle standby systems
- Flexible configuration of your system to meet requirements for business protection and recovery
- Centralized Management through Data Guard Broker
- Automated Managed-Failover through an observer

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Ease of Competition with Storage Mirroring

Function	Active Data Guard	Storage Mirror
Synchronization	Data Guard synchronizes data from the SGA (Oracle memory), configurable on SYNC or ASYNC with compression option.	Synchronizes based on file-to-file data changes. With database composed of hundreds of files, the method is not suitable for Database.
Apps Awareness	Data Guard is application aware. Supporting the database isolation level	Totally not aware of the application transaction and the database isolation level
Network	Run on shared low bandwidth network such as 4MBPS	Requires high-bandwidth to support file synchronization
Proximity	Unlimited	Limited
Cost	With proper sizing of the standby database system, it will more cost effective	Same hardware on the remote site
Skills Set	DBA is skill set is enough to manage Data Guard	SLA with hardware vendor likely required
Add value	Maintenance and reporting instance with Active Data Guard (Active-Active)	For Disaster Recovery (Active-Passive)
Failover	Automatic based on manageable policies handed by an Observer Server	Basic failover

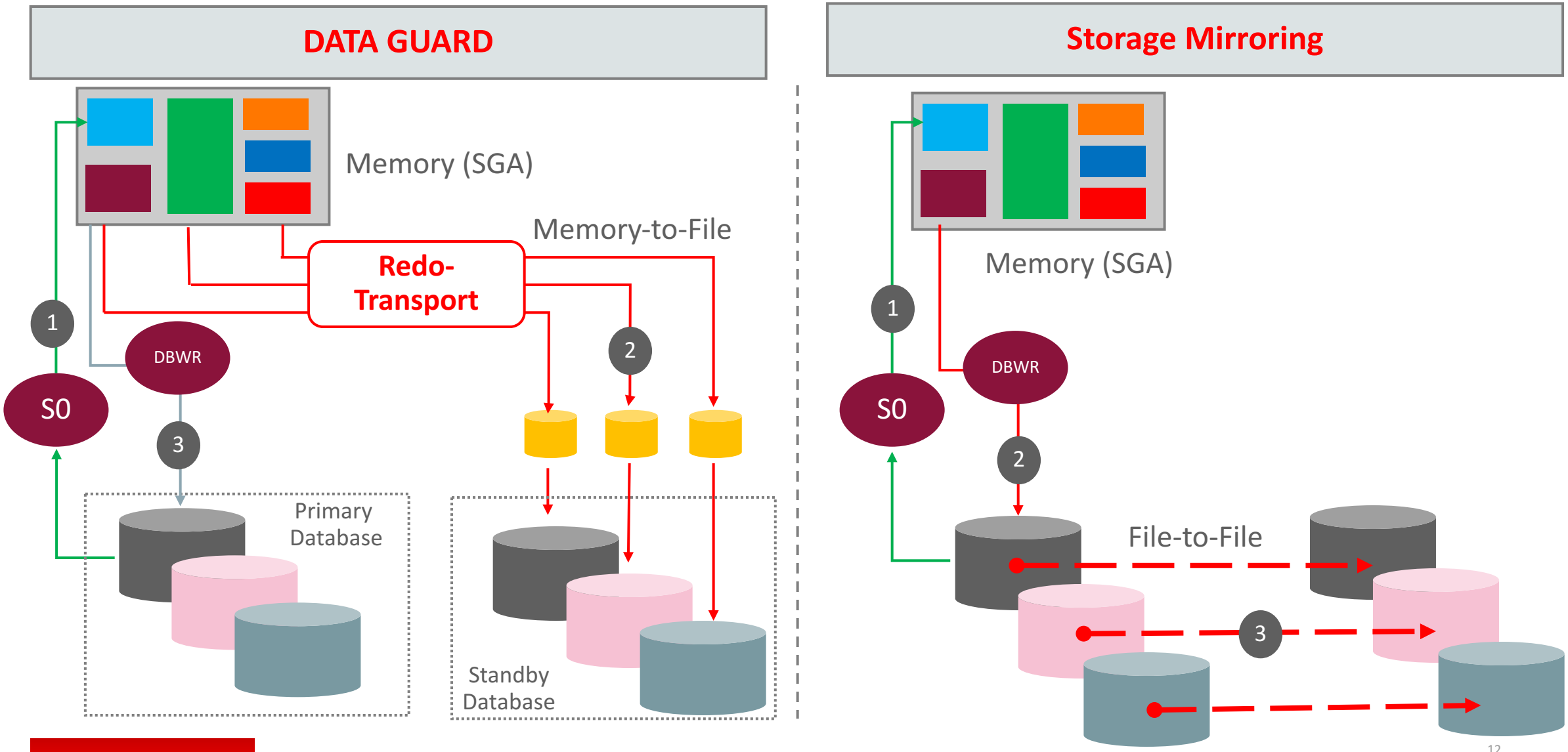
Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Competing with Storage Mirroring

- Synch from Memory and Auto failover using thresholds
- Active Data Guard is Database commit/rollback aware
- Communication bandwidth is minimal for Active Data Guard
- Lower license cost compared to hardware cost through sizing
- DBA skills is sufficient to deploy Disaster Recovery using DG
- Maintenance, Data Protection and Reporting Database

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

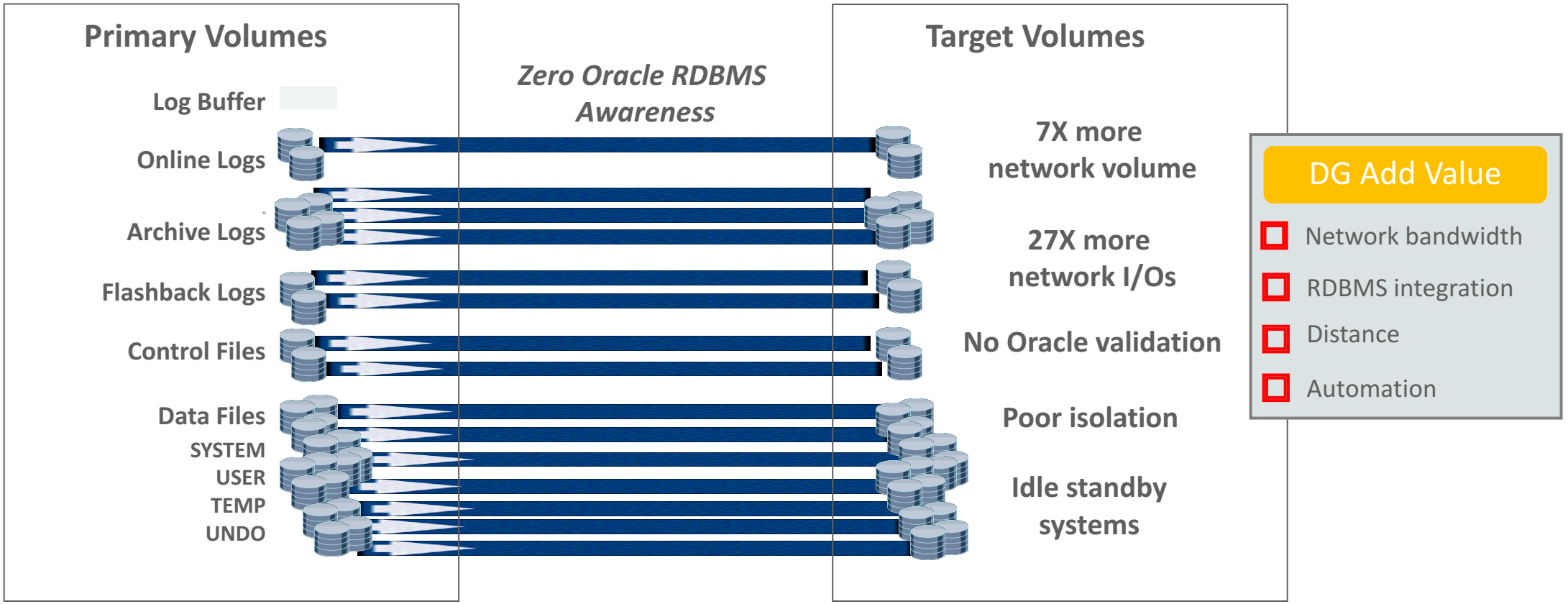
Sync Architecture



ORACLE

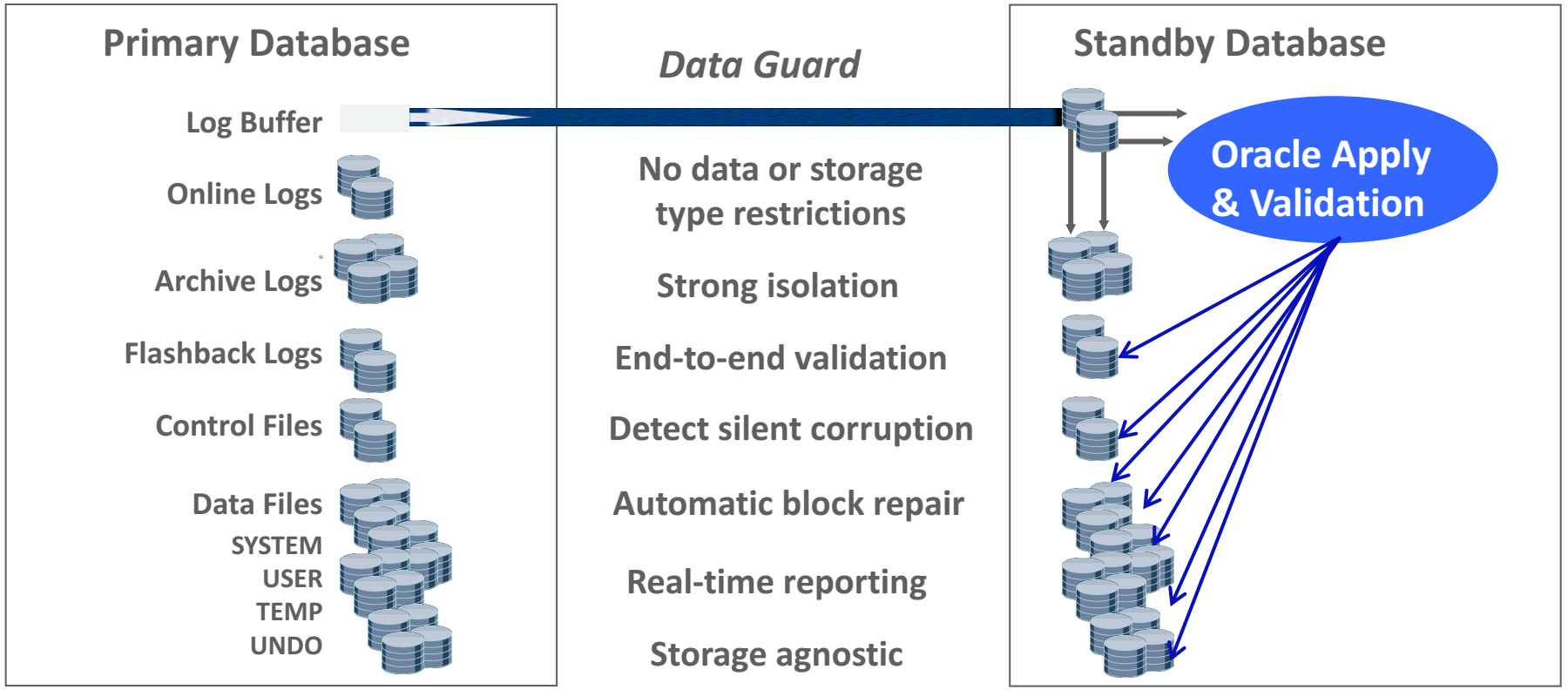
Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Disk Mirroring is **NOT** scalable for Oracle Database VLDB



Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Oracle Data Guard Delivers **HIGH** Scalability for VLDB



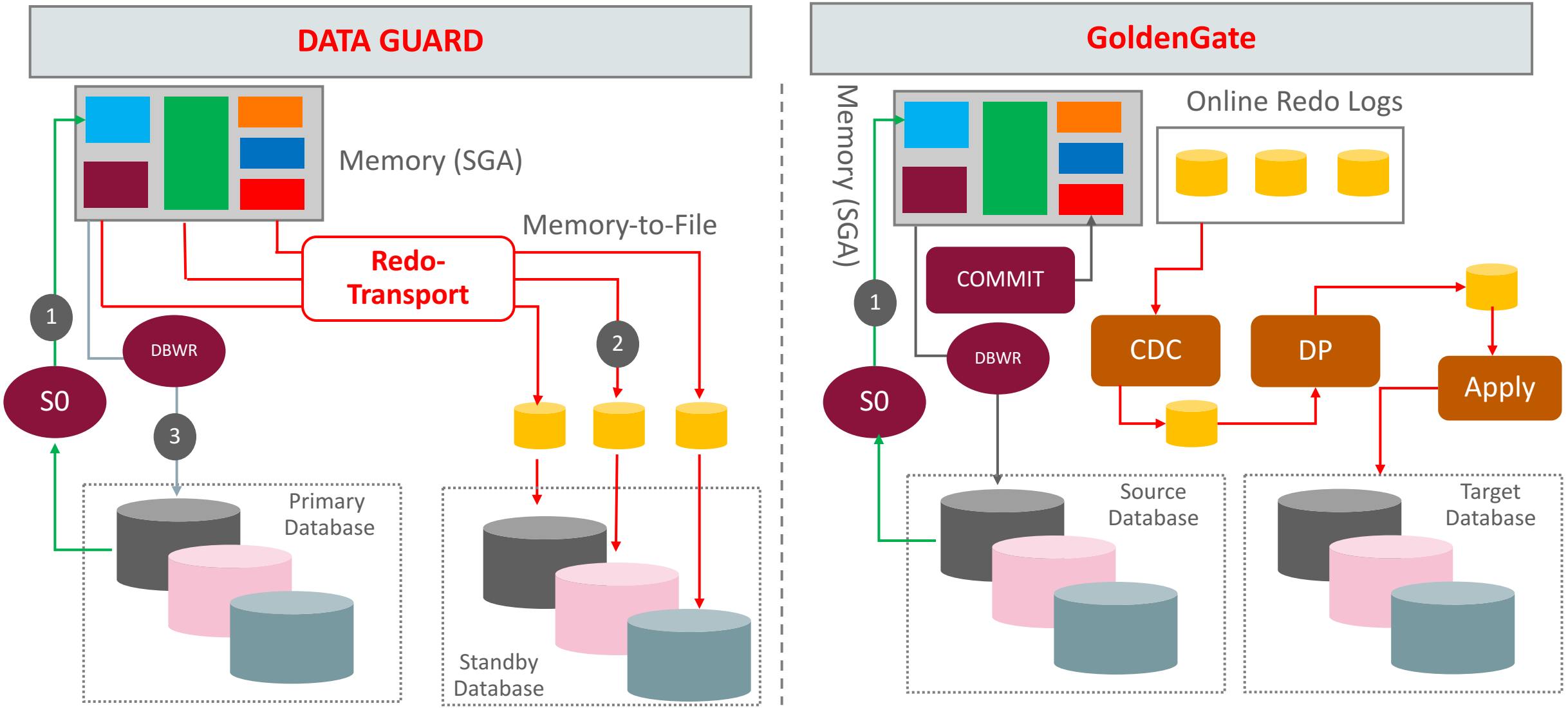
Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Data Guard vs Storage Mirror for Disaster Recovery

Function	Active Data Guard	Storage Mirror
Failover	Automatic	Manual
Reinstate	Automatable	Manual
Synchronization	Oracle Block-to-Block	OS Block-to-Block
Standby Database Mode	Read-only	Read-only
Application	All	All
Database	Oracle Database EE	All
Heterogeneous	No	No
Network Bandwidth	Mid	High
Distance	Unlimited	limited
Gap Resolution	Automatic	Automatic
Granularity	No	No
Offloading	Report and Administration	Not Active

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Sync Architecture



Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Data Guard vs Oracle GoldenGate for Disaster Recovery

Function	Active Data Guard	GoldenGate
Failover	Automatic	Manual or event-based
Reinstate	Automatable	Manual
Synchronization	Block-to-Block	Native SQL
Standby Database Mode	Read-only	Read-write
Application	All	Inter-Operability
Database	Oracle Database EE	Oracle Database EE and SE
Heterogeneous	No	Yes
Network Bandwidth	Mid	Low
Distance	Unlimited	Unlimited
Gap Resolution	Automatic	Automatic
Granularity	No	Yes
Offloading	Report and Administration	Report

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Oracle Data Guard Components

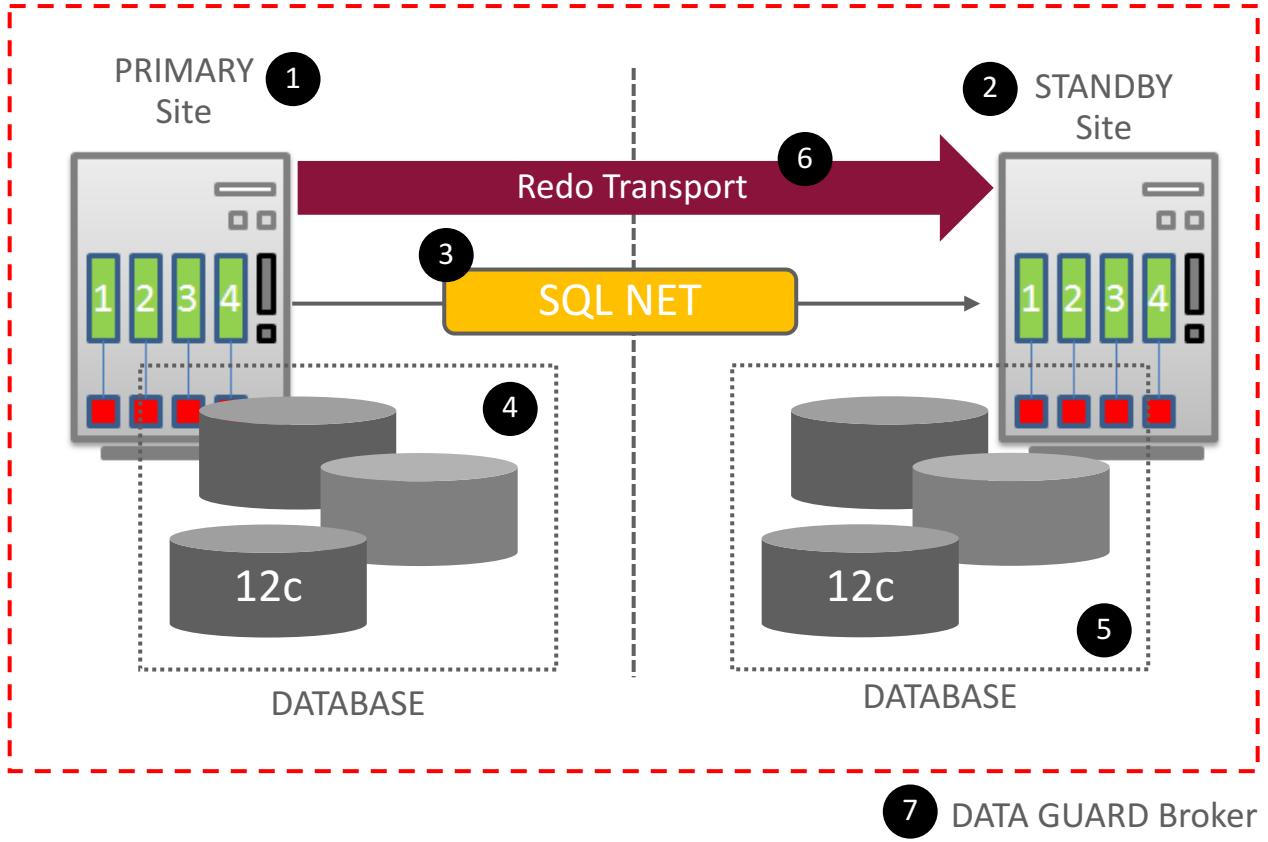
Applications

ORACLE E-BUSINESS SUITE

SAP

ORACLE JD EDWARDS

ORACLE PEOPLESOFT



Administration

- RMAN Backup
- Queries offloading
- Snapshot for Testing
- Database Replay
- Auto-block Repair

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Standby Database Types

Physical Standby Database

- Is identical to the primary database on a block-to-block basis
- Is synchronized with the primary database through application of redo data receive from the primary database
- Can be used concurrently for data protection and reporting

Logical Standby Database

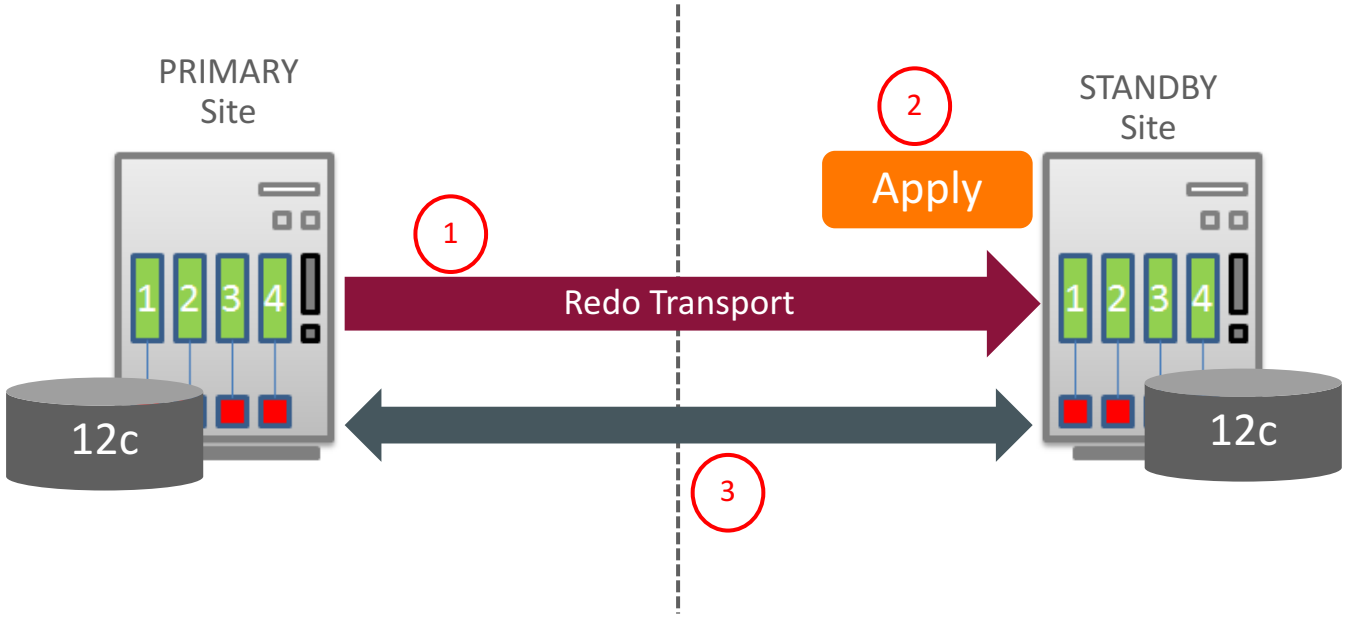
- Shares the schema definition
- Is kept synchronized with the primary database by transforming the data in the redo received from the primary database into SQL statements and then executing the SQL statements
- Can be used concurrently for data protection, reporting, and upgrades

Snapshot Standby Database

- Is a fully updatable standby database
- Is created by converting a physical standby database
- Can be used for updates, but those updates are discarded before the snapshot standby database is converted back into a physical standby database
- Can be used for testing

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Data Guard Services



I. Redo transport services

II. Apply services: Redo Apply, SQL Apply

III. Role management services

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Role Transition

Switchover

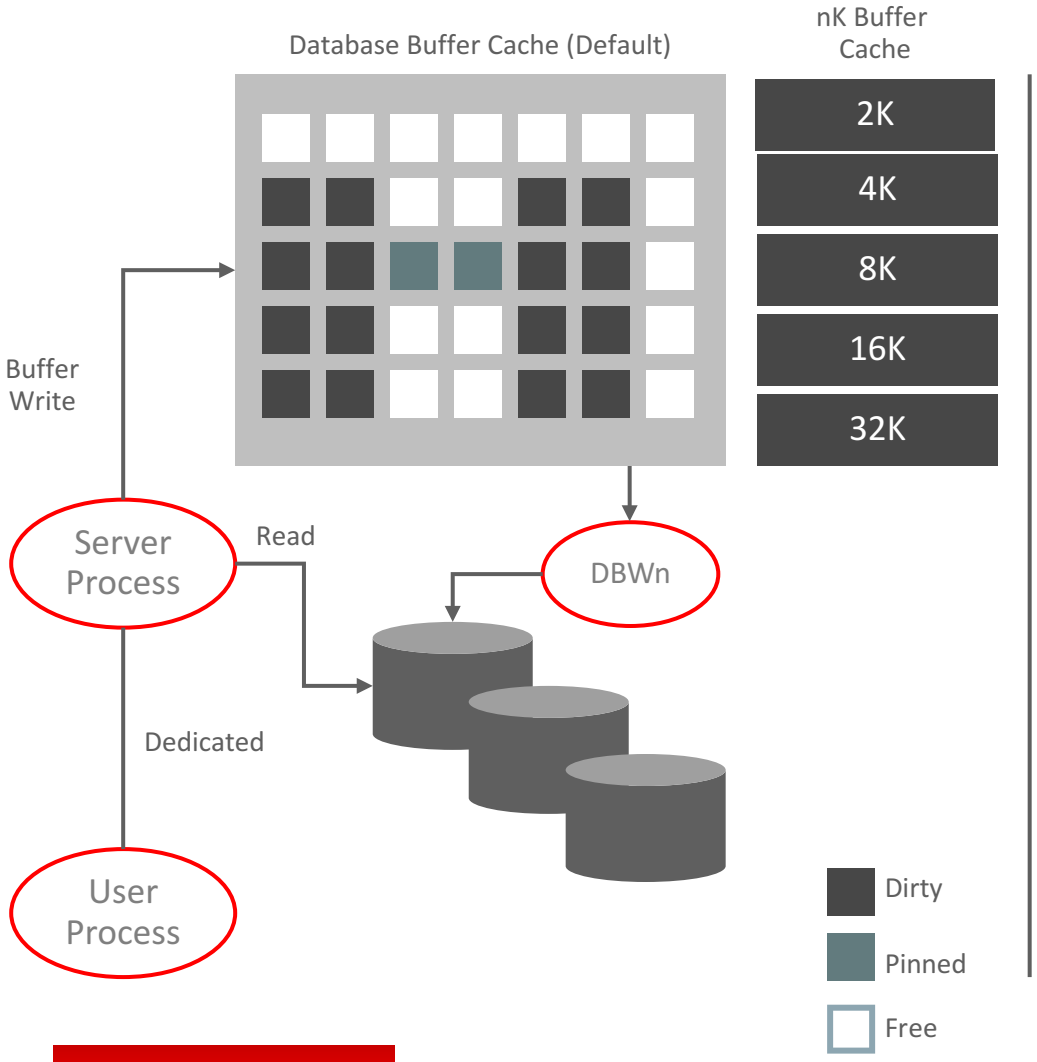
- Planned role reversal
- Used for OS or hardware maintenance

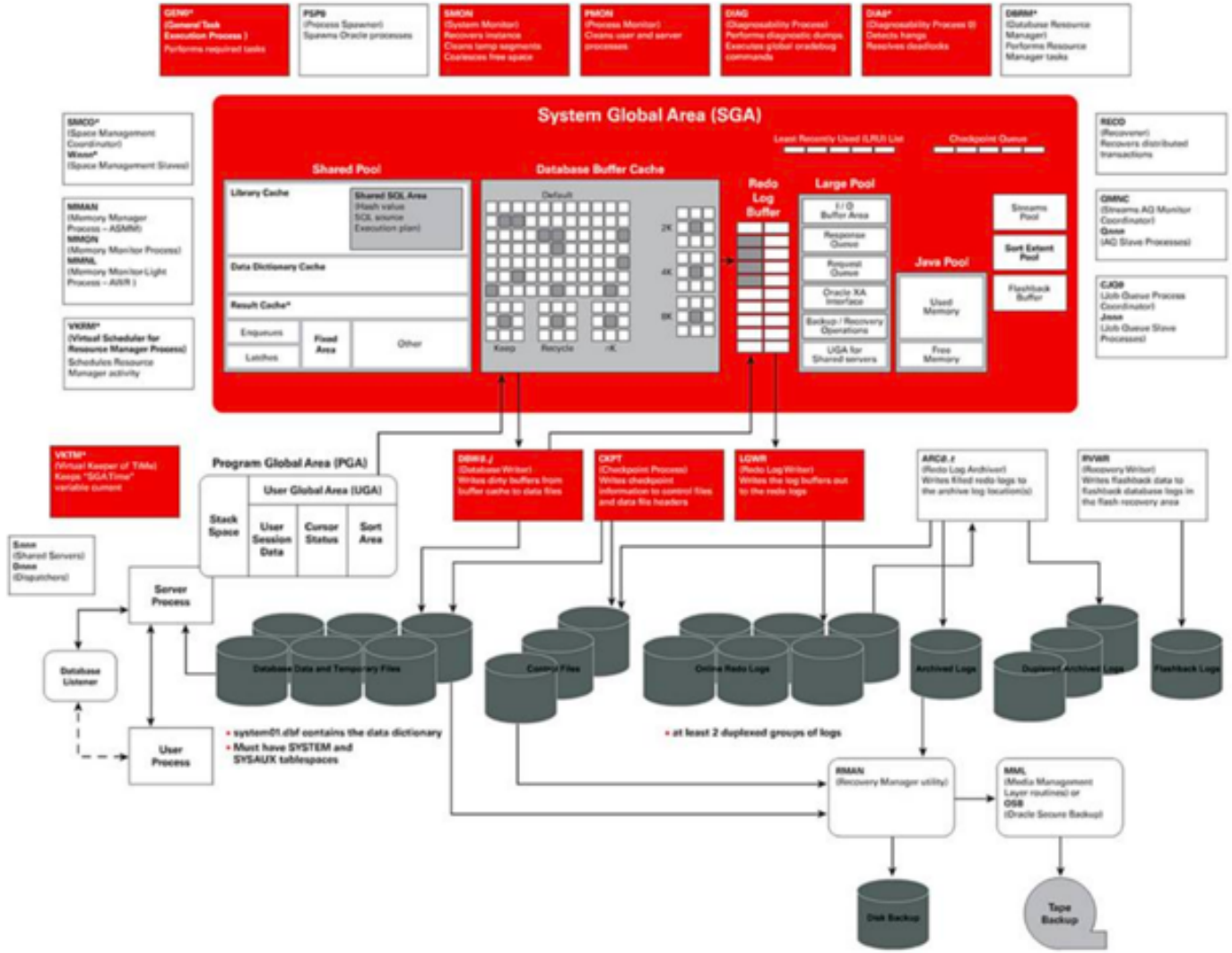
Failover

- Unplanned role reversal
- Emergency use
- Zero or minimal data loss (depending on choice of data-protection mode)
- Can be initiated automatically when fast-start failover is enabled

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

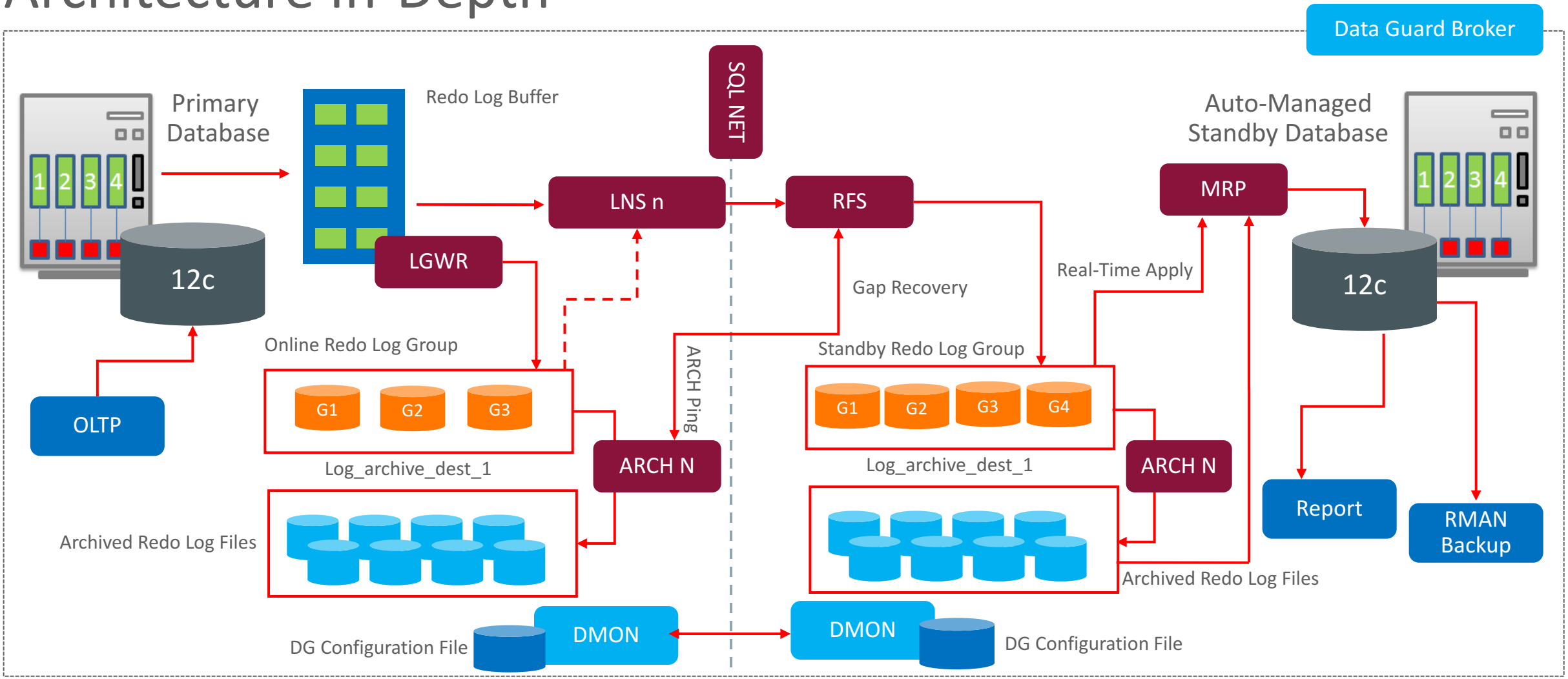
An Overview: Database Architecture





Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

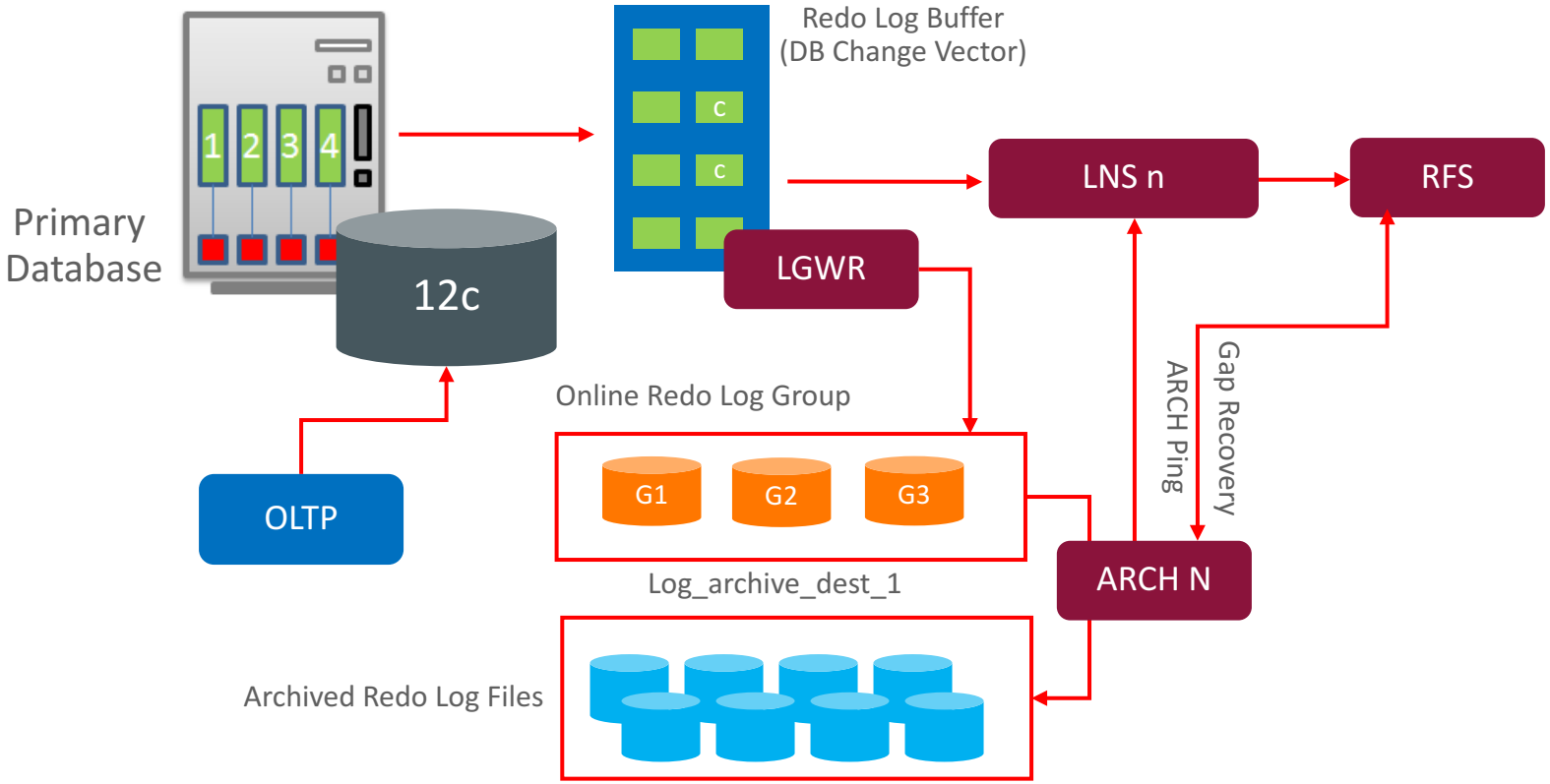
Architecture In-Depth



Data Guard Observer Server | DGMGRL> Interface | EMCC Console

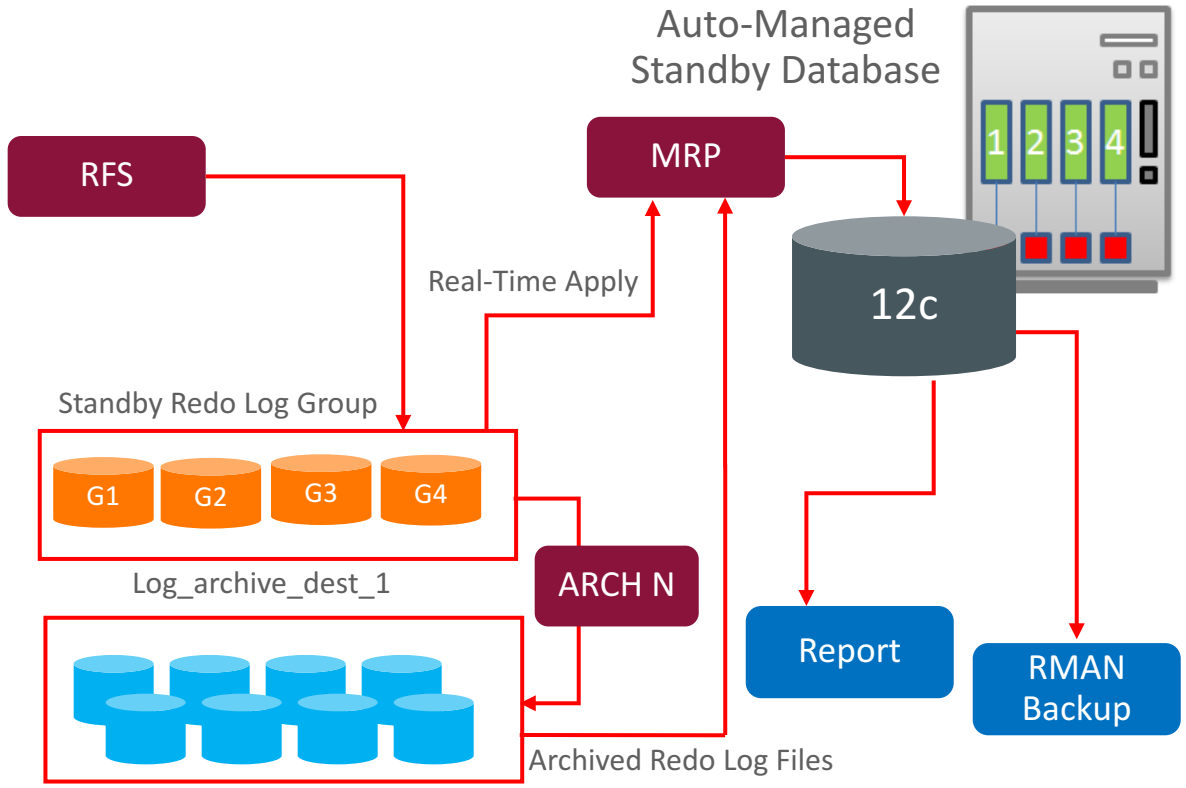
Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

On the Primary System



Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

On the Standby System



Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Protection Modes

Maximum Protection

- This protection mode guarantees that no data loss occurs if the primary database fails
- The redo data that is needed to recover each transaction must be written to both the local online redo log and the standby redo log
- Database shutdown in case of transport services failure

Maximum Availability

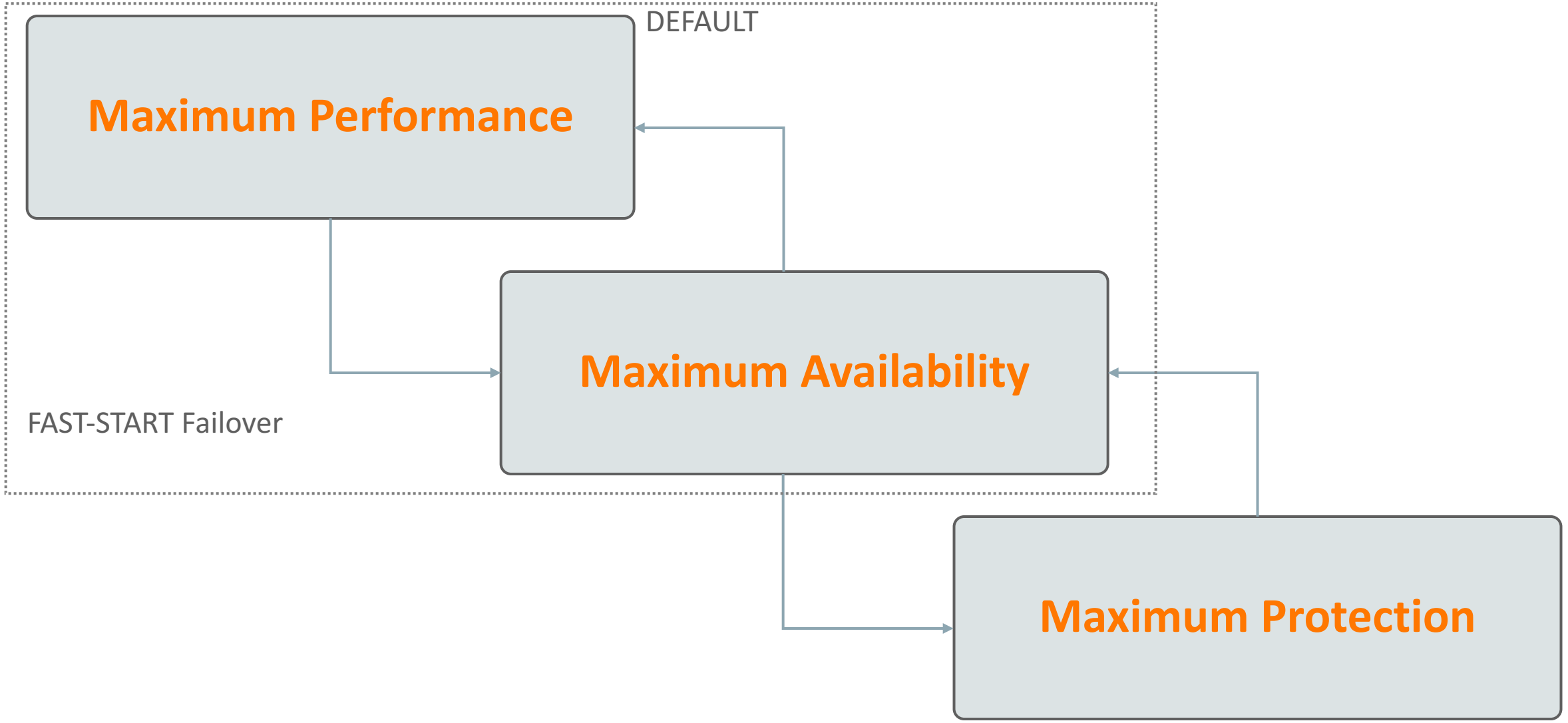
- This protection mode provides the highest possible level of data protection without compromising the availability of the primary database
- A transaction does not commit until the redo needed to recover that transaction is written to the local online redo log and to at least one remote standby redo log

Maximum Performance

- The default protection mode provides the highest possible level of data protection without affecting the performance of the primary database
- Allows commit as soon as the redo data needed to recover that transaction is written to the local online redo log

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Managing Protection Modes



Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Resetting Protection Modes

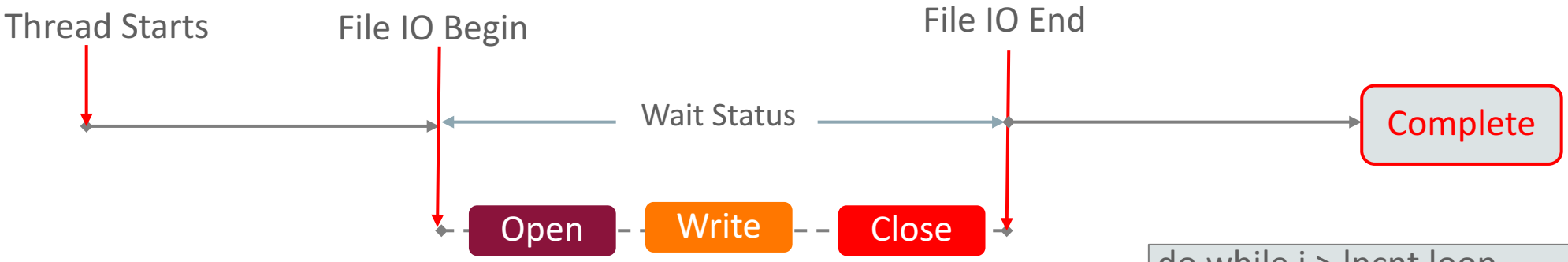
```
DGMGRL> EDIT DATABASE 'DHDB'
> set property LogXptMode='ASYNC';
Property "logxptmode" updated
DGMGRL> EDIT DATABASE 'DXDB'
> set property LogXptMode='ASYNC';
Property "logxptmode" updated
DGMGRL> EDIT CONFIGURATION
> SET PROTECTION MODE AS MaxPerformance;
Succeeded.
DGMGRL>
```

Protection Mode	LogXptMode	Acknowledgement Mode
Maximum Protection	SYNC	AFFIRM
Maximum Availability	SYNC	AFFIRM
Maximum Performance	ASYNC	NOAFFIRM

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

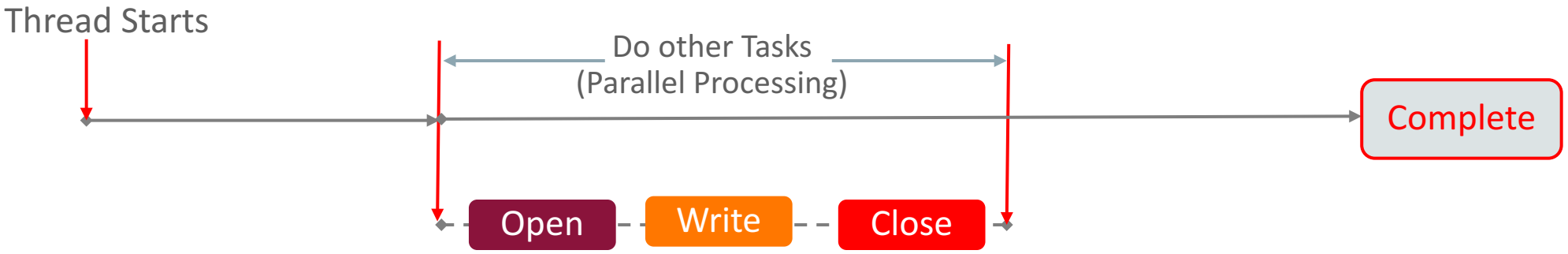
SYNC & ASYNC Technology Illustrated

SYNC IO




```
do while i > lncnt loop
    send_html_page(i)
    DoEvent
end loop
```

ASYNC IO



Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Comparing Protection Modes

Mode	Risk of Data Loss	Transport	No Acknowledgment means?
Maximum Protection	Zero data loss Double failure protection	SYNC	Stall or suspend the primary database until an acknowledgement is received from the standby system
Maximum Availability	Zero data loss	SYNC	Stall or suspend the primary database until an internal threshold period expired, then the primary database resume processing.
Maximum Performance (Default Setting)	Potential for minimal data loss	ASYNC	The primary database never wait for the acknowledgement from the standby database.

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Controlling Redo-Transport Services

Parameter	Description
LOG_ARCHIVE_CONFIG	Lists the unique database name for each database in the data guard configuration.
LOG_ARCHIVE_DEST_n	Specify the archive log destination for controlling the data guard redo-transport service
LOG_ARCHIVE_DEST_STATE_n	Enable or Disable the specific destination
ARCHIVE_LAG_TARGET	Forces log switch after an elapsed timeframe
LOG_ARCHIVE_TRACE	Control the output generated level

```
SQL> alter system set log_archive_dest_2='service=DXDB
valid_for=(online_logfile, primary_role) db_unique_name=DXDB' scope=both;

System altered.

SQL>
```

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

LOG_ARCHIVE_LOG_n Attributes, where n=1..31

Attribute	Description
SERVICE	Specifies a valid Oracle Net service name (SERVICE=net_service_name) that identifies the remote Oracle database instance to which redo data will be sent. Each destination must specify either the LOCATION or the SERVICE attribute.
REOPEN	Specifies the minimum number of seconds before the archiver processes (ARCn) or the log writer process (LGWR) should try again to access a previously failed destination. The default is 300 seconds.
DELAY	Specifies a minimum time lag between when redo data is archived on a standby site retrieving redo from a primary and when the archived redo log file is applied to the standby database or any standbys cascading from it. If you specify the DELAY attribute without a time interval, the default is 30 minutes.
NOREGISTER	Indicates that the location of the archived redo log file is not to be recorded at the corresponding destination.
MAX_FAILURE	Controls the consecutive number of times redo transport services attempt to reestablish communication and transmit redo data to a failed destination before the primary database gives up on the destination.

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

LOG_ARCHIVE_LOG_n Attributes, where n=1..31

Attribute	Description
SYNC | ASYNC	ASYNC, the redo data generated by a transaction need not have been received at a destination which has this attribute before that transaction can commit. SYNC, The redo data generated by a transaction must have been received by every enabled destination which has this attribute before that transaction can commit.
AFFIRM | NOAFFIRM	Control whether a redo transport destination acknowledges received redo data before or after writing it to the standby redo log.
NET_TIMEOUT	Specifies the number of seconds the log writer process on the primary system waits for status from the SYNC (NSSn) process before terminating the network connection. The default is 30 seconds.
VALID_FOR	Identifies when redo transport services can transmit redo data to destinations
DB_UNIQUE_NAME	Specifies a unique name for the database at this destination
MAX_CONNECTIONS	Specifies the maximum number of network connections that can be used to transmit redo data to this destination
COMPRESSION	Indicates whether network compression is enabled or disabled

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

VALID_FOR Attribute

VALID_FOR Attribute Combination	Primary	Physical	Logical
ONLINE_LOGFILE, PRIMARY_ROLE	Valid	Ignored	Ignored
ONLINE_LOGFILE, STANDBY_ROLE	Ignored	Ignored	Valid
ONLINE_LOGFILE, ALL_ROLES	Valid	Ignored	Valid
STANDBY_LOGFILE,STANDBY_ROLE	Ignored	Valid	Valid
STANDBY_LOGFILE, ALL_ROLES	Ignored	Valid	Valid
ALL_LOGFILES, PRIMARY_ROLE	Valid	Ignored	Ignored
ALL_LOGFILES, STANDBY_ROLE	Ignored	Valid	Valid
ALL_LOGFILES, ALL_ROLES	Valid	Valid	Valid

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Database Role Transition

Switchover

- Planned role transition
- Used for operating system and hardware maintenance
- No data loss
- Manually invoked

Failover

- Unplanned role transition
- Used in an emergency
- Minimal or no data loss
- Initiated at the standby System
- Fast-Start failover enables automatic failover

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Prior to Switchover DGMGRL Command

1. Verifies the states of primary and target standby database
2. Shuts down any instances as necessary
3. Switches roles between the primary and standby database
4. Update the broker configuration file with current role details
5. Restarts the new standby database
6. Open the new primary database in read/write
7. Start the redo transport services

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Verify Database Current Role

Primary

```
SQL> select DATABASE_ROLE, open_mode, protection_mode
  2  from   v$database;

DATABASE_ROLE       OPEN_MODE            PROTECTION_MODE
----------------    -------------------  --------------------
PRIMARY             READ WRITE           MAXIMUM PERFORMANCE

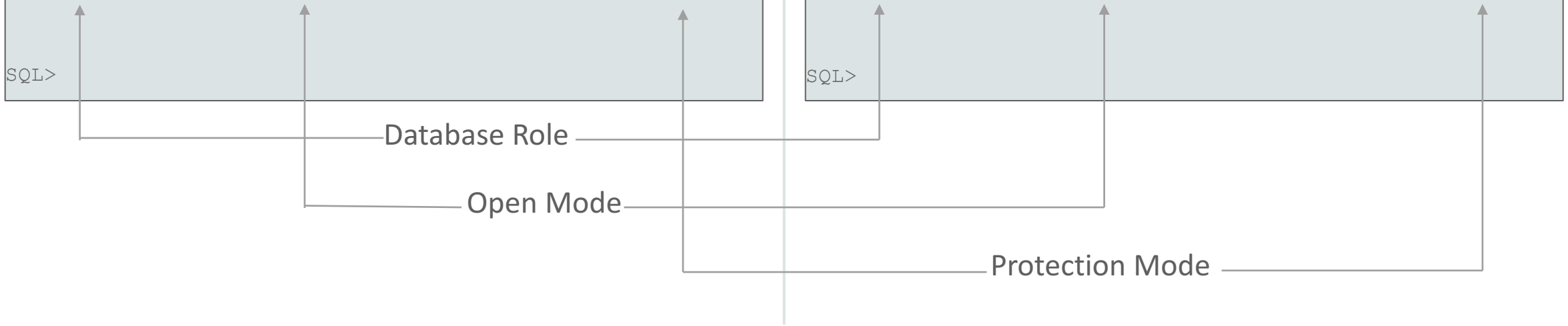
SQL>
```

Standby

```
SQL> select DATABASE_ROLE, open_mode, protection_mode
  2  from   v$database;

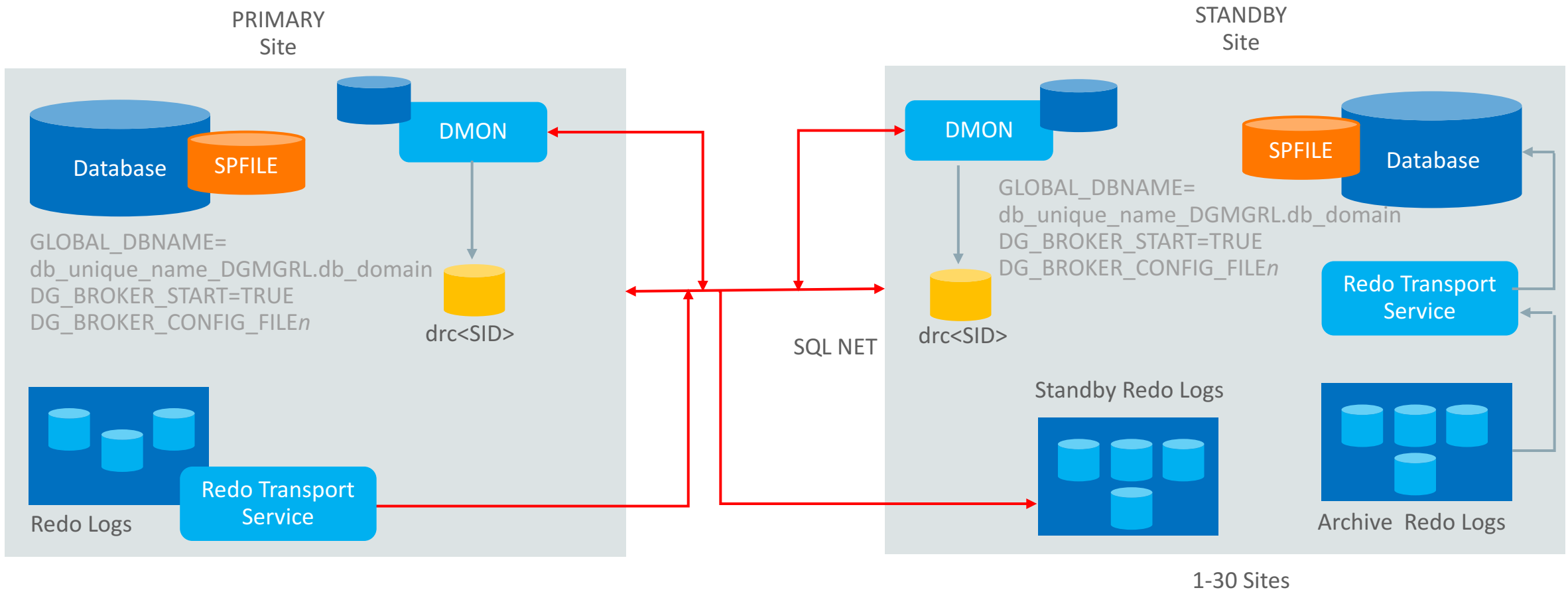
DATABASE_ROLE     OPEN_MODE              PROTECTION_MODE
----------------- --------------------   ------------------
PHYSICAL STANDBY READ ONLY WITH APPLY    MAXIMUM PERFORMANCE

SQL>
```



Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Data Guard Broker Configuration



Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

DGMGRL Interface

ADD	Adds a standby database to the Data Guard Broker configuration
CONNECT	Connect to a specified instance
CREATE	Create a data guard broker configuration
DISABLE	Disable broker configuration so that management can be performed outside of the broker
EDIT	Edit the properties for a configuration, database or configured instance
ENABLE	Enable broker configuration so that management is performed using the data guard broker
FAILOVER	Perform a failover operation from the standby using DGMGRL command line
REINSTATE	Change the status of disabled database into a valid standby database
REMOVE	Remove the database and its configuration from the data guard broker configuration
SHOW	Display the database status, summary and detailed
STARTUP/SHUTDOWN	Startup and shutdown the database using DGMGRL>
START/STOP	Start and stop the Observer
SWITCHOVER	Perform switchover operation, changing the role of the databases
EXIT/QUIT	Exit the data guard broker command line (DGMGRL)

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Starting DGMGRL

```
[oracle@prmy01 admin]$ . oraenv
ORACLE_SID = [DHDB] ?
The Oracle base remains unchanged with value /u01/app/oracle
[oracle@prmy01 admin]$ dgmgrl
DGMGRL for Linux: Version 12.1.0.1.0 - 64bit Production


Copyright (c) 2000, 2012, Oracle. All rights reserved.


Welcome to DGMGRL, type "help" for information.
DGMGRL> connect sys/oracle
Connected as SYSDG.
DGMGRL>
```



12*c*

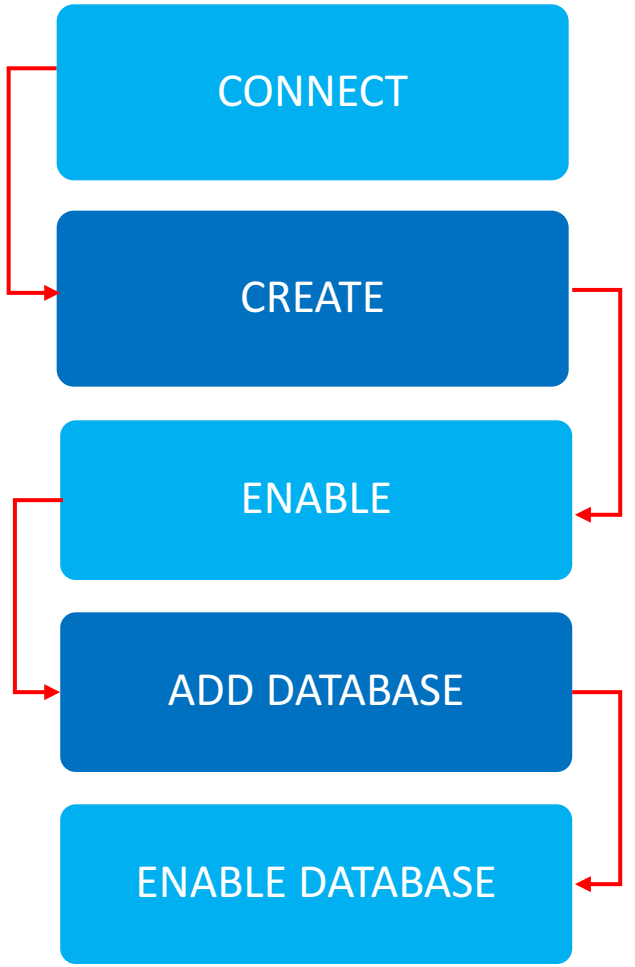
Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Creating Data Guard Broker Configuration

```
DGMGRL> connect sys/oracle
Connected as SYSDG.

DGMGRL> create configuration 'DG_CONF' as primary database
is 'DHDB' connect identifier is 'DHDB';
Configuration "DG_CONF" created with primary database "DHDB"

DGMGRL> enable configuration;
Enabled.
DGMGRL> add database 'DXDB' as connect identifier is 'DXDB';
Database "DXDB" added
DGMGRL> enable database 'DHDB';
Enabled.
DGMGRL> enable database 'DXDB';
Enabled.
DGMGRL>
```



Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Display Configuration Details

```
DGMGRL> show database verbose 'DHDB';

Database - DHDB

  Role:               PRIMARY
  Intended State:     TRANSPORT-ON
  Instance(s):
    DHDB


  Properties:
    DGConnectIdentifier             = 'DHDB'
    ObserverConnectIdentifier       = ''
    LogXptMode                      = 'ASYNC'
    RedoRoutes                      = ''
    DelayMins                       = '0'
    Binding                         = 'optional'
    MaxFailure                      = '0'
```

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Edit Database Properties

```
DGMGRL> EDIT CONFIGURATION
> SET PROPERTY FastStartFailoverThreshold = 45;
Property "faststartfailoverthreshold" updated
DGMGRL>
```

```
DGMGRL> edit configuration set property
> FastStartFailoverLagLimit = 45;
Property "faststartfailoverlaglimit" updated
DGMGRL>
```

```
DGMGRL> edit configuration set property
> FastStartFailoverPmyShutdown = false;
Property "faststartfailoverpmyshutdown" updated
DGMGRL>
```

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Display Data Guard Summary

```
DGMGRL> connect sys/oracle
Connected as SYSDG.
DGMGRL>
DGMGRL> show configuration

Configuration - DG_CONF

  Protection Mode: MaxPerformance
   Databases:
   DHDB - Primary database
     DXDB - Physical standby database


Fast-Start Failover: DISABLED

Configuration Status:
SUCCESS

DGMGRL>
```

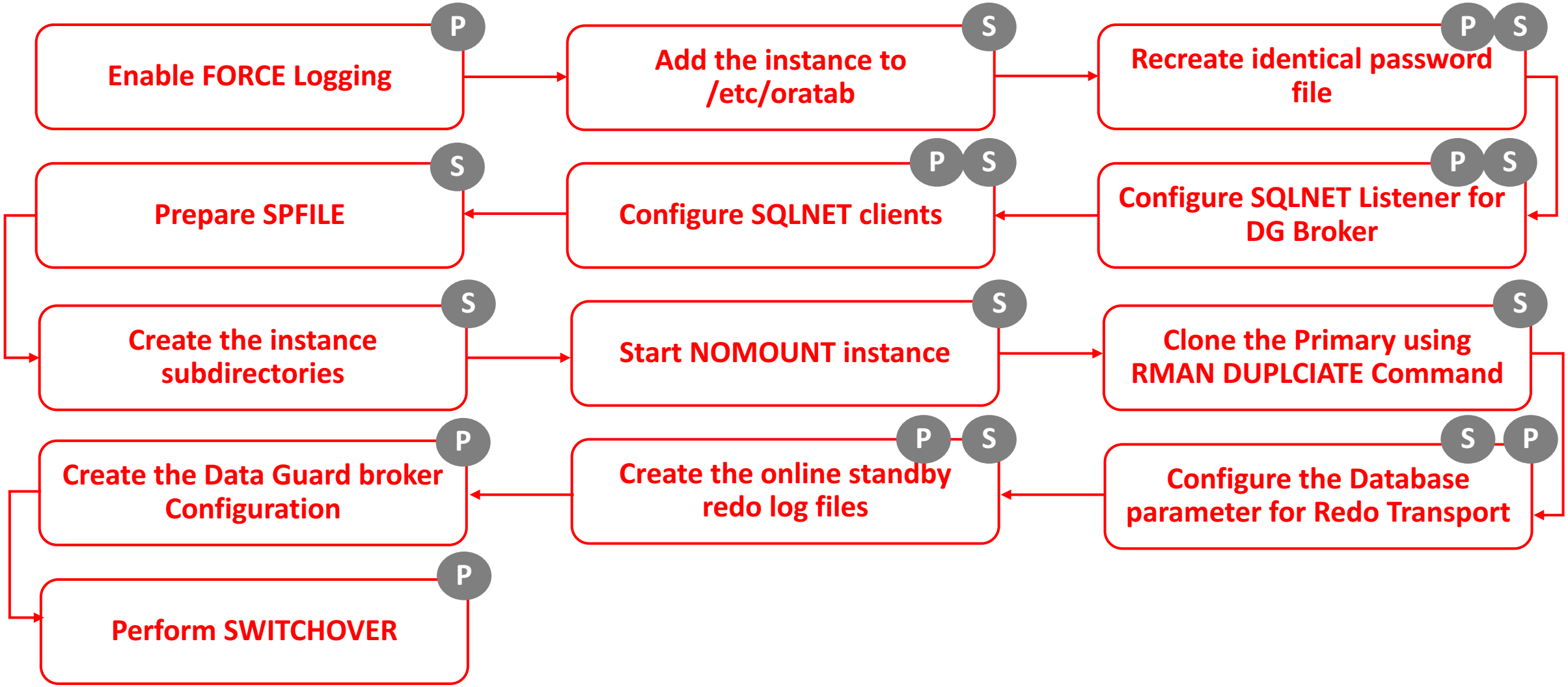
Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Deployment Options

- Manual using offline backup/restore and SQL commands
- Oracle Enterprise Manager Cloud Control (EMCC)
- Online RMAN Duplicate command for standby database

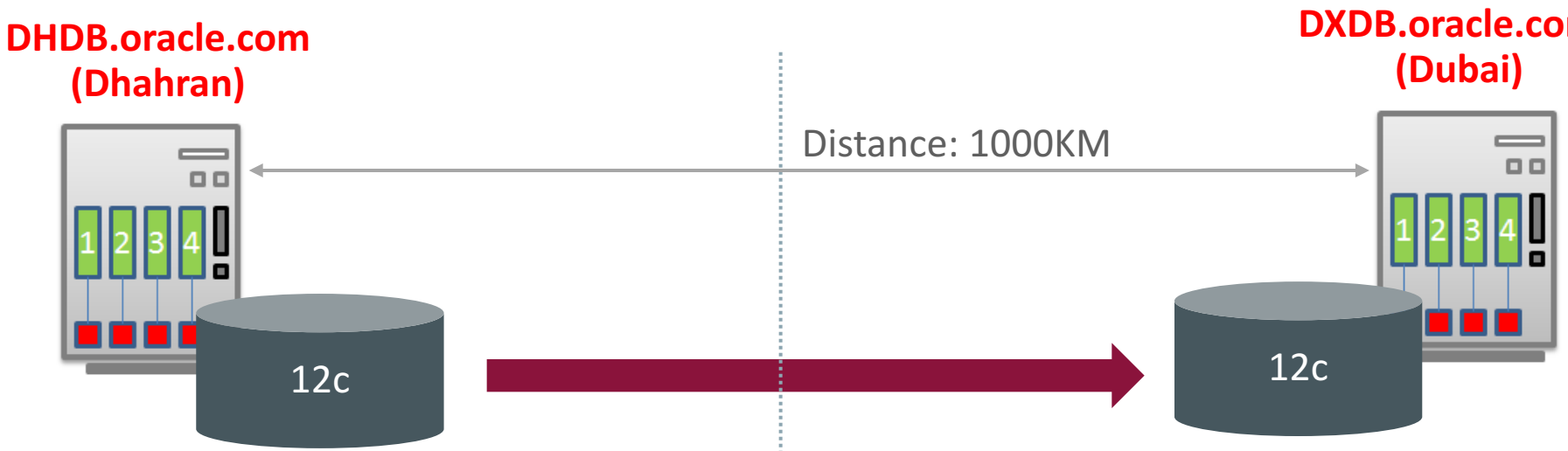
Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Deploying Data Guard using RMAN



Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Disaster Recovery Layout



Primary Database (Dhahran)

Name	DHDB.oracle.com
Version	12.1
RAC	No
ASM	No

SQL NET

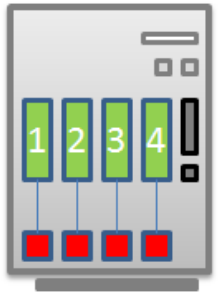
Standby Database (Dubai)

Name	DHDB.oracle.com
Version	12.1
RAC	No
ASM	No

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Local DNS Settings

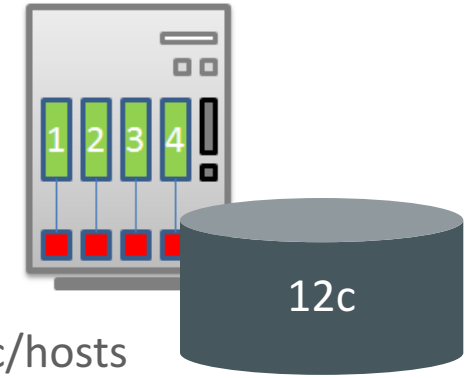
dgobserver.oracle.com



/etc/hosts

```
192.168.1.40    prmy01.oracle.com        prmy01
192.168.1.50    stdby01.oracle.com       stdby01
192.168.1.30    dgobserver.oracle.com    dgobserver
```

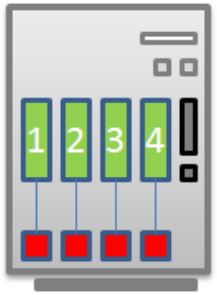
prmy01.oracle.com
(Primary)



/etc/hosts

```
192.168.1.40    prmy01.oracle.com        prmy01
192.168.1.50    stdby01.oracle.com       stdby01
192.168.1.30    dgobserver.oracle.com    dgobserver
```

stdby01.oracle.com
(Stansdby)



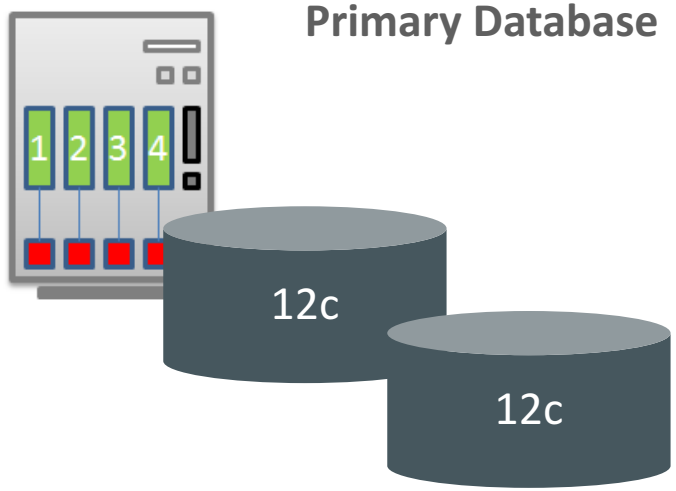
/etc/hosts

```
192.168.1.40    prmy01.oracle.com        prmy01
192.168.1.50    stdby01.oracle.com       stdby01
192.168.1.30    dgobserver.oracle.com    dgobserver
```

```
[root@prmy01 admin]# ping stdby01.oracle.com
PING stdby01.oracle.com (192.168.1.50) 56(84) bytes of data.
64 bytes from stdby01.oracle.com (192.168.1.50): icmp_seq=1 ttl=64 time=1.62 ms
```

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Prepare the Primary Database



Oracle Home	/u01/app/oracle/product/12.1.0/dbhome_1
Oracle SID	DHDB
Global Name	oracle.com
Archive Log	Enabled
SYS Password	oracle_4U
SYSTEM Password	oracle_4U
Character set	AL32UTF8

```
SQL> conn / as sysdba
Connected.
SQL> alter database force logging;

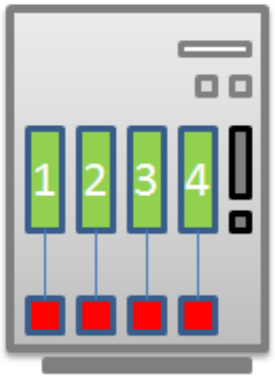
Database altered.

SQL>
```

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Prepare for the Standby Database

Standby Database




```
# The first and second fields are the system identifier and home
# directory of the database respectively.  The third field indicates
# to the dbstart utility that the database should , "Y", or should not,
# "N", be brought up at system boot time.
#
# Multiple entries with the same $ORACLE_SID are not allowed.


DXDB:/u01/app/oracle/product/12.1.0/dbhome_1:Y
```

Database SID

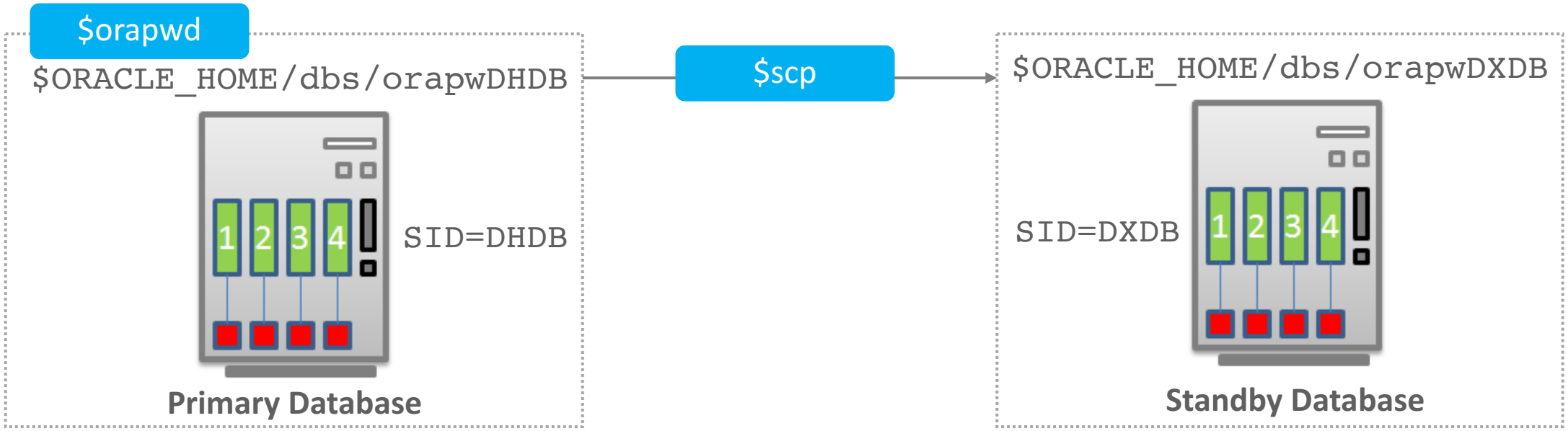
Oracle Home

Y | N

Add this line to /etc/oratab

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Create the Database Password Files

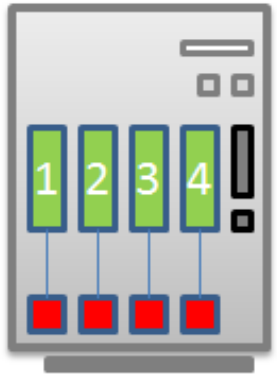



```
$orapwd file=$ORACLE_HOME/dbs/orapwDHDB password=oracle ignorecase=y
force=y entries=30

$scp $ORACLE_HOME/dbs/orapwDHDB
oracle@stdby01.oracle.com:$ORACLE_HOME/dbs/orapwDXDB
```

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Configure SQLNET Listener - Primary



Primary Database

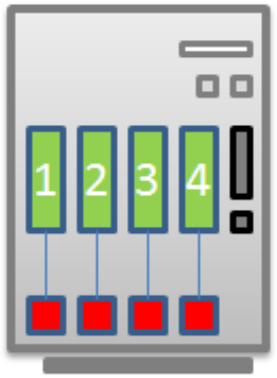
```
SID_LIST_LISTENER =
  (SID_LIST =
    (SID_DESC =
      (GLOBAL_DBNAME = DHDB.oracle.com)
      (ORACLE_HOME = /u01/app/oracle/product/12.1.0/dbhome_1)
      (SID_NAME = DHDB)
    )
    (SID_DESC =
      (GLOBAL_DBNAME = DHDB_DGMGRL.oracle.com)
      (ORACLE_HOME = /u01/app/oracle/product/12.1.0/dbhome_1)
      (SID_NAME = DHDB)
    )
  )
```

TNS_ADMIN

```
/u01/app/oracle/product/12.1.0/dbhome_1/network/admin
```

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Configure SQLNET Listener - Standby



Standby Database

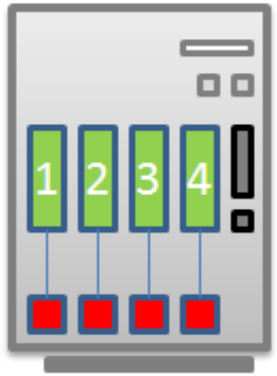
```
SID_LIST_LISTENER =
  (SID_LIST =
    (SID_DESC =
      (GLOBAL_DBNAME = DXDB.oracle.com)
      (ORACLE_HOME = /u01/app/oracle/product/12.1.0/dbhome_1)
      (SID_NAME = DXDB)
    )
    (SID_DESC =
      (GLOBAL_DBNAME = DXDB_DGMGRL.oracle.com)
      (ORACLE_HOME = /u01/app/oracle/product/12.1.0/dbhome_1)
      (SID_NAME = DXDB)
    )
  )
```

TNS_ADMIN

/u01/app/oracle/product/12.1.0/dbhome_1/network/admin

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Configure SQLNET Clients - Primary



Primary Database

```
DXDB =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = stdby01.oracle.com)(PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = DXDB.oracle.com)
    )
  )
DHDB =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = prmy01.oracle.com)(PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = DHDB.oracle.com)
    )
  )
```

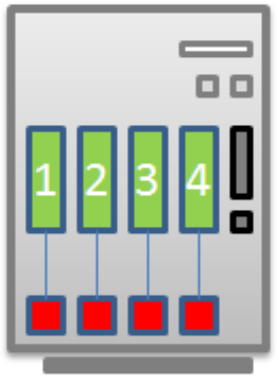
TNS_ADMIN

/u01/app/oracle/product/12.1.0/dbhome_1/network/admin

$ tnsping dxdb

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Configure SQLNET Clients - Standby



Standby Database

```
DXDB =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = stdby01.oracle.com)(PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = DXDB.oracle.com)
    )
  )
DHDB =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = prmy01.oracle.com)(PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = DHDB.oracle.com)
    )
  )
```

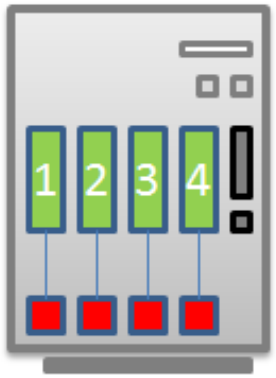
TNS_ADMIN

```
/u01/app/oracle/product/12.1.0/dbhome_1/network/admin
```

```
$tnsping dhdb
```

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Prepare Initialization Parameter File



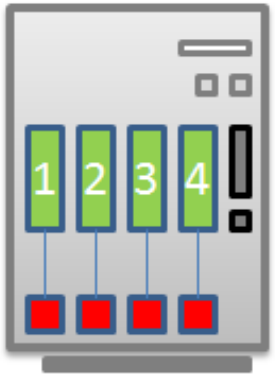
Standby Database

```
db_name=DXDB
control_files='/u01/app/oracle/oradata/DXDB/control01.ctl'
log_file_name_convert='/u01/app/oracle/oradata/DHDB',
                      '/u01/app/oracle/oradata/DXDB'
db_file_name_convert='/u01/app/oracle/oradata/DHDB',
                     '/u01/app/oracle/oradata/DXDB'
```

Parameter	Description
log_file_name_convert	LOG_FILE_NAME_CONVERT converts the filename of a new log file on the primary database to the filename of a log file on the standby database. If you add a log file to the primary database, you must add a corresponding file to the standby database.
db_file_name_convert	DB_FILE_NAME_CONVERT is useful for creating a duplicate database for recovery purposes. It converts the filename of a new datafile on the primary database to a filename on the standby database.

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Create Sub-Directories



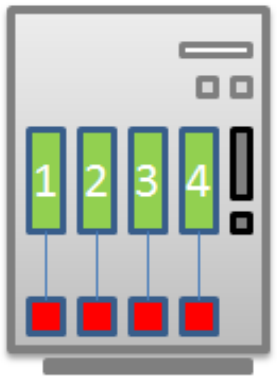
Standby Database

```
[oracle@stdby01 admin]$ mkdir -p /u01/app/oracle/oradata/DXDB
[oracle@stdby01 admin]$ mkdir -p /u01/app/oracle/oradata/DXDB/arch
[oracle@stdby01 admin]$ mkdir -p /u01/app/oracle/admin/DXDB/adump
[oracle@stdby01 admin]$ mkdir -p /u01/app/oracle/admin/DXDB/bdump
[oracle@stdby01 admin]$ mkdir -p /u01/app/oracle/admin/DXDB/cdump
[oracle@stdby01 admin]$ mkdir -p /u01/app/oracle/admin/DXDB/dpdump
[oracle@stdby01 admin]$ mkdir -p /u01/app/oracle/admin/DXDB/pfile
[oracle@stdby01 admin]$ mkdir -p /u01/app/oracle/admin/DXDB/scripts
[oracle@stdby01 admin]$ mkdir -p /u01/app/oracle/admin/DXDB/udump
[oracle@stdby01 admin]$ mkdir -p /u01/app/oracle/fast_recovery_area
```

Directory	Description
arch	Stores archived redo log file
adump	Audit dump files
bdump	Backup ground dump files
cdump	Core dump files
dpdump	Data pump dump file
pfile	Parameter file
sripts	SQL scripts
udump	User’s dump file
Fast_recovery_area	Fast Recovery Area (FRA)

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Startup NOMOUNT Database



Standby Database

$ORACLE_HOME/dbs/initDXDB.ora

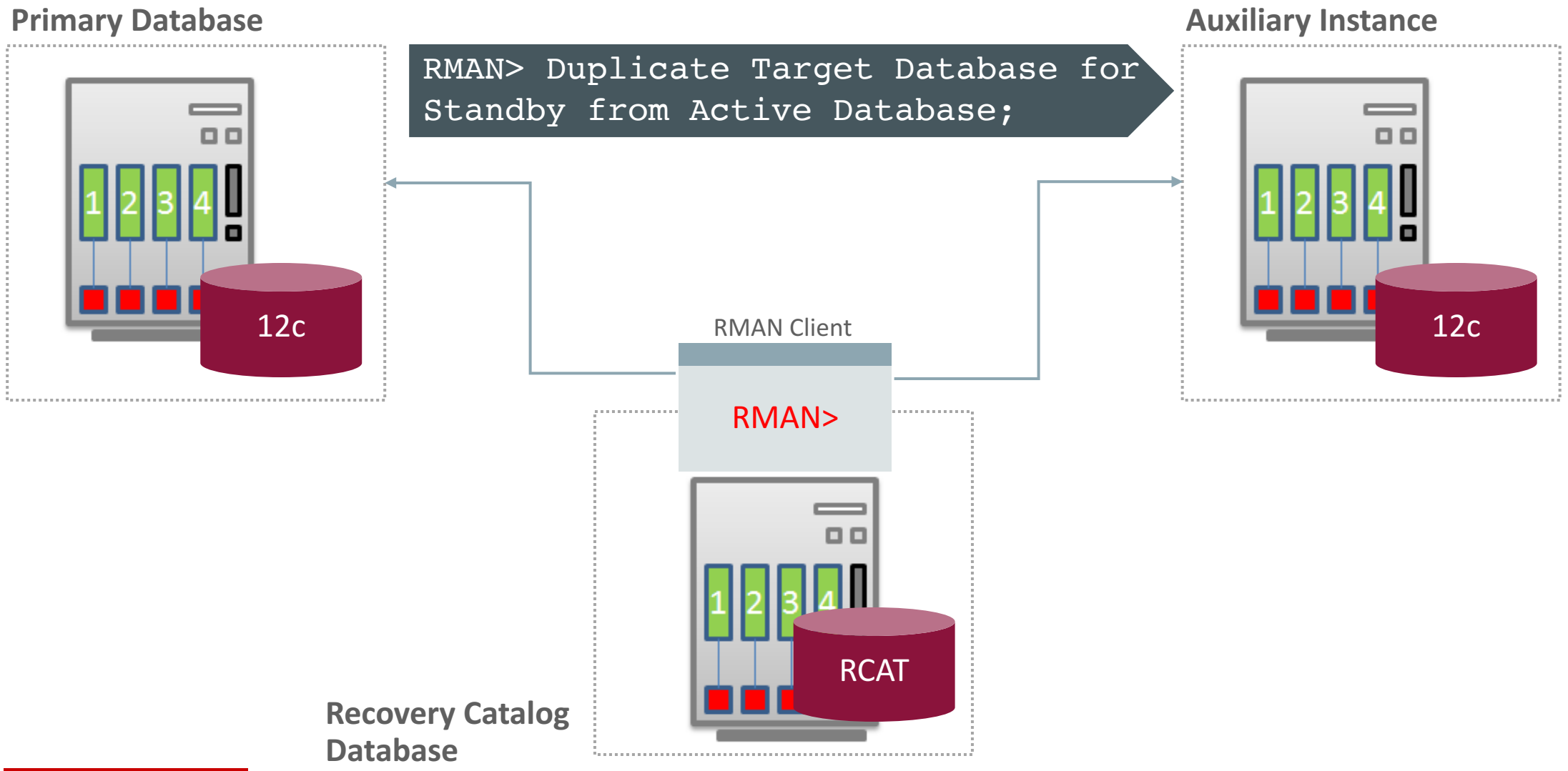
```
db_name=DXDB
control_files='/u01/app/oracle/oradata/DXDB/control01.ctl'
log_file_name_convert='/u01/app/oracle/oradata/DHDB',
                      '/u01/app/oracle/oradata/DXDB'
db_file_name_convert='/u01/app/oracle/oradata/DHDB',
                     '/u01/app/oracle/oradata/DXDB'
```

```
SQL> conn / as sysdba
Connected to an idle instance.
SQL> startup pfile='$ORACLE_HOME/dbs/initDXDB.ora' nomount
ORACLE instance started.


Total System Global Area      217157632 bytes
Fixed Size                      2286656 bytes
Variable Size                 159386560 bytes
Database Buffers               50331648 bytes
Redo Buffers                    5152768 bytes
SQL>
```

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

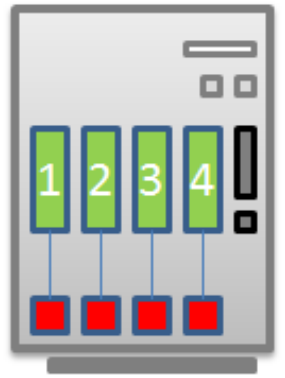
RMAN Auxiliary Instance Architecture



Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

RMAN Duplicate Command

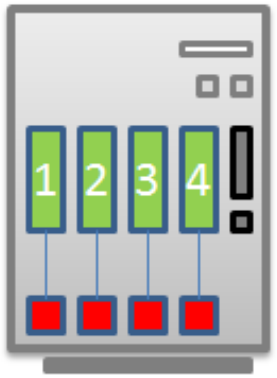
```
run{
allocate channel c1 type disk;
allocate channel c2 type disk;
allocate channel c3 type disk;
allocate auxiliary channel c5 type disk;
duplicate target database for standby from active database
spfile
   parameter_value_convert '/DHDB', '/DXDB'
   set db_unique_name='DXDB'
   set db_file_name_convert='/DHDB/', '/DXDB/'
   set log_file_name_convert='/DHDB/', '/DXDB/'
   set control_files='/u01/app/oracle/oradata/DXDB/control01.ctl'
   set log_archive_max_processes='5'
   set fal_client='DXDB'
   set fal_server='DHDB'
   set standby_file_management='AUTO'
   set log_archive_config='dg_config=(DHDB, DXDB)'
   set log_archive_dest_2='service=DHDB ASYNC valid_for=(online_logfile, primary_role)
                           db_unique_name=DHDB';
}
```



Standby Database

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Run RMAN Duplicate Command



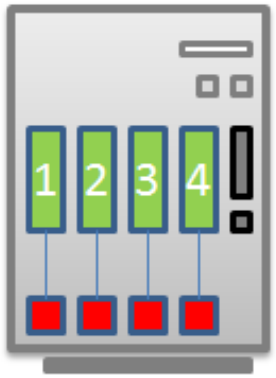
Standby Database

```
run{
allocate channel c1 type disk;
allocate channel c2 type disk;
allocate channel c3 type disk;
allocate auxiliary channel c5 type disk;
duplicate target database for standby from active database
spfile
   parameter_value_convert '/DHDB', '/DXDB'
   set db_unique_name='DXDB'
   set db_file_name_convert='/DHDB/', '/DXDB/'
   set log_file_name_convert='/DHDB/', '/DXDB/'
   set control_files='/u01/app/oracle/oradata/DXDB/control01.ctl'
   set log_archive_max_processes='5'
   set fal_client='DXDB'
   set fal_server='DHDB'
   set standby_file_management='AUTO'
   set log_archive_config='dg_config=(DHDB, DXDB)'
   set log_archive_dest_2='service=DHDB ASYNC valid_for=(online_logfile, primary_role)
                           db_unique_name=DHDB';
}
```

```
rman target=sys/oracle@DHDB auxiliary=sys/oracle@DXDB @/home/oracle/admin/s9_dupDB.rmn;
```

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Verify Status



Standby Database

```
SQL> conn / as sysdba
Connected.
SQL> select database_role from v$database;


DATABASE_ROLE
----------------
PHYSICAL STANDBY


SQL> select open_mode from v$database;


OPEN_MODE
--------------------
MOUNTED


SQL>
```

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Configure the Primary Database

```
alter system set dg_broker_start=true scope=both;
alter system set
dg_broker_config_file1='/u01/app/oracle/product/12.1.0/dbhome_1/dbs/DHDB_dg1.conf'
scope=spfile;
alter system set
dg_broker_config_file2='/u01/app/oracle/product/12.1.0/dbhome_1/dbs/DHDB_dg2.conf'
scope=spfile;
alter system set log_archive_format='DHDB_%t_%s_%r.arc' scope=spfile;
alter system set log_archive_config='DG_CONFIG=(DHDB, DXDB)' scope=both;
alter system set log_archive_dest_2='service=DXDB valid_for=(online_logfile, primary_role)
db_unique_name=DXDB' scope=both;
alter system set log_archive_dest_state_2=enable scope=both;
alter system set log_archive_max_processes=4 scope=both;
alter system set fal_server=DXDB scope=spfile;
alter system set fal_client=DHDB scope=spfile;
alter system set log_file_name_convert='/u01/app/oracle/oradata/DXDB',
'/u01/app/oracle/oradata/DHDB' scope=spfile;
alter system set db_file_name_convert='/u01/app/oracle/oradata/DXDB',
'/u01/app/oracle/oradata/DHDB' scope=spfile;
alter system set standby_file_management=auto scope=spfile;
```

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

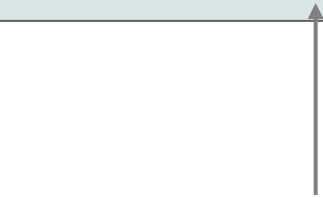
Configure the Standby Database

```
alter system set dg_broker_start=true scope=both;
alter system set
dg_broker_config_file1='/u01/app/oracle/product/12.1.0/dbhome_1/dbs/DXDB_dg1.conf'
scope=spfile;
alter system set
dg_broker_config_file2='/u01/app/oracle/product/12.1.0/dbhome_1/dbs/DXDB_dg2.conf'
scope=spfile;
alter system set log_archive_format='DXDB_%t_%s_%r.arc' scope=spfile;
alter system set log_archive_config='DG_CONFIG=(DHDB, DXDB)' scope=both;
alter system set log_archive_dest_2='service=DHDB valid_for=(online_logfile,
primary_role) db_unique_name=DHDB' scope=both;
alter system set log_archive_dest_state_2=enable scope=both;
alter system set log_archive_max_processes=4 scope=both;
alter system set fal_server=DHDB scope=spfile;
alter system set fal_client=DXDB scope=spfile;
alter system set
log_file_name_convert='/u01/app/oracle/oradata/DHDB','/u01/app/oracle/oradata/DXDB'
scope=spfile;
alter system set db_file_name_convert='/u01/app/oracle/oradata/DHDB',
'/u01/app/oracle/oradata/DXDB' scope=spfile;
alter system set standby_file_management=auto scope=spfile;
```

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Startup the Standby Database

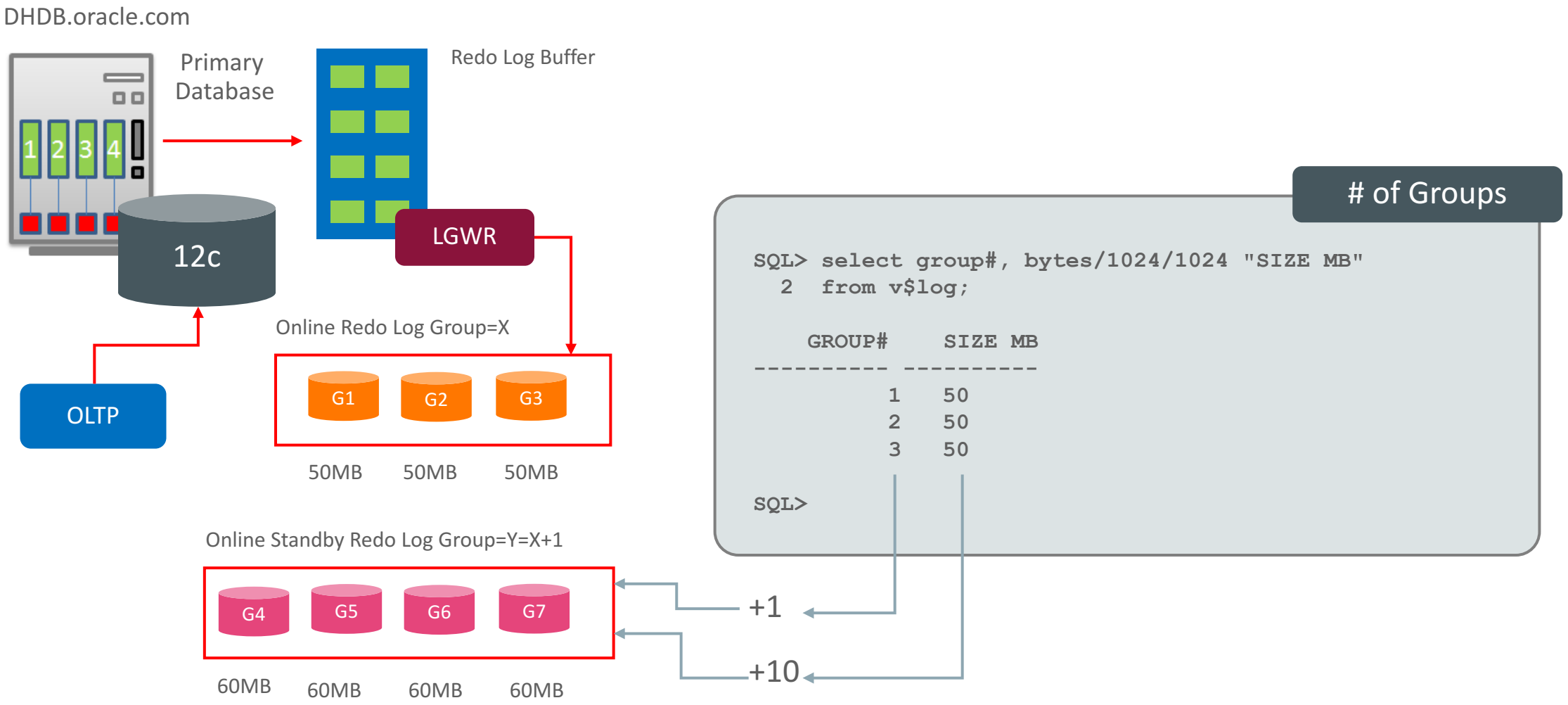
```
shutdown immediate;
startup;
alter database recover managed standby database disconnect from session;
```



Running as Background session

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Calculate and Size the Standby Redo Log Files



Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Create Standby Online Groups on Standby System

```
conn / as sysdba
recover managed standby database cancel;
alter database add standby logfile group 4
'/u01/app/oracle/oradata/DXDB/redo04.log' size 60M;
alter database add standby logfile group 5
'/u01/app/oracle/oradata/DXDB/redo05.log' size 60M;
alter database add standby logfile group 6
'/u01/app/oracle/oradata/DXDB/redo06.log' size 60M;
alter database add standby logfile group 7
'/u01/app/oracle/oradata/DXDB/redo07.log' size 60M;
alter database recover managed standby database disconnect from session;
```

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Create Standby Online Groups on the Primary System

```
alter database add standby logfile group 4
'/u01/app/oracle/oradata/DHDB/redo04.log' size 60M;
alter database add standby logfile group 5
'/u01/app/oracle/oradata/DHDB/redo05.log' size 60M;
alter database add standby logfile group 6
'/u01/app/oracle/oradata/DHDB/redo06.log' size 60M;
alter database add standby logfile group 7
'/u01/app/oracle/oradata/DHDB/redo07.log' size 60M;
```

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Create Data Guard Broker Configuration

Primary

```
connect / as sysdba
alter system set log_archive_dest_2='' scope=both;
eof
dgmgrl<<eof
connect sys/oracle@DHDB
create configuration 'DG_CONF' as primary database is 'DHDB' connect
identifier is 'DHDB';
enable configuration;
add database 'DXDB' as connect identifier is 'DXDB';
enable database 'DHDB';
enable database 'DXDB';
eof
```

Standby

```
SQL> alter system set log_archive_dest_2='' scope=both;

System altered.

SQL>
```

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Generate Workload

```
SQL> exec call_trans_load;
```

Run the workload simulator using the default values. The actual parameters uses the default values; N=500 and S=5.

```
SQL> exec call_trans_load(2000);
```

Run the workload simulator using N=2000, this will sustain transactions load of approximately 4hrs.

```
SQL> exec call_trans_load(300000, 1);
```



The following call is an extreme level, use this parameter setting when performing the workshop on 8-16 cores system.

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Verify Redo-Transport

```
Select sequence#, first_time, next_time
From v$archived_log
Order by sequence#;
```

```
SQL>
 SEQUENCE# FIRST_TIM NEXT_TIME
---------- --------- ---------
12 03-NOV-17 03-NOV-17
13 03-NOV-17 03-NOV-17
14 03-NOV-17 03-NOV-17


SQL>
```

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Switchover to Current Standby System

```
connect sys/oracle@DHDB
switchover to 'DXDB';
```

```
DGMGRL> Connected as SYSDBA.
DGMGRL> Performing switchover NOW, please wait...
Operation requires a connection to instance "DXDB" on database "DXDB"
Connecting to instance "DXDB"...
Connected as SYSDBA.
New primary database "DXDB" is opening...
Operation requires startup of instance "DHDB" on database "DHDB"
Starting instance "DHDB"...
ORACLE instance started.
Database mounted.
Database opened.
Switchover succeeded, new primary is "DXDB"
DGMGRL>
```

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

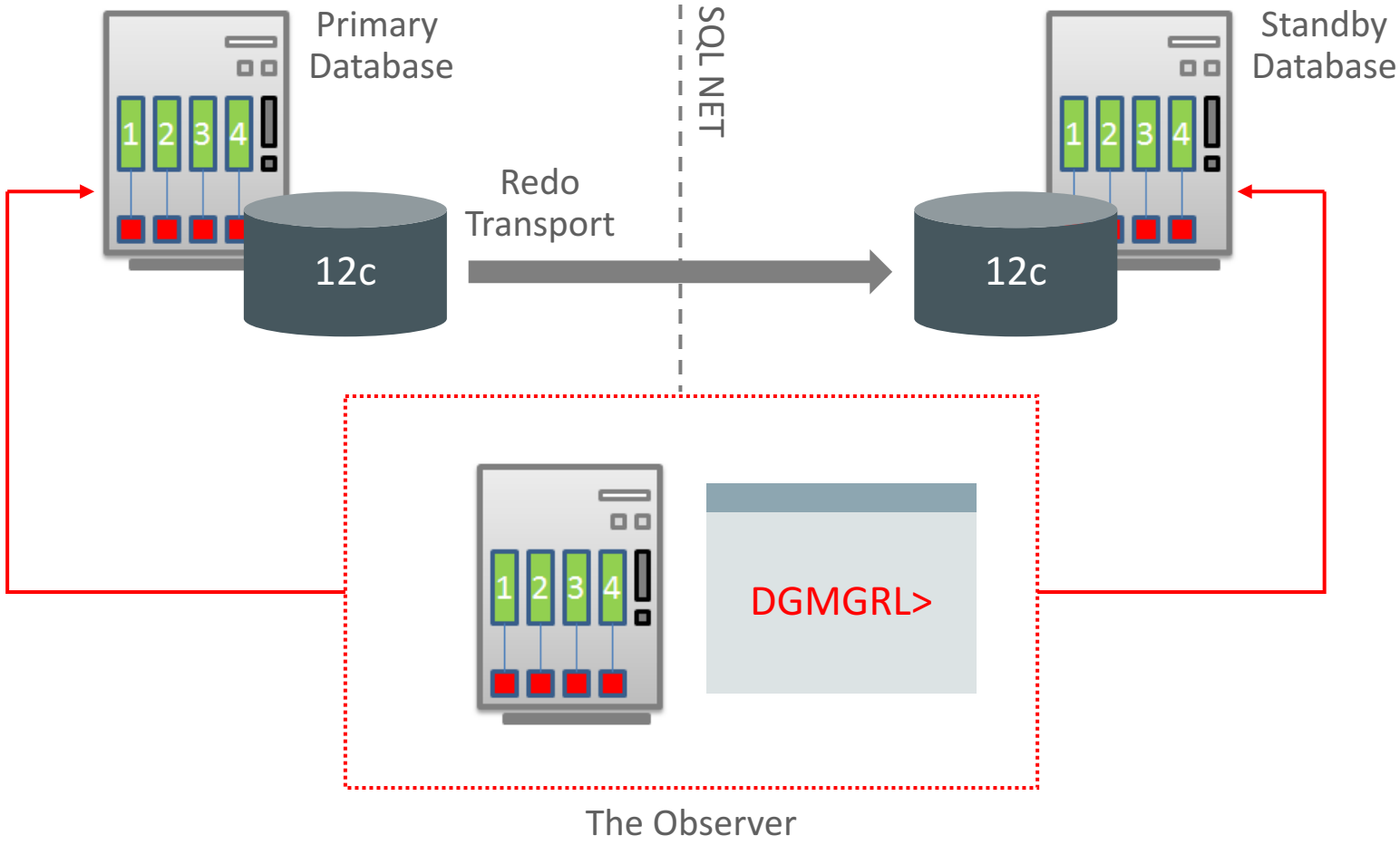
Switchover to Current Standby System

```
connect sys/oracle@DXDB
switchover to 'DHDB';
```

```
DGMGRL> Connected as SYSDBA.
DGMGRL> Performing switchover NOW, please wait...
Operation requires a connection to instance "DHDB" on database "DHDB"
Connecting to instance "DHDB"...
Connected as SYSDBA.
New primary database "DHDB" is opening...
Operation requires startup of instance "DXDB" on database "DXDB"
Starting instance "DXDB"...
ORACLE instance started.
Database mounted.
Database opened.
Switchover succeeded, new primary is "DHDB"
```

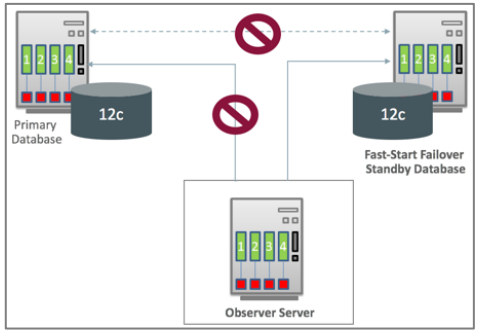
Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Automating Data Guard Failover using FAST-START Feature



Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

When a FAST-START Failover is Triggered?



- Based on pre-determined User's defined Threshold

Loss of connectivity between both the primary database and the observer; and between the primary database and the fast-start failover target standby database exceeds the fast-start failover threshold.

- Default Threshold value is 30 Seconds

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

When a FAST-START Failover is Triggered?

- When the below database health check is configured for FAST-START failover
  - A data file is offline because of a write error.
  - Dictionary corruption of a critical database object.
  - Control file is permanently damaged
  - LGWR is unable to write to the log group
  - Archiver is unable to archive a redo log because

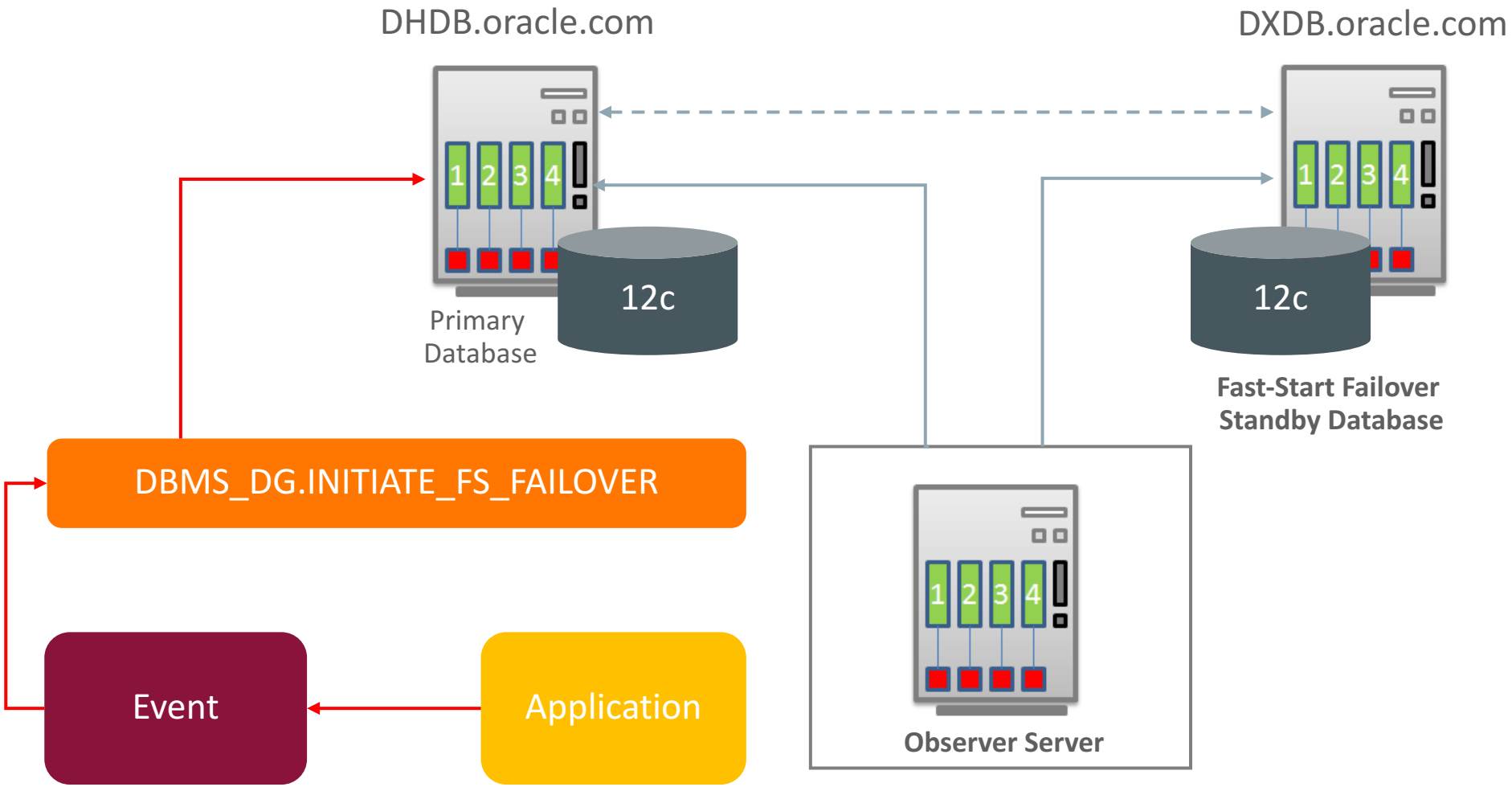
Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

When a FAST-START Failover is Triggered?

- An instance crash occurs for a single-instance database.
- All instance crash occurs for a single-instance database. instances of a Real Application Clusters (RAC) primary database crash.
- Shutdown abort of the primary database occurs.
- An application initiates a fast-start failover by calling the DBMS_DG.INITIATE_FS_FAILOVER function.

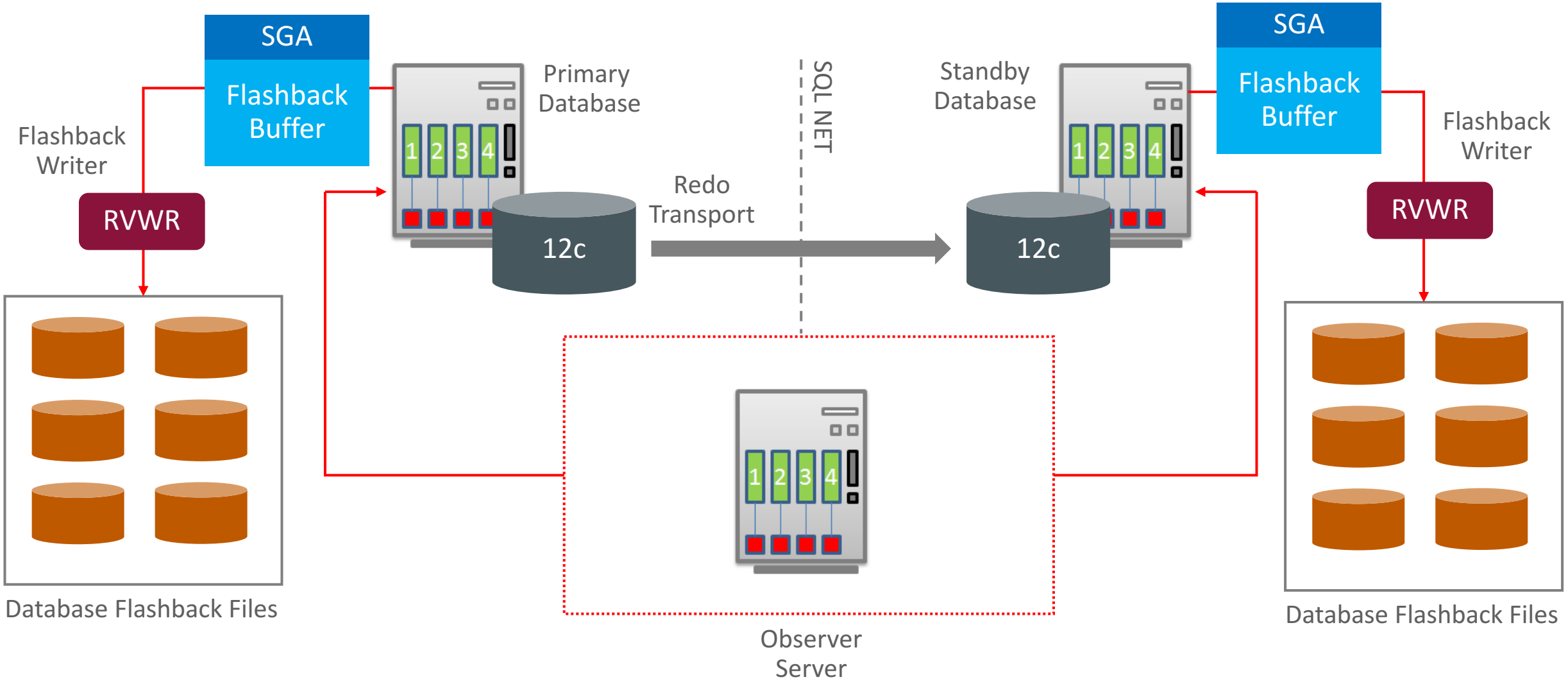
Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Programmatically Performing FAST-START Failover



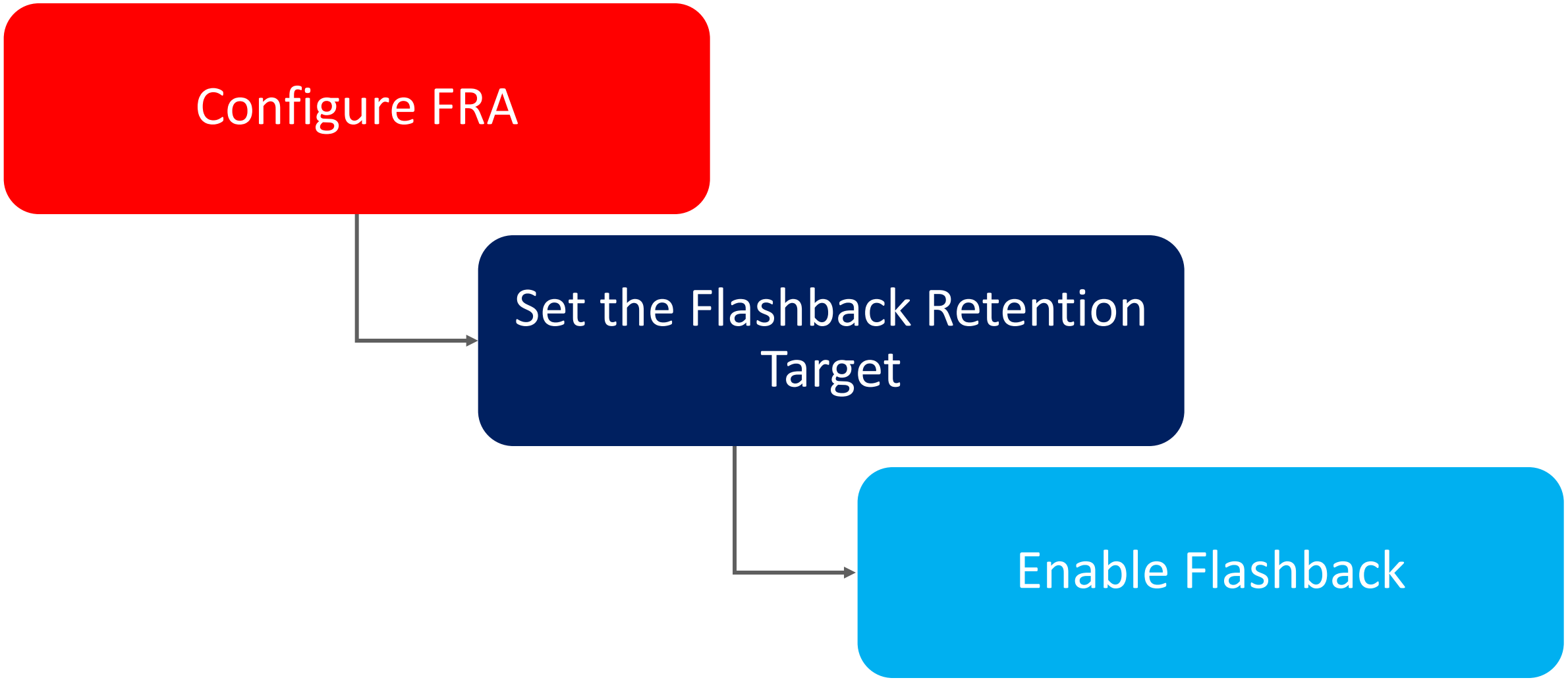
Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Flashback Database



Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Flashback Database Configuration



Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Define and Size Fast Recovery Area (FRA)

Parameter	Description
DB_RECOVERY_FILE_DEST_SIZE	Sets the disk current upper limit
DB_RECOVERY_FILE_DEST	Sets the location of the FRA

```
SQL> show parameters db_recovery_file_dest

NAME                                 TYPE        VALUE
------------------------------------ ----------- ------------------------------
db_recovery_file_dest                string      /u01/app/oracle/fast_recovery_area
db_recovery_file_dest_size           big integer 4800M

SQL> alter system set db_recovery_file_dest_size=9000M scope=both;

System altered.

SQL>
```

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Configure Database Flashback Retention

Database on ARCHIVELOG Mode → Enable Flashback Database

Parameter	Description
DB_FLASHBACK_RETENTION_TARGET	Specify an upper limit, in minutes, on how far back you want to be able to flash back the database

```
SQL> show parameters DB_FLASHBACK_RETENTION_TARGET

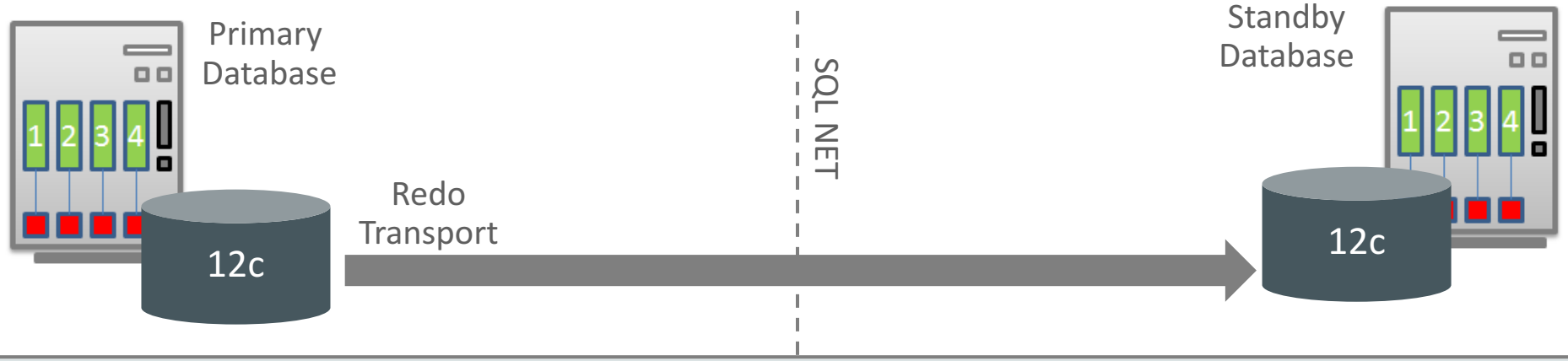
NAME     TYPE VALUE
------------------------------------ -----------    ------------------------------
db_flashback_retention_target        integer        2880
SQL> alter system set DB_FLASHBACK_RETENTION_TARGET=5000 scope=both;

System altered.


SQL>
```

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Enable Database Flashback on the Primary




```
SQL> alter database flashback on;

Database altered.

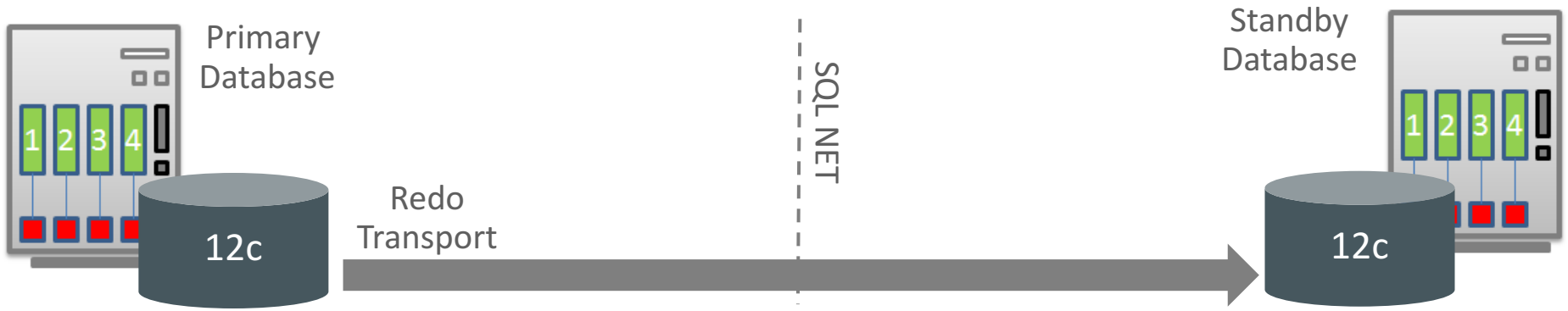
SQL> select flashback_on from v$database;

FLASHBACK_ON
------------------
YES

SQL>
```

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Enable Database Flashback on the Standby




```
SQL> alter database recover managed standby database cancel;

Database altered.

SQL> alter database flashback on;

Database altered.

SQL> alter database recover managed standby database disconnect from session;

Database altered.

SQL>
```

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Verify Configuration

```
DGMGRL> connect sys/oracle
Connected as SYSDG.
DGMGRL>
DGMGRL> show configuration

Configuration - DG_CONF

  Protection Mode: MaxPerformance
  Databases:
  DHDB - Primary database
    DXDB - Physical standby database

Fast-Start Failover: DISABLED

Configuration Status:
SUCCESS

DGMGRL>
```

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Verify Switchover to ‘DXDB’

```
DGMGRL> switchover to 'DXDB';
Performing switchover NOW, please wait...
Operation requires a connection to instance "DXDB" on database "DXDB"
Connecting to instance "DXDB"...
Connected as SYSDBA.
New primary database "DXDB" is opening...
Operation requires startup of instance "DHDB" on database "DHDB"
Starting instance "DHDB"...
ORACLE instance started.
Database mounted.
Database opened.
Switchover succeeded, new primary is "DXDB"
DGMGRL>
```


Successful Switchover

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Verify Switchover to ‘DHDB’

```
DGMGRL> switchover to 'DHDB';
Performing switchover NOW, please wait...
Operation requires a connection to instance "DHDB" on database "DHDB"
Connecting to instance "DHDB"...
Connected as SYSDBA.
New primary database "DHDB" is opening...
Operation requires startup of instance "DXDB" on database "DXDB"
Starting instance "DXDB"...
ORACLE instance started.
Database mounted.
Database opened.
Switchover succeeded, new primary is "DHDB"
DGMGRL>
```


Successful Switchover

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

FAST-START Failover Checklist

Function	Primary	Standby
Archive Log Mode	Enabled	Enabled
Flashback Database	On	On
Fast Recovery Area (FRA) re-configured	Yes	Yes
Flashback Database Retention re-configured	Yes	Yes
DGMGRL> show configuration command	Success	Success
Major error on alertDHDB.log and alertDXDB.log	No	No
Manual Switchover from current Primary system to current Standby system	Success	Success

Do not attempt Fast-Start failover if any of the above is not met.

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

FAST-START Failover Pre-requisites

- Installed Observer Software
- Configuration mode must be on Maximum Availability or Maximum performance
- LogXptmode must be set as SYNC for Maximum Availability and ASYNC for Maximum Performance
- Flashback must be enabled on the primary and standby database
- Configure SQLNET tnsnames.ora file on the observer server
- Set Statics service name for the observer server

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Observer Software Installation Options

- Install the complete Oracle Client Administrator by choosing the Administrator option in the Oracle Universal Installer.
- Install the full Oracle Database software kit. This installation includes DGMGRL.

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Configure and Test SQLNET Client

```
DXDB =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = stdby01.oracle.com)(PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = DXDB.oracle.com)
    )
  )
```

```
DGMGRL> connect sys/oracle@DXDB
Connected as SYSDBA.
DGMGRL>
```

```
DHDB =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = prmy01.oracle.com)(PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = DHDB.oracle.com)
    )
  )
```

```
DGMGRL> connect sys/oracle@DHDB
Connected as SYSDBA.
DGMGRL>
```

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Set Statics Service Name on the Primary

```
(SID_DESC =
      (GLOBAL_DBNAME = DHDB.oracle.com)
      (ORACLE_HOME = /u01/app/oracle/product/12.1.0/dbhome_1)
      (SID_NAME = DHDB)
)
```

```
Services Summary...
Service "DHDB.oracle.com" has 2 instance(s).
  Instance "DHDB", status UNKNOWN, has 1 handler(s) for this service...
  Instance "DHDB", status READY, has 1 handler(s) for this service...
Service "DHDBXDB.oracle.com" has 1 instance(s).
  Instance "DHDB", status READY, has 1 handler(s) for this service...
Service "DHDB_DGB.oracle.com" has 1 instance(s).
  Instance "DHDB", status READY, has 1 handler(s) for this service...
Service "DHDB_DGMGRL.oracle.com" has 1 instance(s).
  Instance "DHDB", status UNKNOWN, has 1 handler(s) for this service...
The command completed successfully
```

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

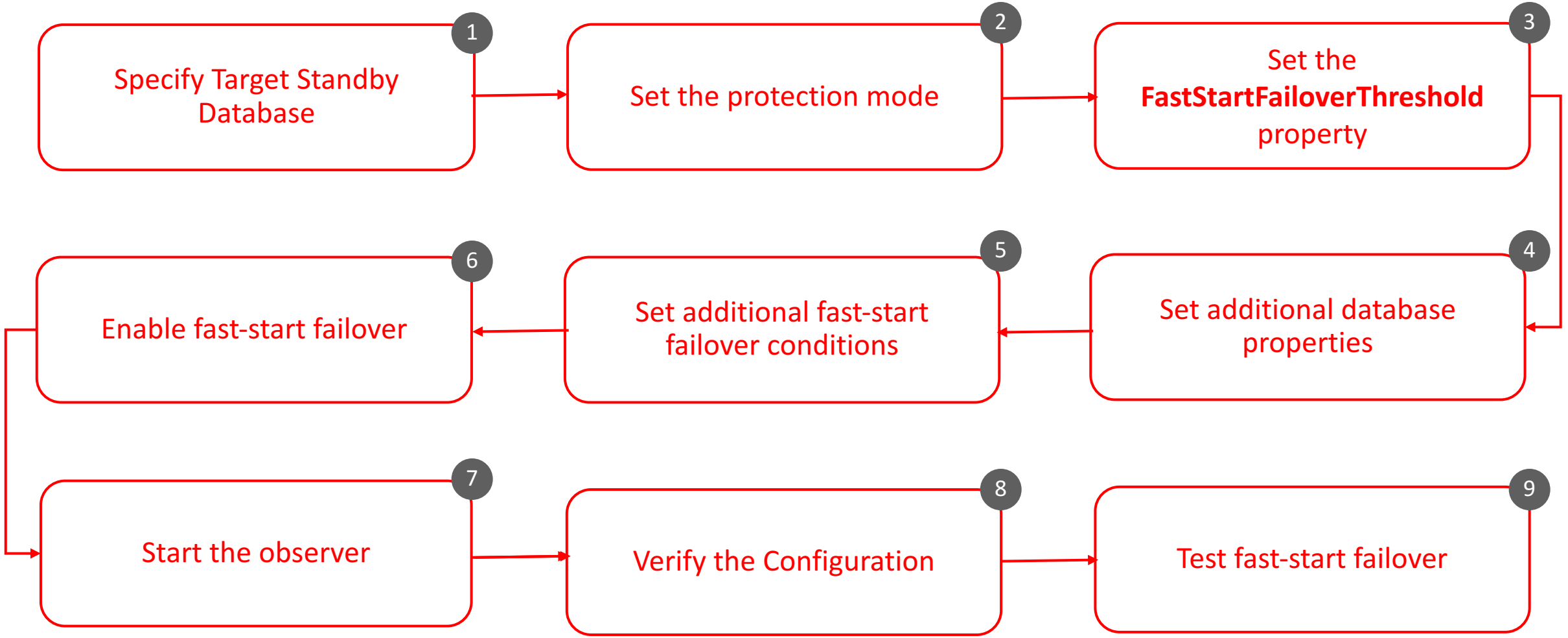
Set Statics Service Name on the Standby

```
(SID_DESC =
      (GLOBAL_DBNAME = DXDB.oracle.com)
      (ORACLE_HOME = /u01/app/oracle/product/12.1.0/dbhome_1)
      (SID_NAME = DXDB)
)
```

```
Services Summary...
Service "DHDBXDB.oracle.com" has 1 instance(s).
  Instance "DXDB", status READY, has 1 handler(s) for this service...
Service "DXDB.oracle.com" has 2 instance(s).
  Instance "DXDB", status UNKNOWN, has 1 handler(s) for this service...
  Instance "DXDB", status READY, has 1 handler(s) for this service...
Service "DXDB_DGB.oracle.com" has 1 instance(s).
  Instance "DXDB", status READY, has 1 handler(s) for this service...
Service "DXDB_DGMGRL.oracle.com" has 1 instance(s).
  Instance "DXDB", status UNKNOWN, has 1 handler(s) for this service...
The command completed successfully
LSNRCTL>
```

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Enable Fast-Start Failover



Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Fast-Start Failover Default Settings

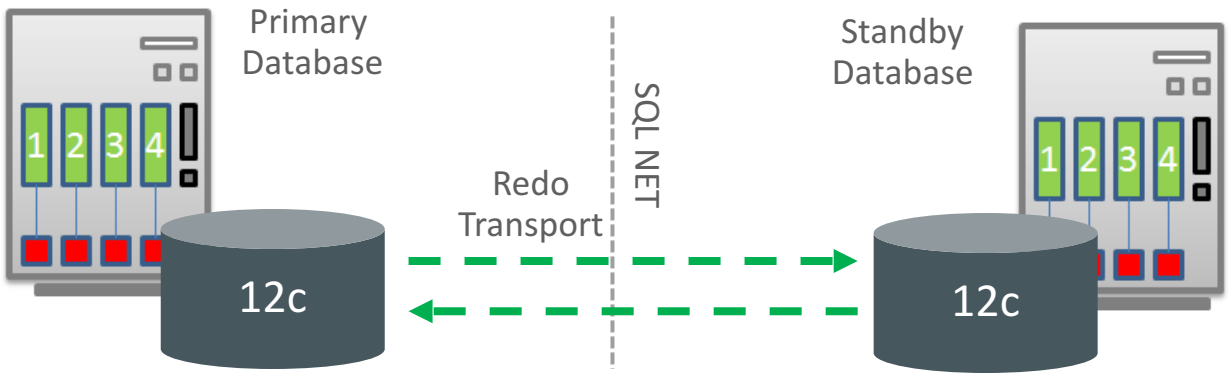
```
  Protection Mode: MaxPerformance
  Databases:
  DHDB - Primary database
    DXDB - Physical standby database

  Properties:
    FastStartFailoverThreshold      = '30'
    OperationTimeout                = '30'
    TraceLevel                      = 'USER'
    FastStartFailoverLagLimit       = '30'
    CommunicationTimeout            = '180'
    ObserverReconnect               = '0'
    FastStartFailoverAutoReinstate  = 'TRUE'
    FastStartFailoverPmyShutdown    = 'TRUE'
    BystandersFollowRoleChange      = 'ALL'
    ObserverOverride                = 'FALSE'

Fast-Start Failover: DISABLED
```

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Specify Target Standby Database




```
DGMGRL> edit database 'DHDB'
> set property FastStartFailoverTarget = 'DXDB';
Property "faststartfailovertarget" updated
DGMGRL>
```

```
DGMGRL> edit database 'DXDB'
> set property FastStartFailoverTarget = 'DHDB';
Property "faststartfailovertarget" updated
DGMGRL>
```

```
DGMGRL> show configuration


Configuration - DG_CONF


  Protection Mode: MaxPerformance
  Databases:
  DHDB - Primary database
    DXDB - Physical standby database

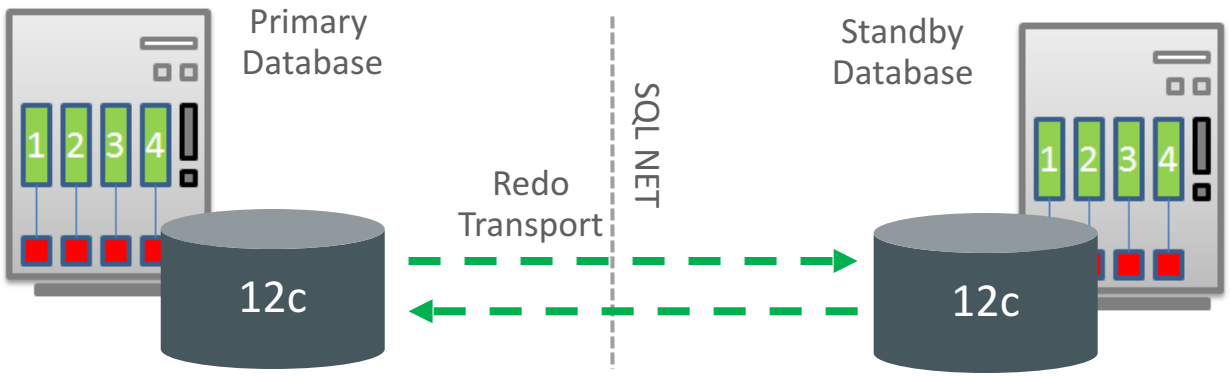

Fast-Start Failover: DISABLED


Configuration Status:
SUCCESS


DGMGRL>
```

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Set the Protection Mode




```
DGMGRL> EDIT DATABASE 'DHDB'
> set property LogXptMode='ASYNC';
Property "logxptmode" updated
DGMGRL> EDIT DATABASE 'DXDB'
> set property LogXptMode='ASYNC';
Property "logxptmode" updated
DGMGRL> EDIT CONFIGURATION
> SET PROTECTION MODE AS MaxPerformance;
Succeeded.
DGMGRL>
```

```
DGMGRL> show configuration


Configuration - DG_CONF


  Protection Mode: MaxPerformance
  Databases:
  DHDB - Primary database
    DXDB - Physical standby database

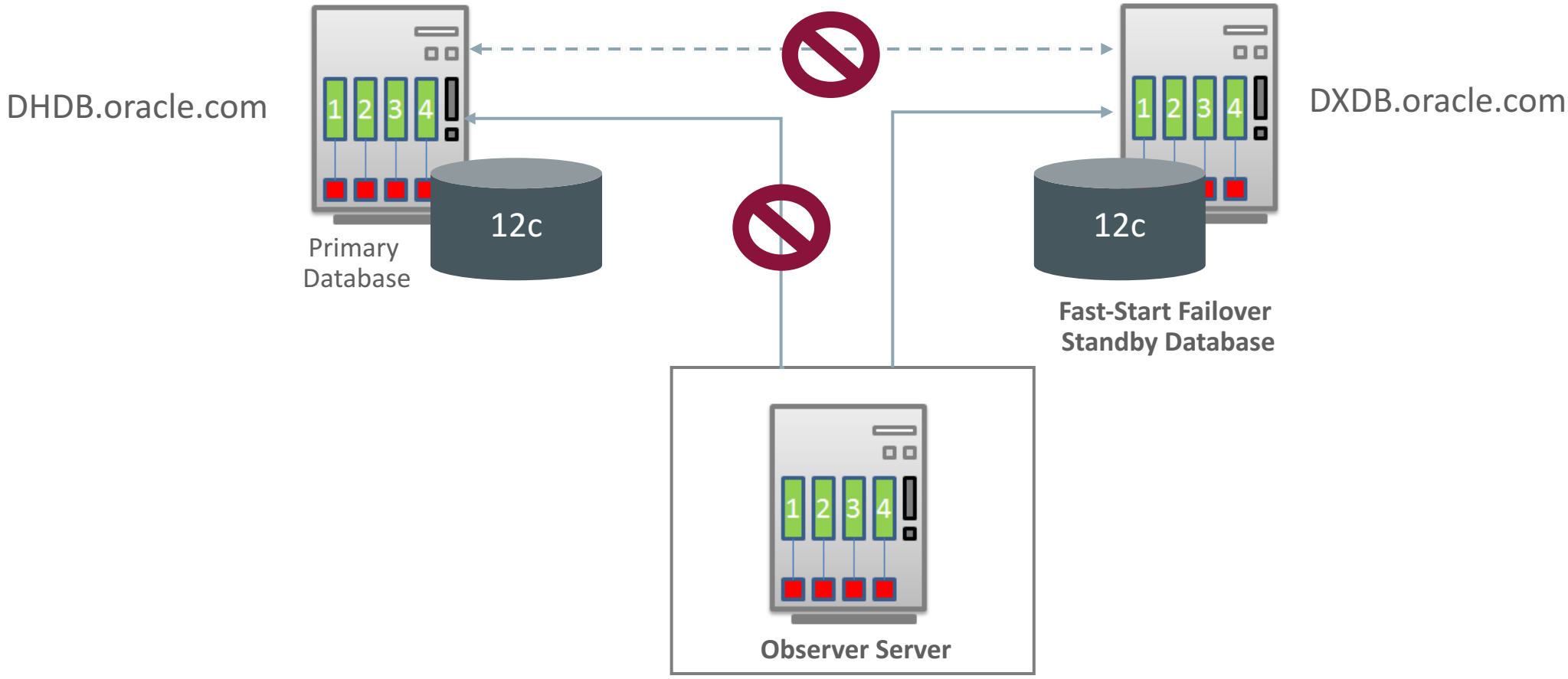

Fast-Start Failover: DISABLED


Configuration Status:
SUCCESS


DGMGRL>
```

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

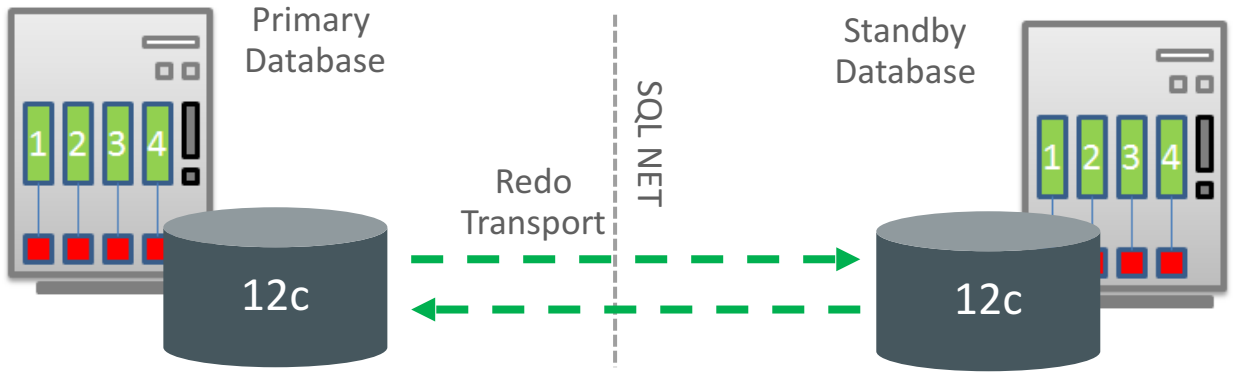
FastStartFailoverThreshold Property



Loss of Connectivity > fast-start failover threshold

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Set the **FastStartFailoverThreshold** Property



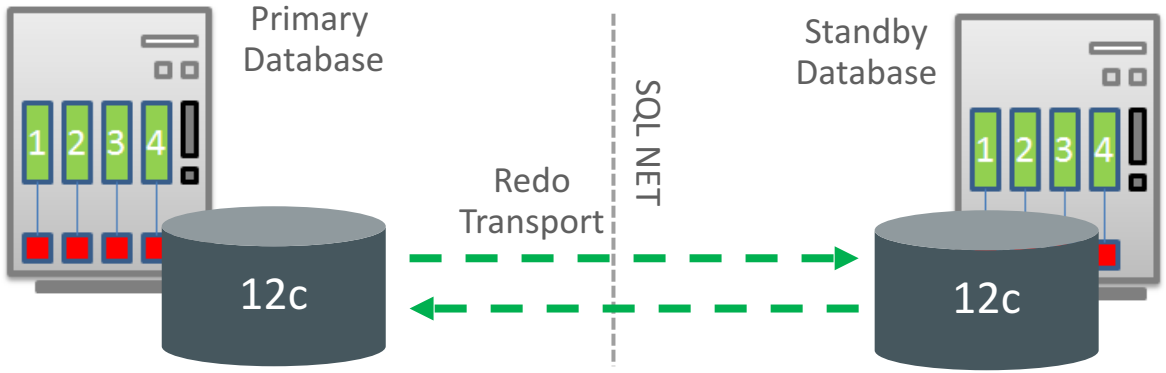

```
DGMGRL> EDIT CONFIGURATION
> SET PROPERTY FastStartFailoverThreshold = 45;
Property "faststartfailoverthreshold" updated
DGMGRL>
```

45 seconds before Automatic failover occur

- The fast-start failover threshold specifies how long the observer and target standby database should simultaneously wait to hear from the primary database before initiating a fast-start failover.
- The threshold value is specified as a positive, nonzero number of seconds. The default value for the FastStartFailoverThreshold property is 30 seconds.
- When choosing a value for this property, you must balance the increased risk of an unnecessary failover (for example, if the network connection was temporarily broken for a few seconds) against the benefits of faster failover and less down time during a critical outage

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Additional Database Properties



Establishes an acceptable length of time for the standby to fall behind the primary database with respect to applied redo

Determines whether the old primary database should be automatically reinstated

Determines whether the primary database shuts down if redo generation has stalled and the primary database has lost connectivity with the observer and target standby database for a longer time than `FastStartFailoverThreshold`

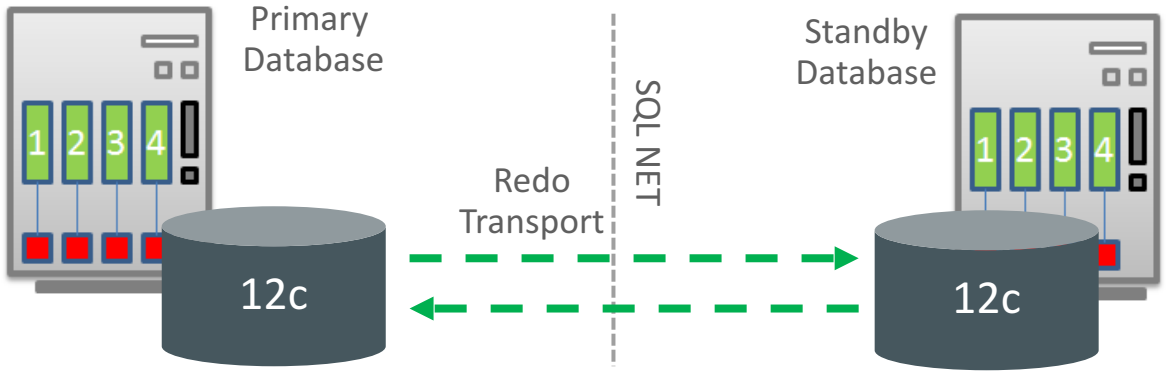
```
  Protection Mode: MaxPerformance
  Databases:
  DHDB - Primary database
    DXDB - Physical standby database

  Properties:
    FastStartFailoverThreshold      = '45'
    OperationTimeout                = '30'
    TraceLevel                      = 'USER'
    FastStartFailoverLagLimit       = '30'
    CommunicationTimeout            = '180'
    ObserverReconnect               = '0'
    FastStartFailoverAutoReinstate  = 'TRUE'
    FastStartFailoverPmyShutdown    = 'TRUE'
    BystandersFollowRoleChange      = 'ALL'
    ObserverOverride                = 'FALSE'

Fast-Start Failover: DISABLED
```

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Set Additional Database Properties




```
DGMGRL> edit configuration set property
> FastStartFailoverLagLimit = 45;
Property "faststartfailoverlaglimit" updated
DGMGRL>
```

```
DGMGRL> edit configuration set property
> FastStartFailoverPmyShutdown = false;
Property "faststartfailoverpmyshutdown"
updated
DGMGRL>
```

```
  Protection Mode: MaxPerformance
  Databases:
  DHDB - Primary database
    DXDB - Physical standby database

  Properties:
    FastStartFailoverThreshold      = '45'
    OperationTimeout                = '30'
    TraceLevel                      = 'USER'
    FastStartFailoverLagLimit       = '30'
    CommunicationTimeout            = '180'
    ObserverReconnect               = '0'
    FastStartFailoverAutoReinstate  = 'TRUE'
    FastStartFailoverPmyShutdown    = 'TRUE'
    BystandersFollowRoleChange      = 'ALL'
    ObserverOverride                = 'FALSE'

Fast-Start Failover: DISABLED
```

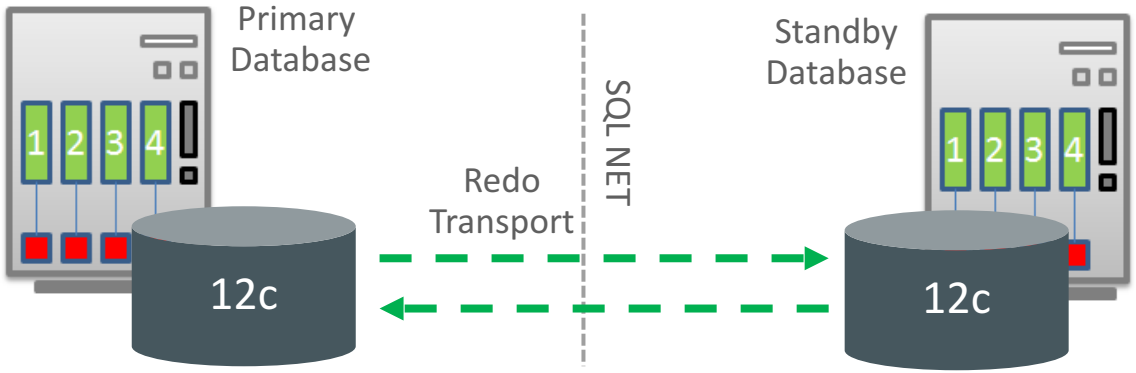
Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Fast-Start Failover Conditions

Condition	Default Setting	Description
Datafile Offline	Enabled	
Corrupted Controlfile	Enabled	
Corrupted Dictionary	Enabled	
Inaccessible Logfile	Disabled	
Stuck Archiver	Disabled	

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Set Additional Fast-Start Failover Conditions




```
DGMGRL> enable fast_start failover
condition '1578';
Succeeded.
DGMGRL>
```

```
DGMGRL> enable fast_start failover
condition "Inaccessible Logfile";
Succeeded.
DGMGRL>
```

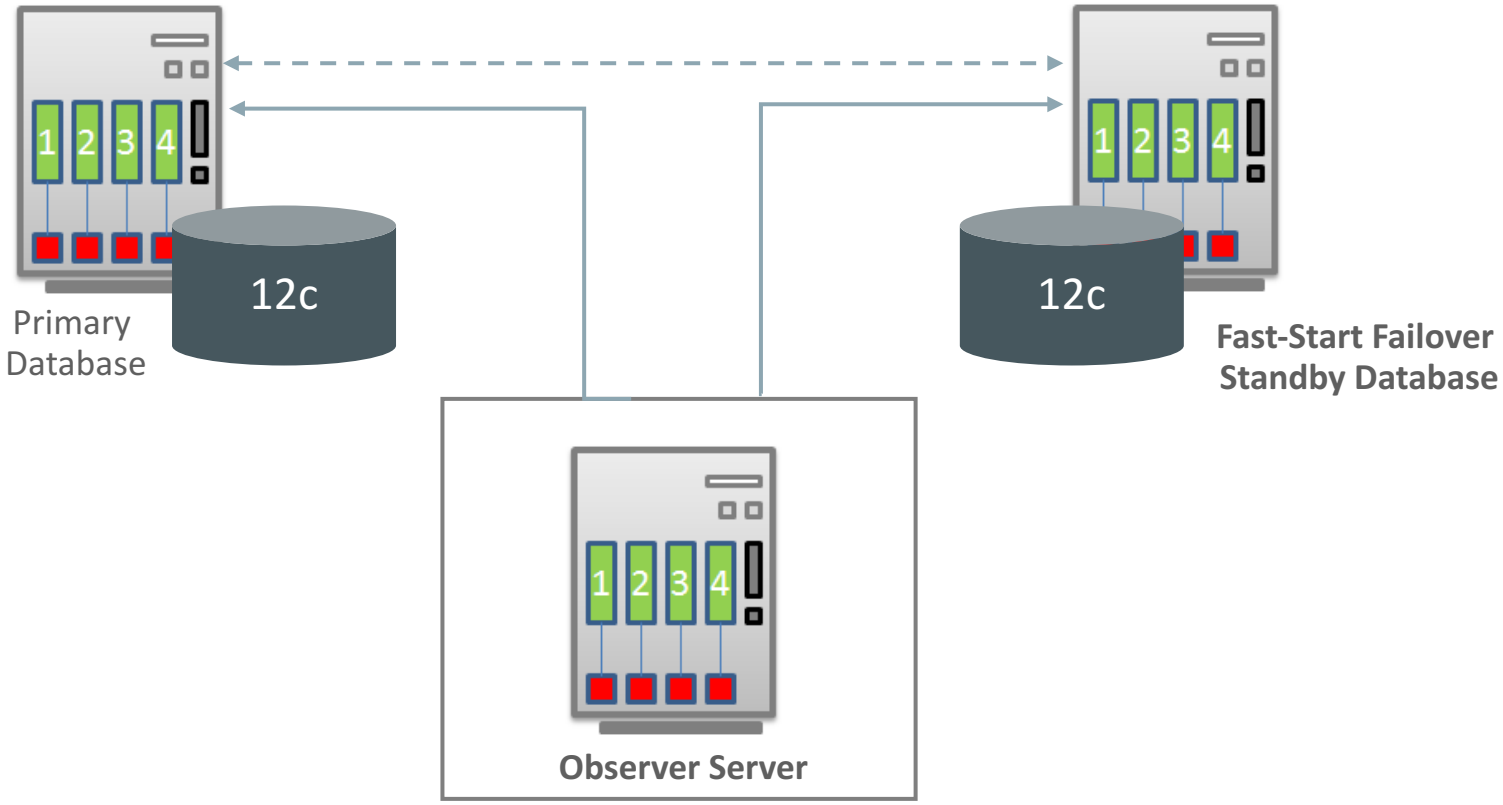
```
DGMGRL> show fast_start failover

  Threshold:           45 seconds
  Target:              (none)
  Observer:            (none)
  Lag Limit:           45 seconds
  Shutdown Primary:    FALSE
  Auto-reinstate:      TRUE
  Observer Reconnect: (none)
  Observer Override:   FALSE

Configurable Failover Conditions
  Health Conditions:
    Corrupted Controlfile          YES
    Corrupted Dictionary           YES
    Inaccessible Logfile            NO
    Stuck Archiver                  NO
    Datafile Offline               YES
```

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Enable Fast-Start Failover




```
DGMGRL> ENABLE FAST_START FAILOVER;
Enabled.
DGMGRL>
```

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Verify the Configuration

```
DGMGRL> show fast_start failover;


Fast-Start Failover: ENABLED


  Threshold:           45 seconds
  Target:              DXDB
  Observer:            (none)
  Lag Limit:           45 seconds
  Shutdown Primary:    FALSE
  Auto-reinstate:      TRUE
  Observer Reconnect:  (none)
  Observer Override:   FALSE
```

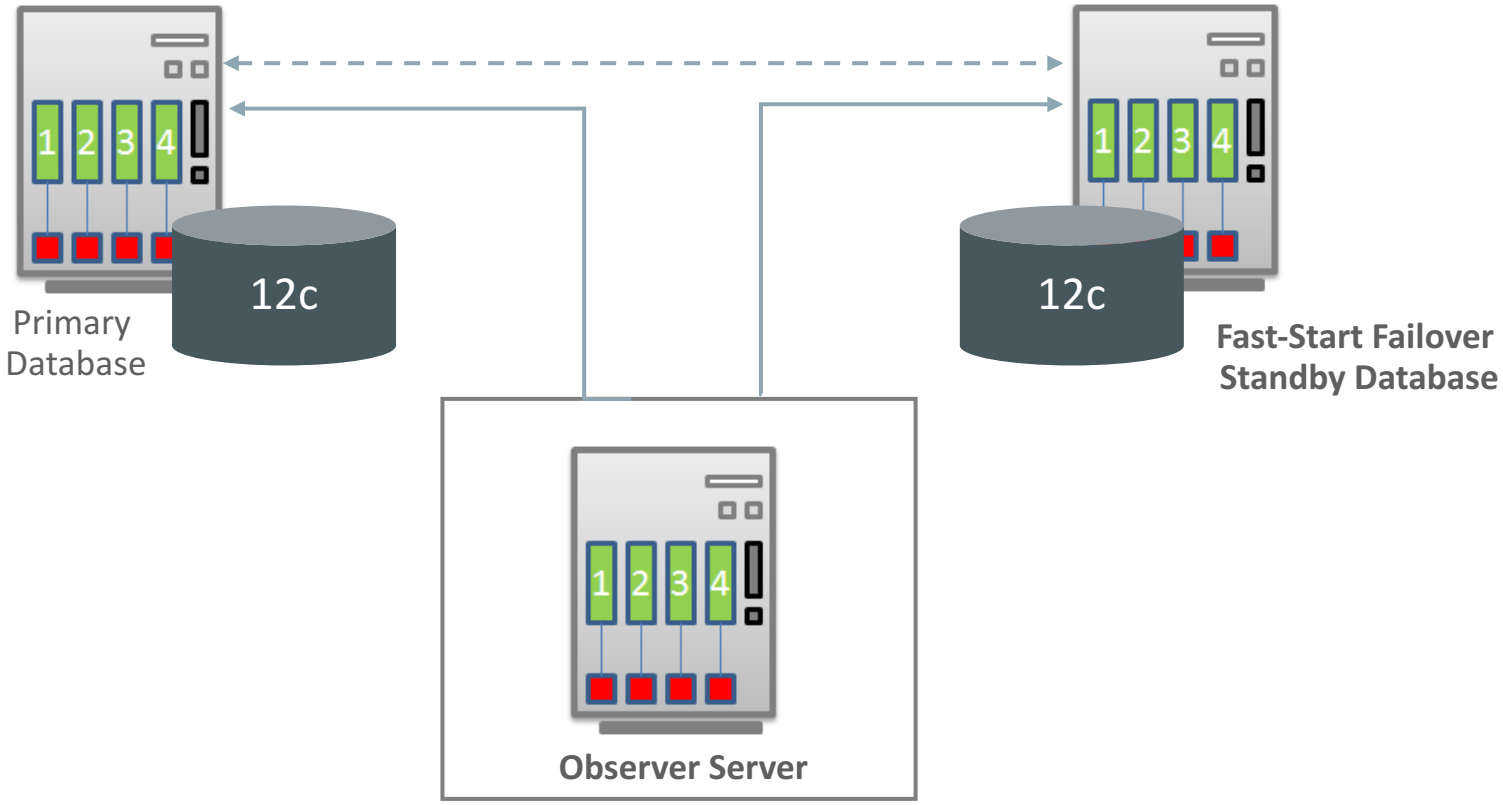
```
Configurable Failover Conditions
  Health Conditions:
    Corrupted Controlfile          YES
    Corrupted Dictionary           YES
    Inaccessible Logfile           YES
    Stuck Archiver                  NO
    Datafile Offline               YES


  Oracle Error Conditions:
    ORA-01578: ORACLE data block corrupted
(file # %s, block # %s)


DGMGRL> start observer;
Observer started
```

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Start the Observer Server




```
DGMGRL> start observer;
Observer started
```

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Test: Fast-Start Failover for "Datafile Offline" Condition

```
DGMGRL> show configuration;

Configuration - DG_CONF

  Protection Mode: MaxPerformance
  Databases:
  DHDB - Primary database
    DXDB - (*) Physical standby database


Fast-Start Failover: ENABLED

Configuration Status:
SUCCESS

DGMGRL>
```

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Trigger Fast-Start Failover by shutdown the Primary Database

```
SQL> conn / as sysdba
Connected.
SQL> shutdown abort;
ORACLE instance shut down.
SQL>
```

```
DGMGRL> start observer;
Observer started


04:00:52.48  Saturday, November 11, 2017
Initiating Fast-Start Failover to database "DXDB"...
Performing failover NOW, please wait...
Failover succeeded, new primary is "DXDB"
04:00:54.67  Saturday, November 11, 2017
```

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Display current Data Guard Status

```
DGMGRL> connect sys/oracle@DXDB
Connected as SYSDBA.
DGMGRL> show configuration;

Configuration - DG_CONF

  Protection Mode: MaxPerformance
  Databases:
  DXDB - Primary database
    Warning: ORA-16829: fast-start failover configuration is lagging

    DHDB - (*) Physical standby database (disabled)
      ORA-16661: the standby database needs to be reinstated

Fast-Start Failover: ENABLED

Configuration Status:
WARNING
DGMGRL>
```

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Automatic Re-Instatement

```
SQL> connect / as sysdba
Connected to an idle instance.
SQL> startup mount
ORACLE instance started.

Total System Global Area  839282688 bytes
Fixed Size     2293928 bytes
Variable Size   620760920 bytes
Database Buffers   213909504 bytes
Redo Buffers     2318336 bytes
Database mounted.
SQL>
```

```
04:06:16.75  Saturday, November 11, 2017
Initiating reinstatement for database "DHDB"...
Reinstating database "DHDB", please wait...
Reinstatement of database "DHDB" succeeded
04:06:29.42  Saturday, November 11, 2017
```

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Verify Fast-Start Failover Occurred

```
DGMGRL> show configuration;

Configuration - DG_CONF

  Protection Mode: MaxPerformance
  Databases:
  DXDB - Primary database
    DHDB - (*) Physical standby database

Fast-Start Failover: ENABLED

Configuration Status:
SUCCESS


DGMGRL>
```

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Show current configuration

```
DGMGRL> stop observer;
Done.
DGMGRL> disable fast_start failover;
Disabled.
DGMGRL> show configuration

Configuration - DG_CONF

  Protection Mode: MaxPerformance
  Databases:
  DXDB - Primary database
    DHDB - Physical standby database

Fast-Start Failover: DISABLED

Configuration Status:
SUCCESS

DGMGRL>
```

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Switchover to DHDB

```
DGMGRL> switchover to 'DHDB';
Performing switchover NOW, please wait...
New primary database "DHDB" is opening...
Operation requires startup of instance "DXDB" on database "DXDB"
Starting instance "DXDB"...
ORACLE instance started.
Database mounted.
Database opened.
Switchover succeeded, new primary is "DHDB"
DGMGRL>
```

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Show Current Data Guard Configuration

```
DGMGRL> show configuration;

Configuration - DG_CONF


  Protection Mode: MaxPerformance
  Databases:
  DHDB - Primary database
    DXDB - Physical standby database

Fast-Start Failover: DISABLED

Configuration Status:
SUCCESS


DGMGRL>
```

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Fast-Start Failover Management

- Show FAST-START Failover Properties
- Viewing FAST-START Failover Information
- Determining reason for FAST-START Failover
- FAST-START Failover Status
- Enable/Disable FAST-START Failover
- Start/Stop the Observer

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Show FAST-START Failover Properties

```
DGMGRL> show fast_start failover

Fast-Start Failover: DISABLED

  Threshold:           45 seconds
  Target:              (none)
  Observer:            (none)
  Lag Limit:           45 seconds
  Shutdown Primary:    FALSE
  Auto-reinstate:      TRUE
  Observer Reconnect:  (none)
  Observer Override:   FALSE

  Oracle Error Conditions:
    ORA-01578: ORACLE data block corrupted (file # %s, block # %s)

DGMGRL>
```

```
Configurable Failover Conditions
  Health Conditions:
    Corrupted Controlfile          YES
    Corrupted Dictionary           YES
    Inaccessible Logfile           YES
    Stuck Archiver                  NO
    Datafile Offline               YES
```

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Viewing FAST-START Failover Information

```
SELECT  fs_failover_status as STATUS,
        fs_failover_current_target as CURR_TGET,
        fs_failover_threshold as THRESHOLD,
        fs_failover_observer_present as OBS_PRES,
        fs_failover_observer_host as OBS_HOST
FROM v$database;


STATUS                    CURR_TGET T   HRESHOLD      OBS_PRES
------------------------  ---------   ----------    --------
TARGET UNDER LAG LIMIT    pc01sby1    90            YES


OBS_HOST
------------------------
edbvr6p2.us.oracle.com
```

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Reason for FAST-START Failover

```
SQL> SELECT last_failover_time, last_failover_reason
   2 FROM v$fs_failover_stats;

LAST_FAILOVER_TIME          LAST_FAILOVER_REASON
--------------------        --------------------------------
03/02/2010 22:30:12         Primary Disconnected
```

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Enable/Disable FAST-START Failover

```
DGMGRL> enable fast_start failover
Enabled.
DGMGRL>
```

FAST-START Failover

```
DGMGRL> disable fast_start failover
Enabled.
DGMGRL>
```

```
DGMGRL> enable fast_start failover condition "Inaccessible Logfile";
Succeeded.
DGMGRL>
```

```
DGMGRL> disable fast_start failover condition "Inaccessible Logfile";
Succeeded.
DGMGRL>
```

FAST-START Failover Conditions

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Start/Stop the Observer

```
DGMGRL> start observer
Observer started


14:13:09.22  Friday, November 17, 2017
Initiating Fast-Start Failover to database "DXDB"...
Performing failover NOW, please wait...
Failover succeeded, new primary is "DXDB"
14:13:11.36  Friday, November 17, 2017
```

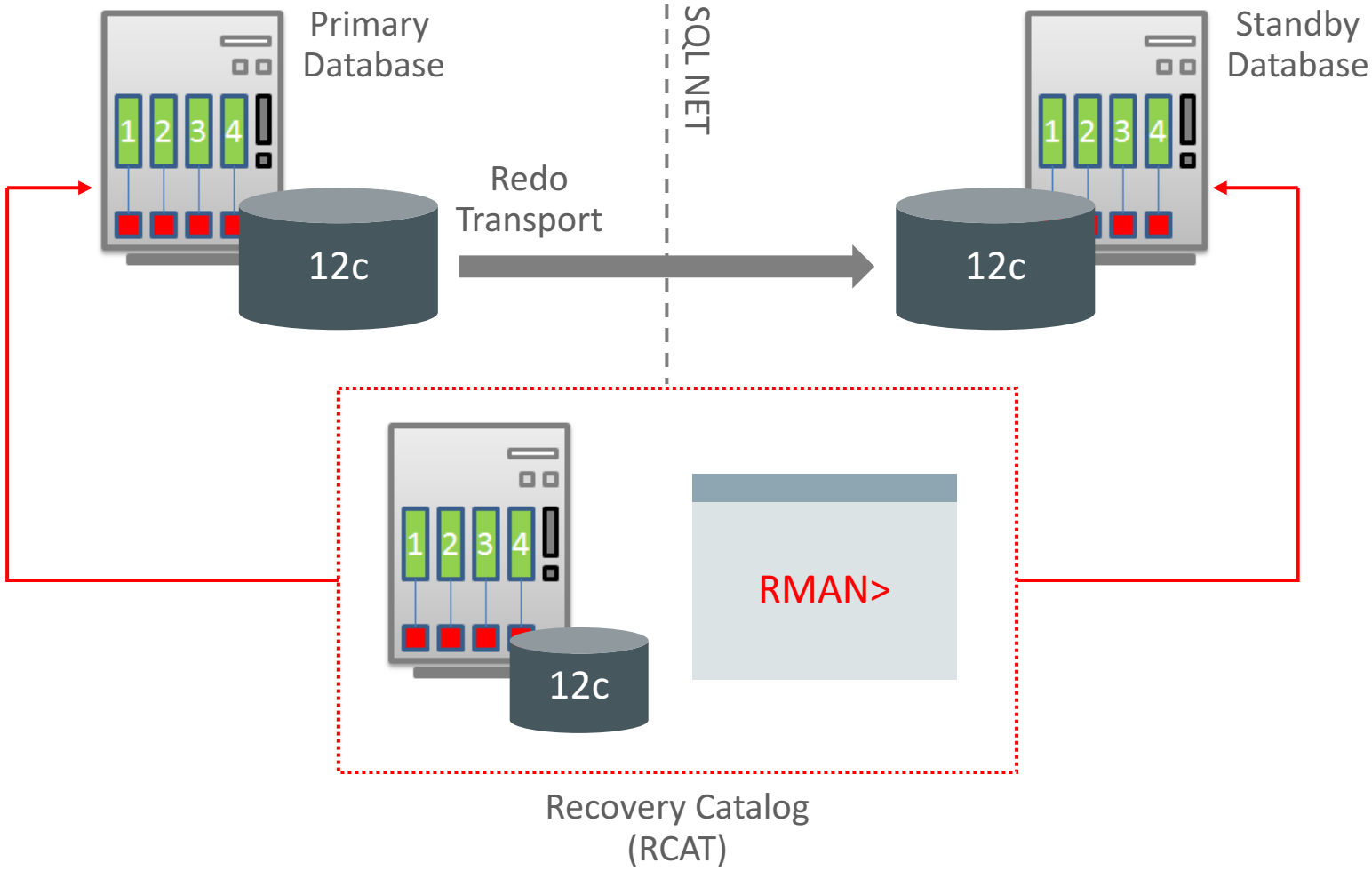
Another Terminal

```
DGMGRL> stop observer
Observer started

DGMGRL>
```

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Backup and Restore Using RMAN



Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Create RMAN Repository Database

```
SQL> conn / as sysdba
Connected to an idle instance.
SQL> startup nomount pfile='$ORACLE_HOME/dbs/initRCAT.ora'
ORACLE instance started.


Total System Global Area   839282688 bytes
Fixed Size                 2293928   bytes
Variable Size              591400792 bytes
Database Buffers           243269632 bytes
Redo Buffers               2318336   bytes
SQL>
SQL> CREATE DATABASE;

Database created.

SQL>
```

```
SQL> @$ORACLE_HOME/rdbms/admin/catalog.sql
SQL> @$ORACLE_HOME/rdbms/admin/catproc.sql
```

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Create RMAN Catalog

```
[oracle@dgobserver ~]$ rman

Recovery Manager: Release 12.1.0.1.0 - Production on Fri Nov 17 23:12:08 2017

Copyright (c) 1982, 2013, Oracle and/or its affiliates.  All rights reserved.

RMAN> connect catalog rcatmgr/oracle_4U

connected to recovery catalog database

RMAN> connect target sys/oracle@DHDB
connected to target database: DHDB (DBID=2763595033)

RMAN>
```

```
RMAN> create catalog;

recovery catalog created

RMAN>
```

```
RMAN> register database;

database registered in recovery catalog
starting full resync of recovery catalog
full resync complete
```

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Enable Database Block Change Trucking Feature

```
[oracle@dgobserver ~]$ rman

Recovery Manager: Release 12.1.0.1.0 - Production on Fri Nov 17 23:28:31 2017

Copyright (c) 1982, 2013, Oracle and/or its affiliates.  All rights reserved.

RMAN> connect catalog rcatmgr/oracle_4U

connected to recovery catalog database

RMAN> connect target sys/oracle@DXDB

connected to target database: DHDB (DBID=2763595033)

RMAN> alter database enable block change tracking;

Statement processed

RMAN>
```

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Perform Backup on the Standby Database

```
RMAN> create script s1_bck {
2> backup incremental level 1 database;
3> }

created script s1_bck

RMAN>
```

```
RMAN> run {execute script s1_bck;}

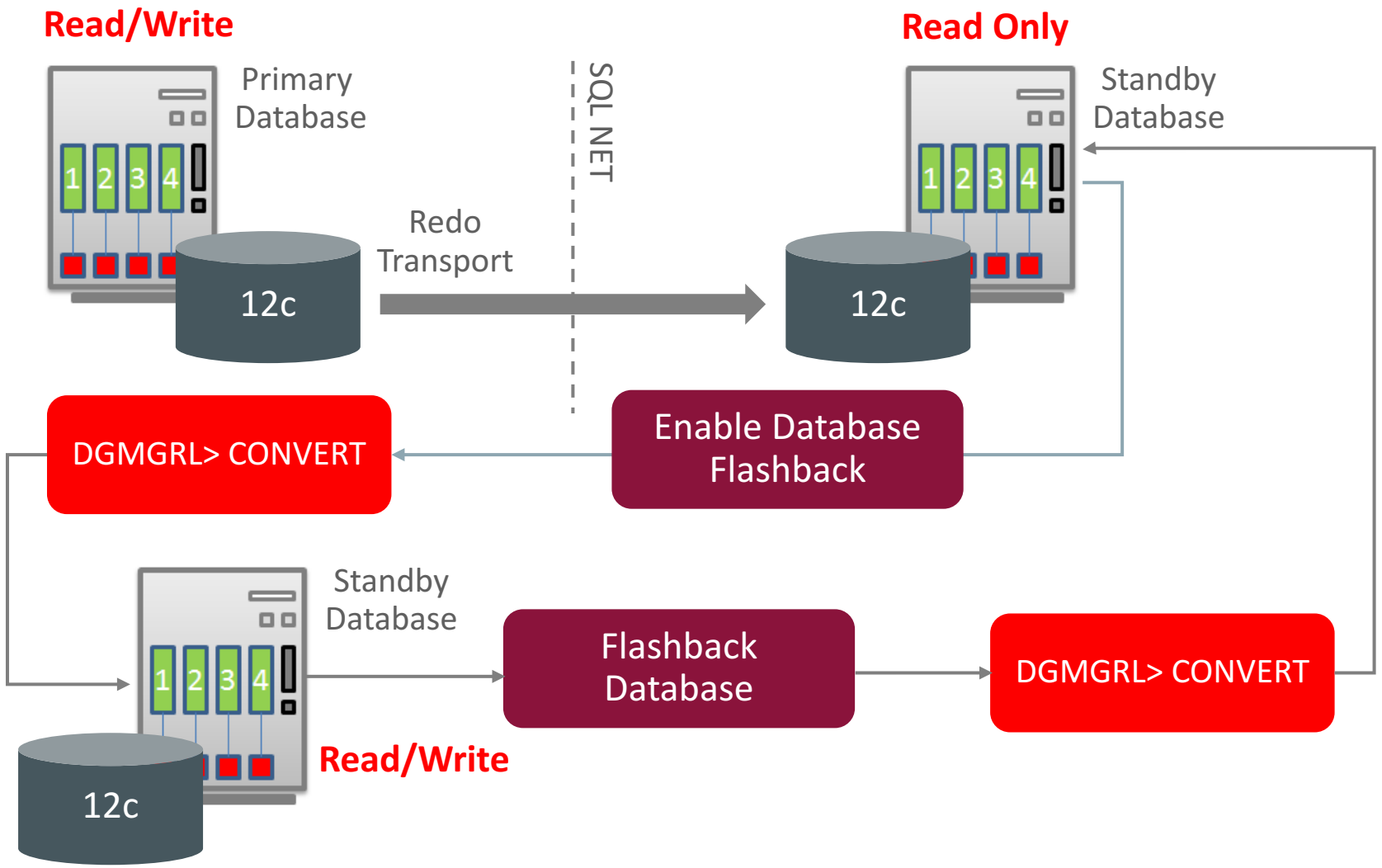
executing script: s1_bck

Starting backup at 17-NOV-17
using channel ORA_DISK_1
…
Finished Control File and SPFILE Autobackup at 17-NOV-17

RMAN>
```

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

What is Snapshot Standby Database



Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Creating Snapshot Standby Database

```
DGMGRL> CONVERT DATABASE 'DXDB' TO SNAPSHOT STANDBY;
Converting database "DXDB" to a Snapshot Standby database, please wait...
Database "DXDB" converted successfully

DGMGRL> show configuration

Configuration - DG_CONF

  Protection Mode: MaxPerformance
  Databases:
  DHDB - Primary database
    DXDB - Snapshot standby database

Fast-Start Failover: DISABLED

Configuration Status:
SUCCESS
DGMGRL>
```

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Disable Snapshot Standby Database

```
DGMGRL> CONVERT DATABASE 'DXDB' TO PHYSICAL STANDBY;
Converting database "DXDB" to a Physical Standby database, please wait...
Database "DXDB" converted successfully
DGMGRL> show configuration;

Configuration - DG_CONF

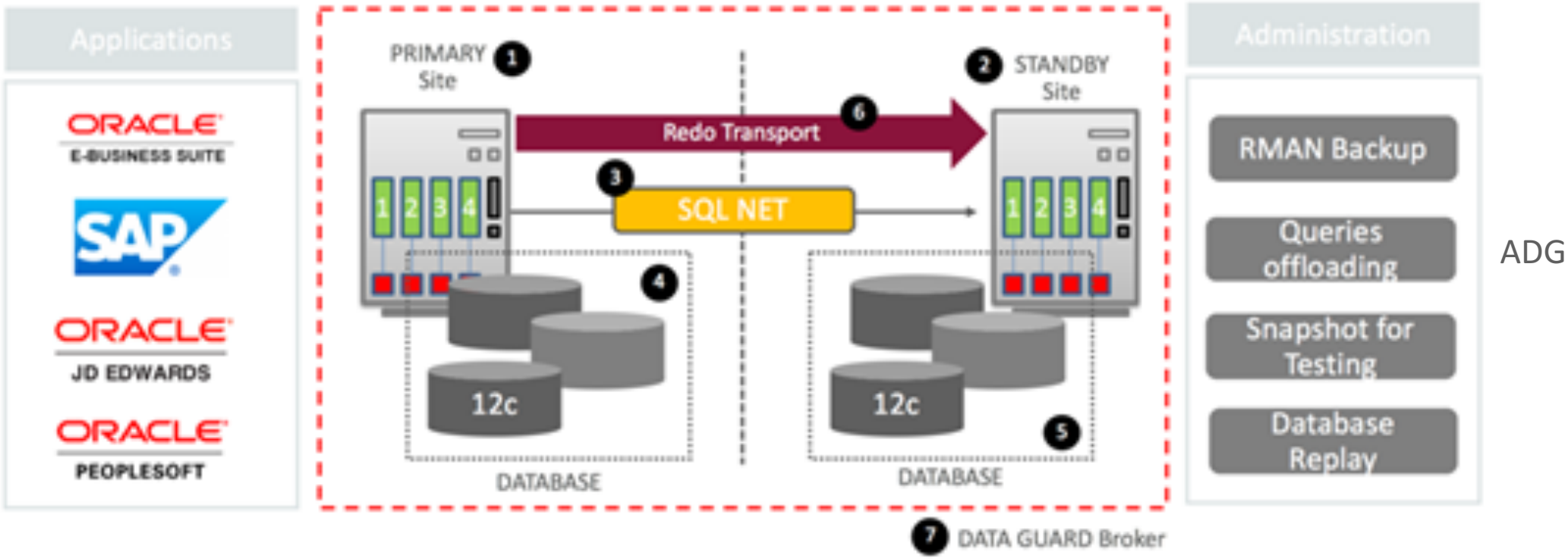
  Protection Mode: MaxPerformance
  Databases:
  DHDB - Primary database
    DXDB - Physical standby database
Fast-Start Failover: DISABLED

Configuration Status:
SUCCESS

DGMGRL>
```

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Oracle Active Data Guard



ADG

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Enable Real-Time Reporting

```
DGMGRL> edit database 'DXDB' set state='apply-on';
Succeeded.
DGMGRL> show database 'DXDB';

Database - DXDB
  Role:               PHYSICAL STANDBY
  Intended State:     APPLY-ON
  Transport Lag:      0 seconds (computed 1 second ago)
  Apply Lag:          0 seconds (computed 1 second ago)
  Apply Rate:         (unknown)
  Real Time Query:    ON
  Instance(s):
    DXDB

Database Status:
SUCCESS

DGMGRL>
```

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Disable Real-Time Reporting

```
DGMGRL> edit database 'DXDB' set state='apply-off';
Succeeded.
DGMGRL> show database 'DXDB';

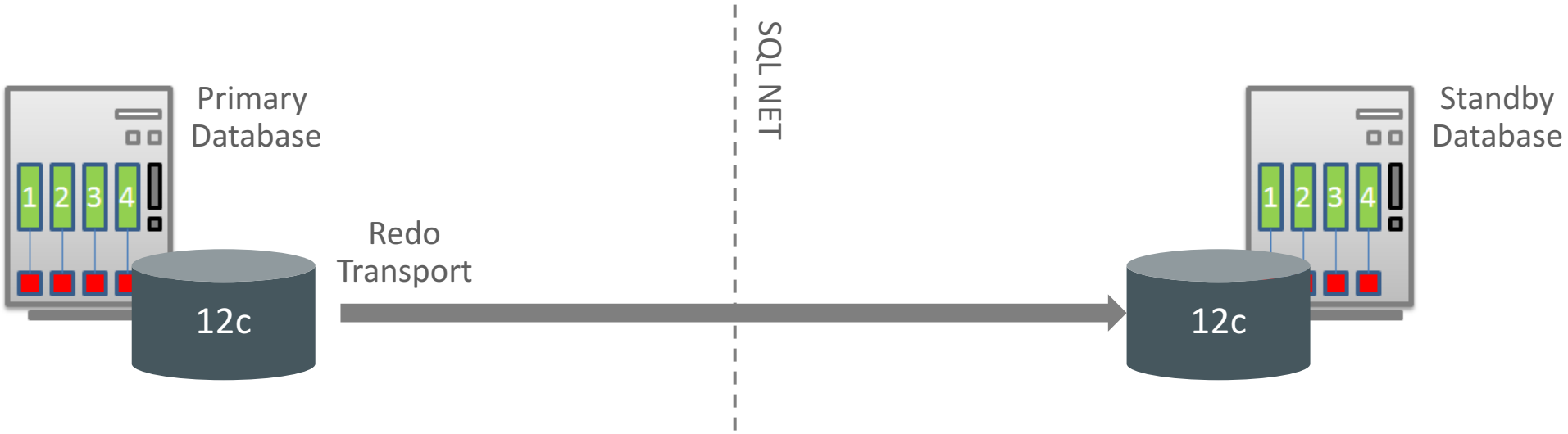
Database - DXDB
  Role:               PHYSICAL STANDBY
  Intended State:     APPLY-OFF
  Transport Lag:      0 seconds (computed 1 second ago)
  Apply Lag:          8 seconds (computed 1 second ago)
  Apply Rate:         (unknown)
  Real Time Query:    OFF
  Instance(s):
    DXDB

Database Status:
SUCCESS

DGMGRL>
```

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Compression Redo-Transport




```
DGMGRL> EDIT DATABASE 'DHDB'
> SET PROPERTY 'RedoCompression'='ENABLE';
Property "RedoCompression" updated
DGMGRL> EDIT DATABASE 'DXDB'
> SET PROPERTY 'RedoCompression'='ENABLE';
Property "RedoCompression" updated
DGMGRL>
```

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Monitoring Data Guard Performance

- Verify Data Guard Configuration is SUCCESS
- Verify Database properties are correct
- Display Data Guard health check report
- Monitor Redo Apply
- Monitor Redo Apply Lag
- Monitor Block Change Tracking

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Verify Data Guard Configuration is SUCCESS

```
DGMGRL> show database 'DXDB';

Database - DXDB

  Role:               PHYSICAL STANDBY
  Intended State:     APPLY-ON
  Transport Lag:      0 seconds (computed 1 second ago)
  Apply Lag:          0 seconds (computed 1 second ago)
  Apply Rate:         0 Byte/s
  Real Time Query:    ON
  Instance(s):
    DXDB

Database Status:
SUCCESS

DGMGRL>
```

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Verify Database properties are correct

```
DGMGRL> show database verbose 'DXDB';

Database - DXDB
  Role:               PHYSICAL STANDBY
  Intended State:     APPLY-ON
  Transport Lag:      0 seconds (computed 0 seconds ago)
  Apply Lag:          0 seconds (computed 0 seconds ago)
  Apply Rate:         0 Byte/s
  Real Time Query:    ON
  Instance(s):
    DXDB
  Properties:
    DGConnectIdentifier                = 'DXDB'
    …
Database Status:
SUCCESS

DGMGRL>
```

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Display Data Guard health check report

```
DGMGRL> SHOW DATABASE 'DHDB' 'StatusReport';
STATUS REPORT
       INSTANCE_NAME   SEVERITY ERROR_TEXT
DGMGRL> SHOW DATABASE 'DHDB' 'LogXptStatus';
LOG TRANSPORT STATUS
PRIMARY_INSTANCE_NAME STANDBY_DATABASE_NAME                STATUS
                DHDB                    DXDB
DGMGRL> SHOW DATABASE 'DHDB' 'InconsistentProperties';
INCONSISTENT PROPERTIES
   INSTANCE_NAME         PROPERTY_NAME          MEMORY_VALUE          SPFILE_VALUE
        BROKER_VALUE
DGMGRL> SHOW DATABASE 'DHDB' 'InconsistentLogXptProps';
INCONSISTENT LOG TRANSPORT PROPERTIES
   INSTANCE_NAME          STANDBY_NAME         PROPERTY_NAME          MEMORY_VALUE
        BROKER_VALUE

DGMGRL>
```

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Monitor Redo Apply

```
SQL> SELECT process, status, group#, thread#
  2  FROM v$managed_standby
  3  order by process, group#, thread#;

PROCESS   STATUS        GROUP#   THREAD#
--------- ------------ ---------- ----------
ARCH   CLOSING      4 1
ARCH   CLOSING      5 1
ARCH   CLOSING      5 1
ARCH   CONNECTED    N/A 0
MRP0   APPLYING_LOG N/A 1
RFS   IDLE        2 1
RFS   IDLE        N/A 0
RFS   IDLE        N/A 0

8 rows selected.

SQL>
```

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Monitor Redo Apply Lag

```
SQL> SELECT name, value, datum_time, time_computed
  2  FROM v$dataguard_stats
  3  WHERE name like 'apply lag';


NAME
--------------------------------
VALUE
----------------------------------------------------------------
DATUM_TIME        TIME_COMPUTED
------------------------------ ------------------------------
apply lag
+00 00:00:00
11/18/2017 02:34:59       11/18/2017 02:35:01


SQL>
```

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Monitor Block Change Tracking

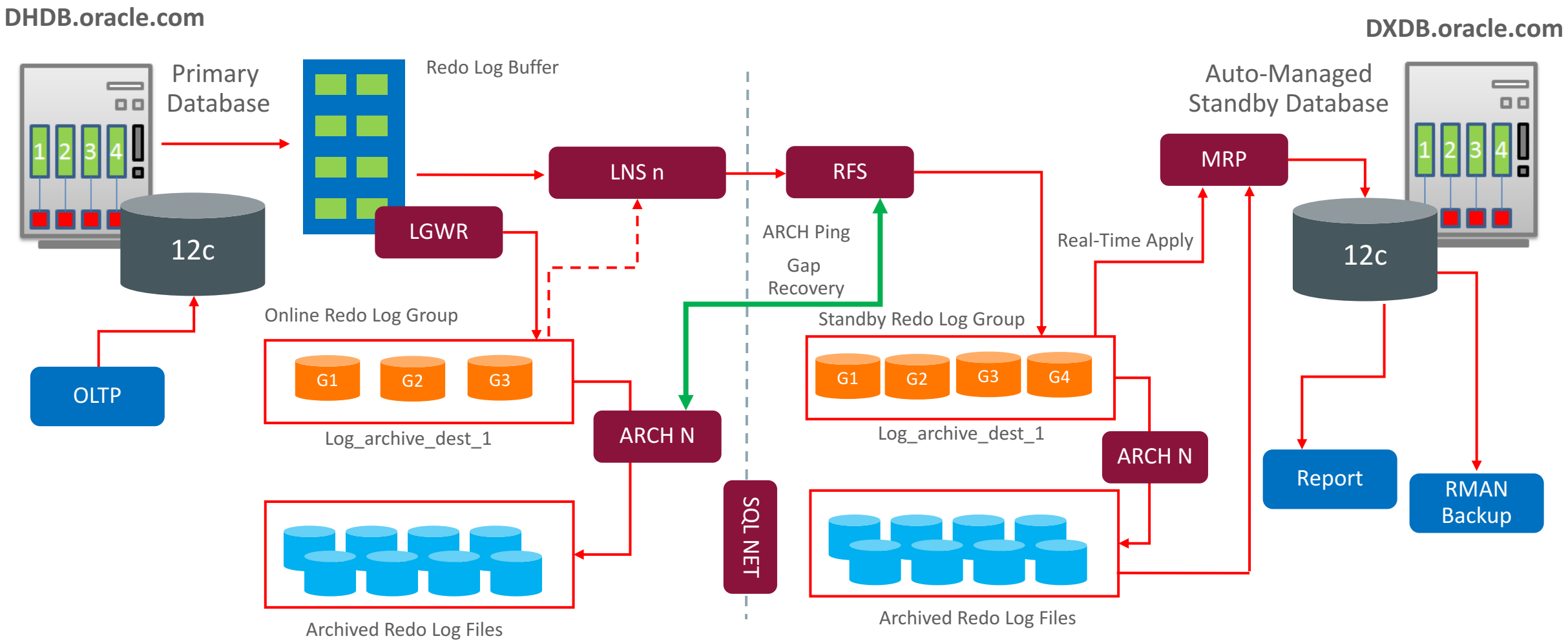
```
SQL> SELECT file#, avg(datafile_blocks),
  2  avg(blocks_read),
  3  avg(blocks_read/datafile_blocks)
  4  * 100 AS PCT_READ_FOR_BACKUP,
  5  avg(blocks)
  6  FROM   v$backup_datafile
  7  WHERE  used_change_tracking = 'YES'
  8  AND    incremental_level > 0
  9  GROUP  BY file#;

     FILE# AVG(DATAFILE_BLOCKS) AVG(BLOCKS_READ) PCT_READ_FOR_BACKUP AVG(BLOCKS)
---------- -------------------- ---------------- -------------------- ------
         1               101120             9891           9.78144778           5087
         6                51360             1233           2.40070093            414
         4                18560             4171           22.4730603           4110
         3               102400            18459           18.0263672          10782

SQL>
```

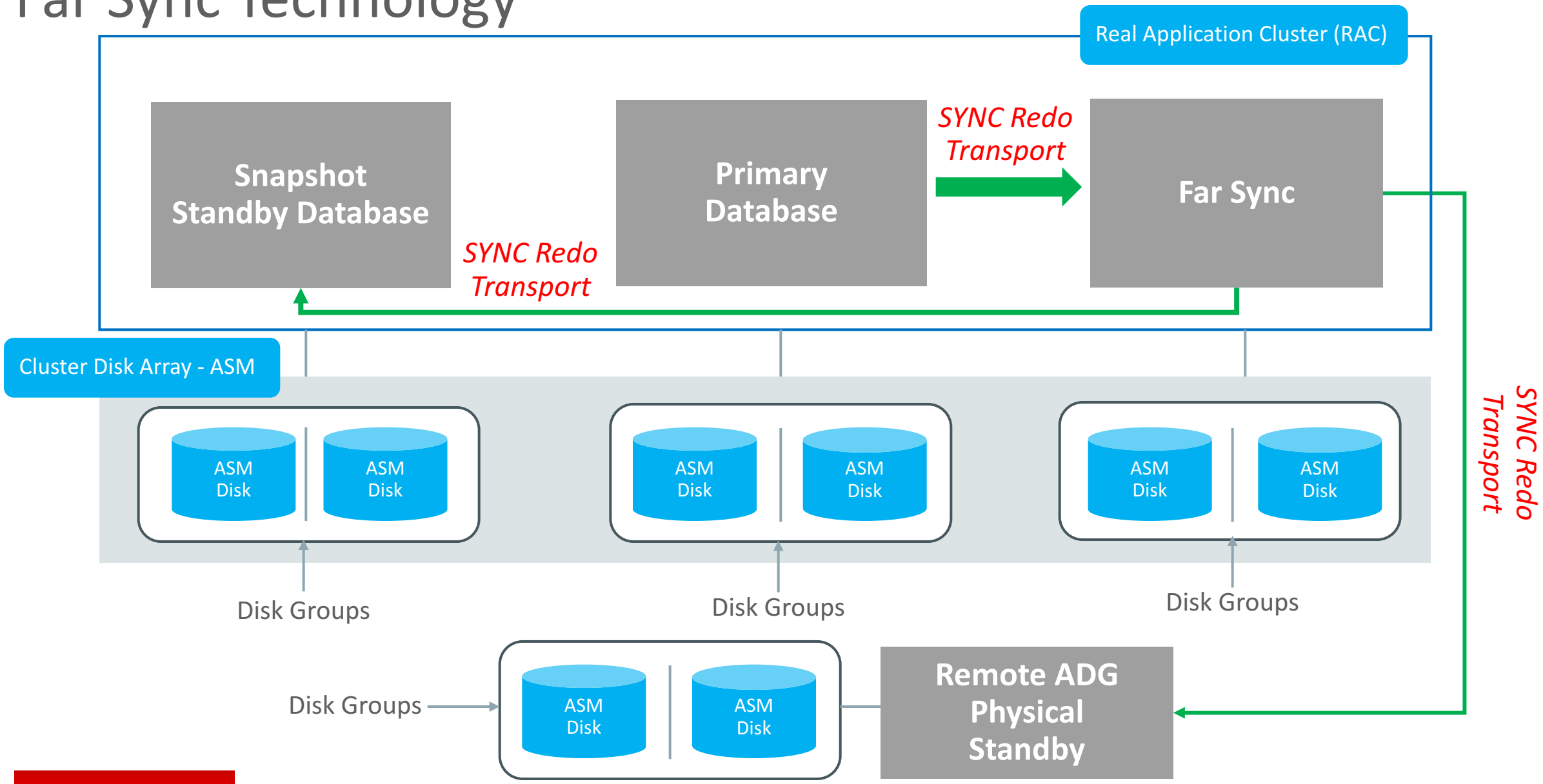
Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Automatic Gap Resolution



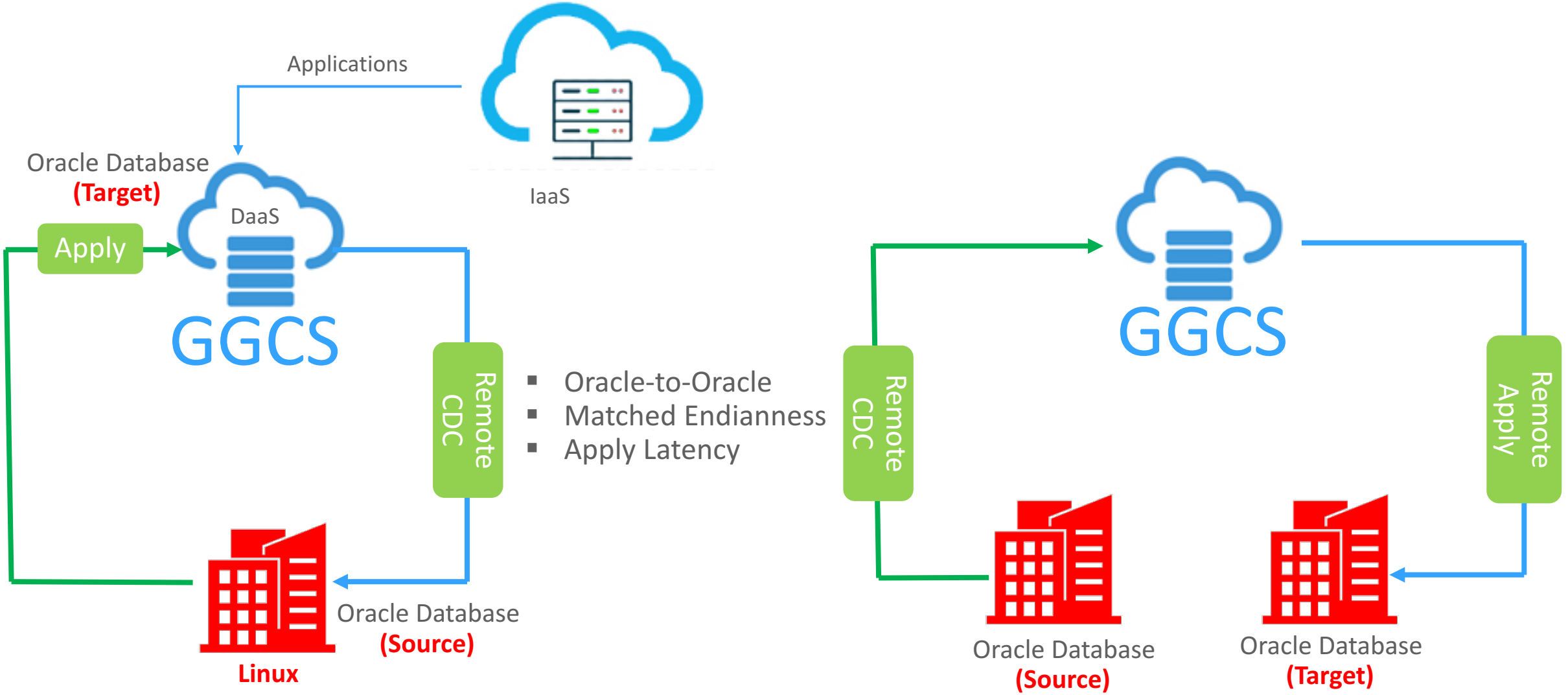
Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Far Sync Technology



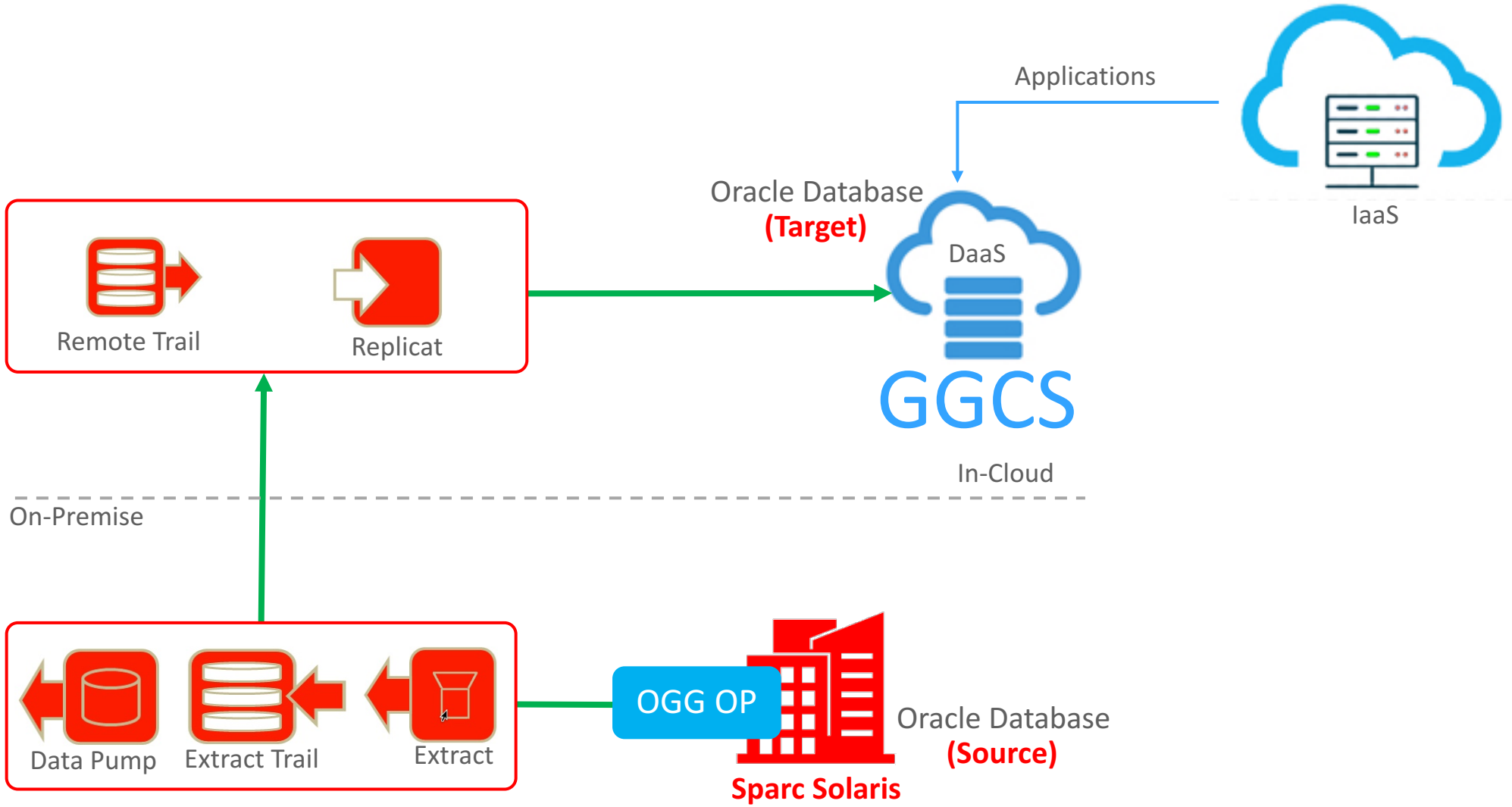
Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Disaster Recovery using GGCS *Remote Capture and Apply*



Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

On-Premise-to-In-Cloud GoldenGate



Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Quiz 1

- What are the types of standby database
  - ❑ Physical Standby database
  - ❑ Logical Standby database
  - ❑ Snapshot Standby database
  - ❑ Hybrid Standby database
- What process used by the DG Broker
  - ○ LGWR
  - ○ DBWR
  - ○ DMON
  - ○ ARCH
- How many standby database configurable
  - ○ 1
  - ○ 10
  - ○ 16
  - ○ 30
- Data Guard come pre-configured
  - ○ TRUE
  - ○ FALSE

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Quiz 2

- Which of the following allowed protection mode?
  - ❑ Maximum Performance
  - ❑ Maximum Availability
  - ❑ Maximum Protection
  - ❑ Maximum Throughput

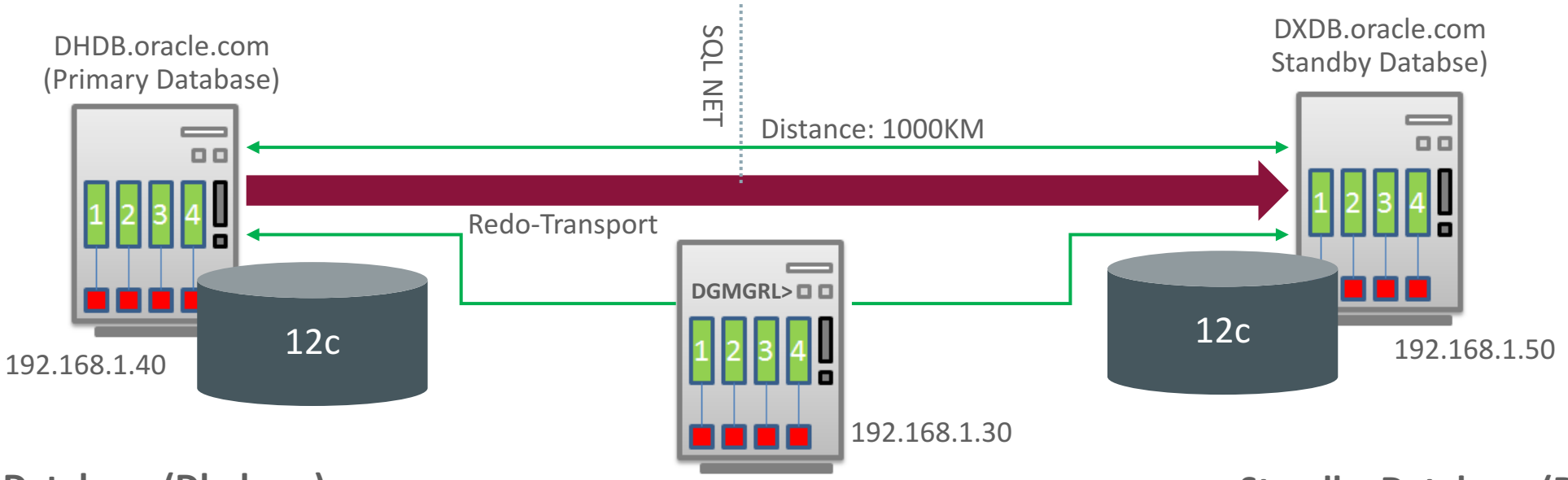
- Standby online redo log file can not be created on the primary database?
  - ○ TRUE
  - ○ FALSE

- What is the Default protection mode?
  - ○ Maximum Performance
  - ○ Maximum Availability
  - ○ Maximum Protection

- How many Data Guard broker configuration files are created?
  - ○ 1
  - ○ 2
  - ○ 3
  - ○ 4

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Demonstration



Primary Database (Dhahran)

Name	DHDB.oracle.com
Version	12.1
RAC	No
ASM	No
SYS	oracle_4U
SYSTEM	oracle_4U
Home	/u01/app/oracle/product/12.1.0/dbhome_1

Standby Database (Dubai)

Name	DXDB.oracle.com
Version	12.1
RAC	No
ASM	No
SYS	oracle_4U
SYSTEM	oracle_4U
Home	/u01/app/oracle/product/12.1.0/dbhome_1

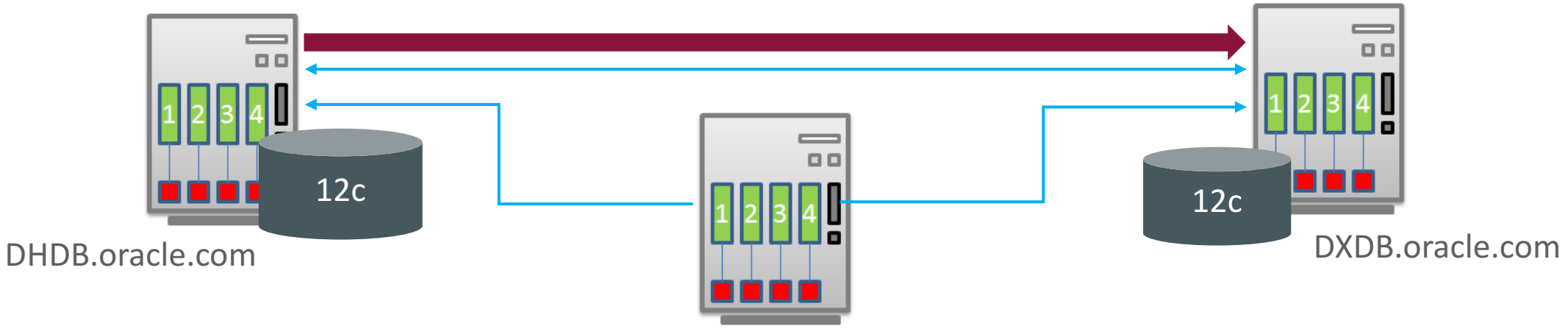
Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Troubleshooting

Error	Resolution
ORA-16775: target standby database in broker operation has potential data loss when performing SWITCHOVER.	Perform the command: SQL> alter system switch logfile; to rotate across all online redo log groups
ORA-01153: an incompatible media recovery is active, when performing SQL>alter database flashback on;	Run the commands: SQL> alter database recover managed standby database cancel; SQL> alter database flashback on; SQL> alter database recover managed standby database disconnect from session;
Error: ORA-16619: condition for fast-start failover is already enabled	Do not attempt to enable default enabled conditions
Warning: ORA-16857: standby disconnected from redo source for longer than specified threshold	Make sure the listener is started, stop and re-start the listener. Reissue the command after waiting for 1min.

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Hands-On Lab Configuration



Primary Database (Dhahran)

Name	DHDB.oracle.com
Version	12.1
RAC	No
ASM	No

root	**oracle**
oracle	oracle
SYS	oracle_4U
SYSTEM	oracle_4U

Observer Server

Name	
Version	
RAC	
ASM	

root	
oracle	
SYS	
SYSTEM	

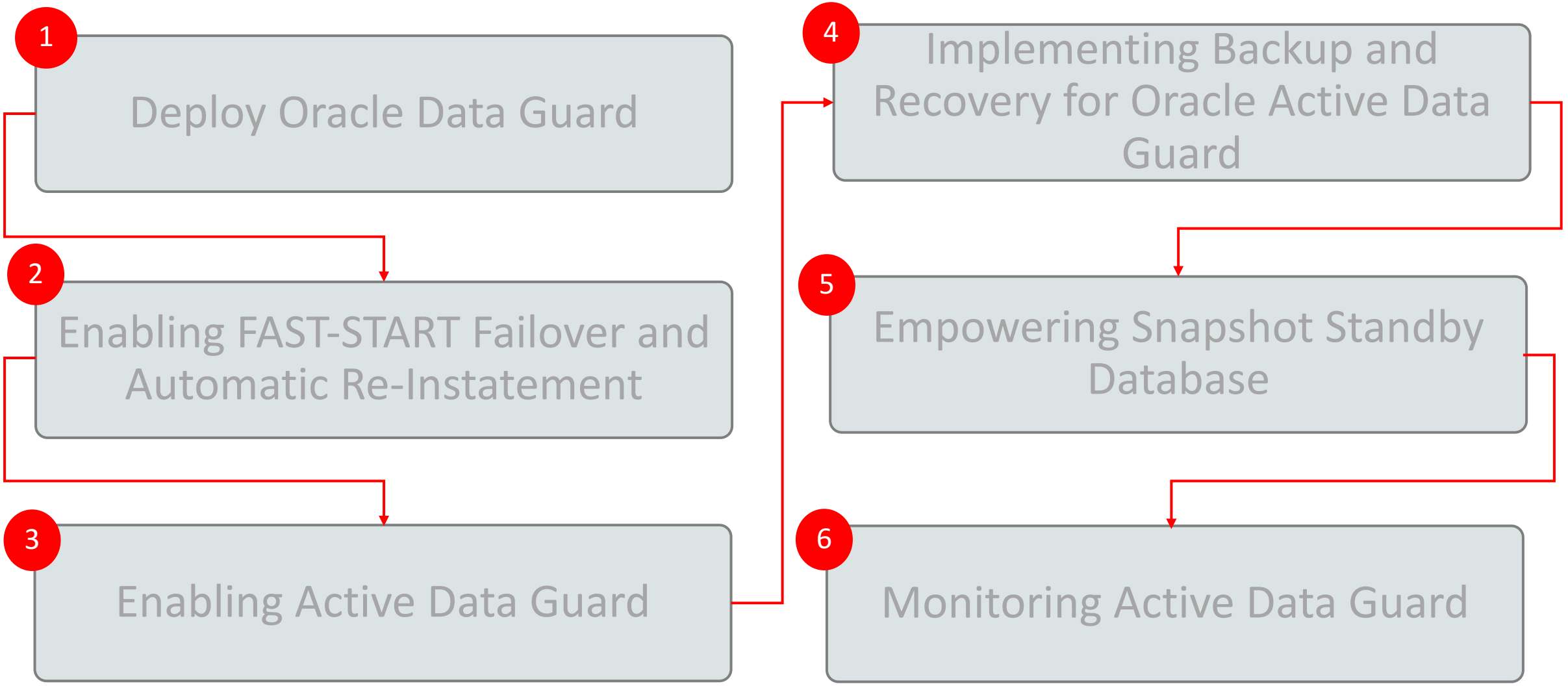
Standby Database (Dubai)

Name	**DXDB.oracle.com**
Version	12.1
RAC	No
ASM	No

root	**oracle**
oracle	oracle
SYS	oracle_4U
SYSTEM	oracle_4U

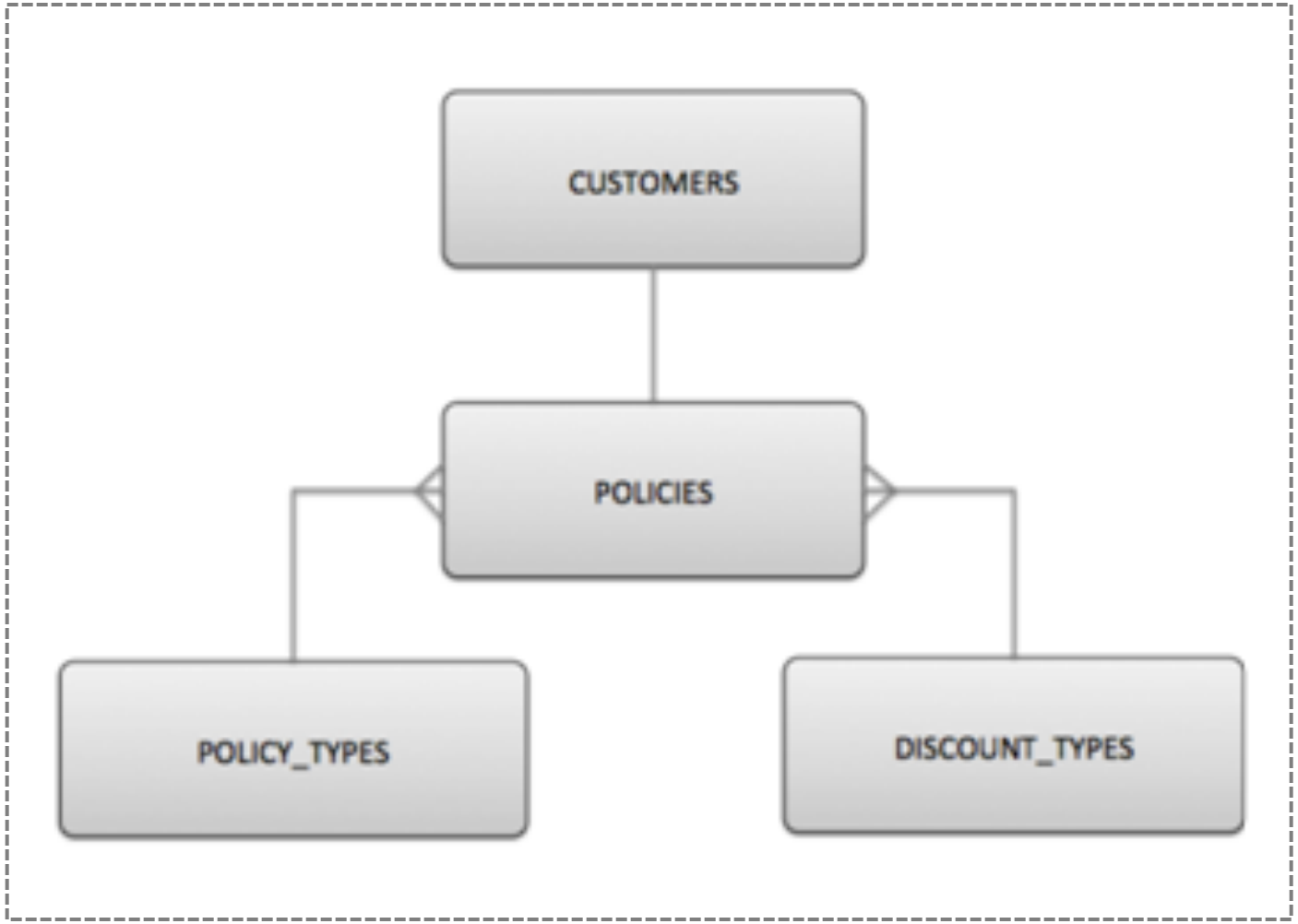
Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Hands-On Lab Sections



Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

HOL Database and Schema



DATABASE	DHDB
SCHEMA	OSM_REPAPI
PASSWORD	oracle

TABLES

- CUSTOMERS
- POLICIES
- POLICY_TYPES
- DISCOUNT_TYPES

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Working with Workload Simulator

```
SQL> desc trans_load
PROCEDURE trans_load
Argument Name Type In/Out Default?
 ------------------------ ------------------ ------ --------
 PLOAD_LEVEL     NUMBER IN     DEFAULT
 PSTARTFROM     NUMBER IN     DEFAULT

SQL> desc call_trans_load
PROCEDURE call_trans_load
Argument Name Type In/Out Default?
 ------------------------ ------------------ ------ --------
 N       NUMBER IN    DEFAULT
 S       NUMBER IN     DEFAULT

SQL>
```

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Invoke Workload Simulator

```
SQL> exec call_trans_load;
```

Run the workload simulator using the default values. The actual parameters uses the default values; N=500 and S=5.

```
SQL> exec call_trans_load(2000);
```

Run the workload simulator using N=2000, this will sustain transactions load of approximately 4hrs.

```
SQL> exec call_trans_load(300000, 1);
```



The following call is an extreme level, use this parameter setting when performing the workshop on 8-16 cores system.

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |
152

Laptop/Desktop Specification

Operating System	MS Windows or MAC OS
HDD Free Space	64GB
Memory	8GB (12GB-16GB preferred)
CPU	i5 or higher
Virtualization Support	Enabled and Tested for Linux on Oracle VirtualBox

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

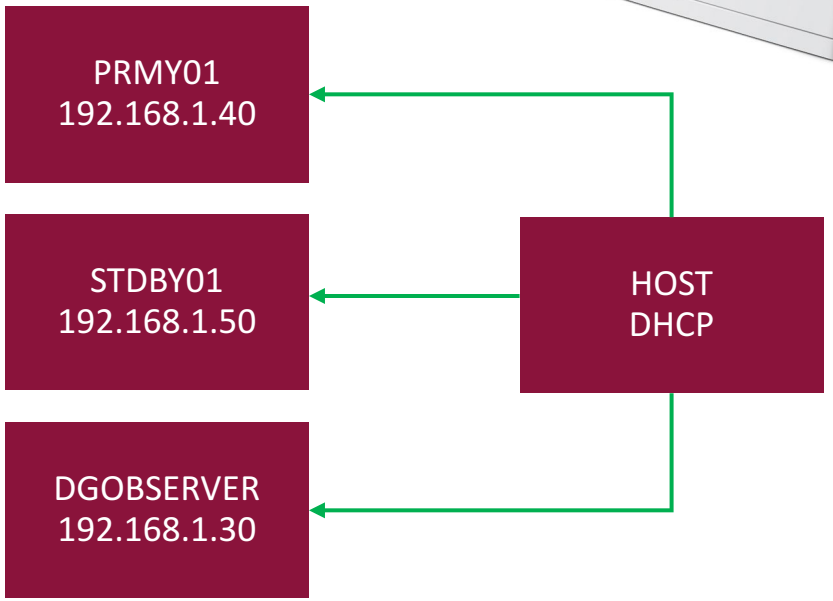
Connectivity: Wi-Fi | Loopback Network

Wi-FI: OracleGoldenGateW

IP: DHCP Address

Password: Welcome10

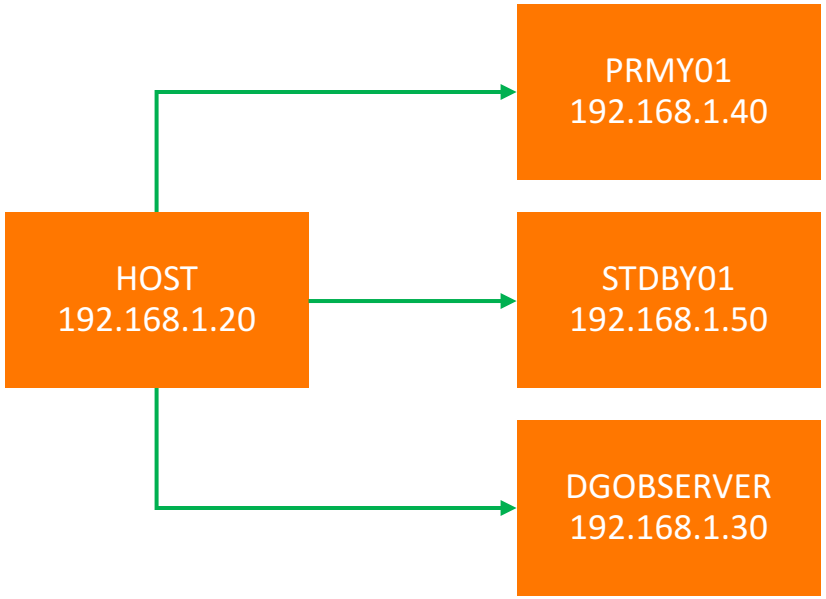




Loopback Adapter

Adapter: Create Loopback Network

Static IP: 192.168.1.20

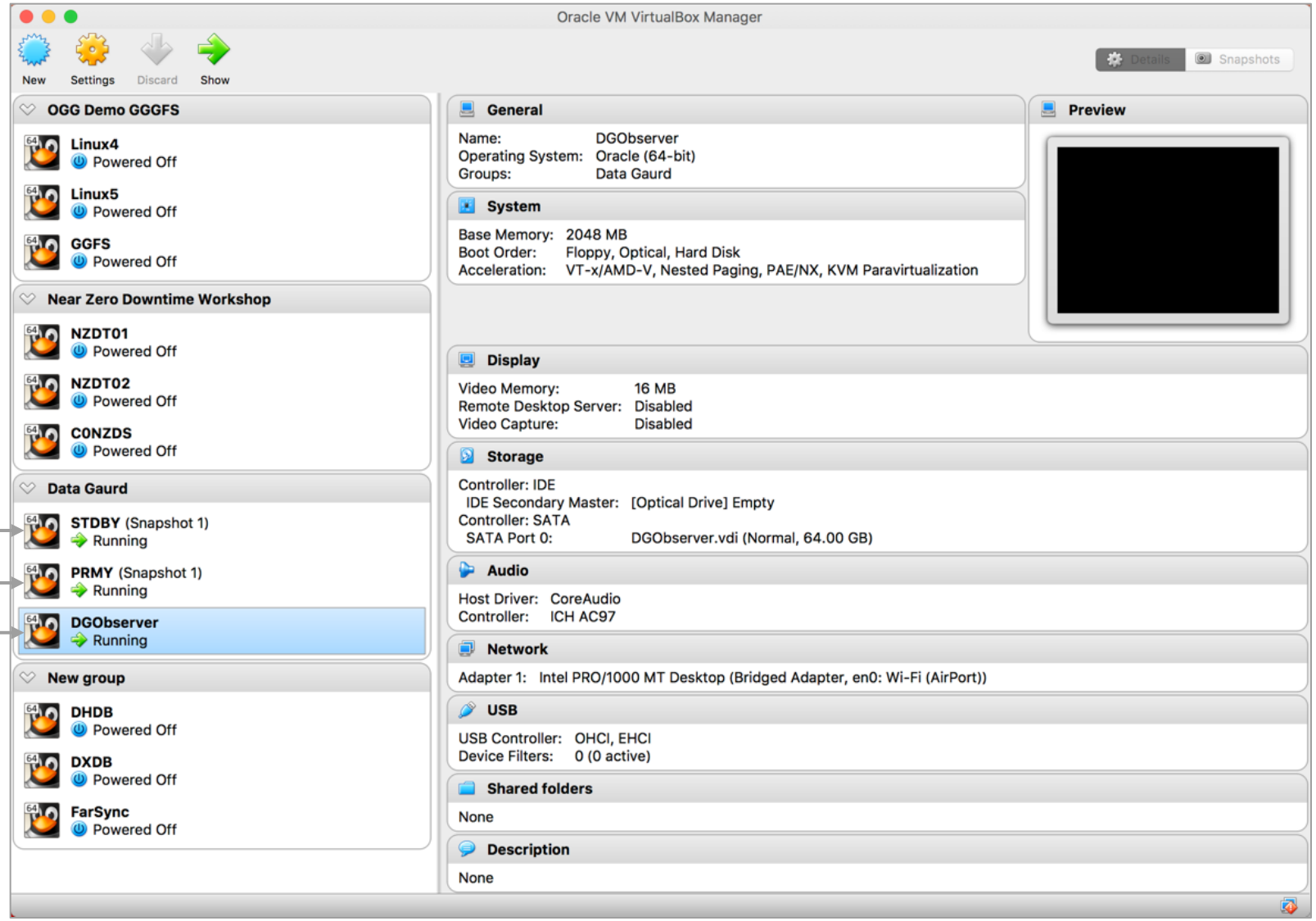


Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

HOL VirtualBox Machines

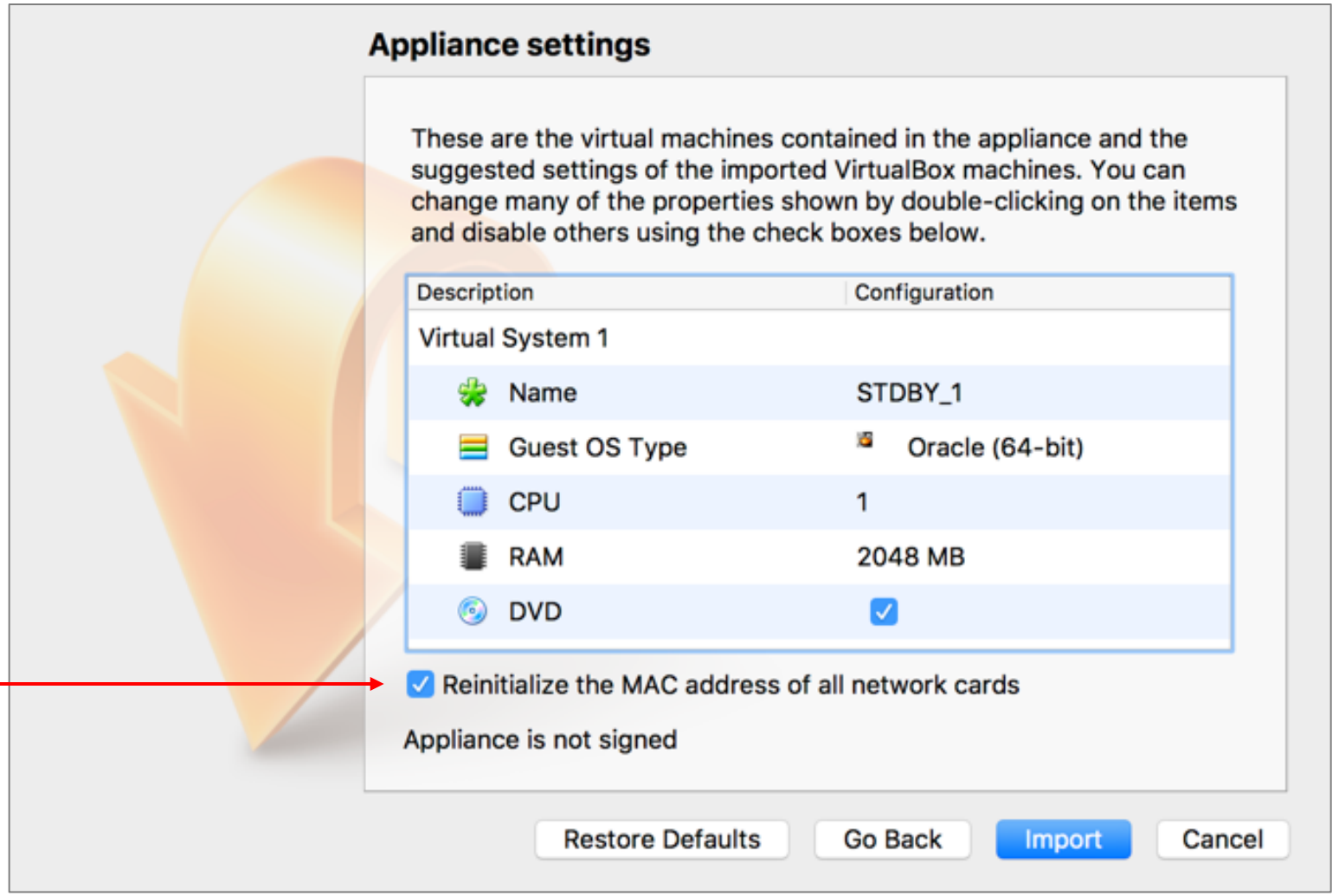
DGv1.OFA

- Standby Database
- Primary Database
- Observer Server



Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

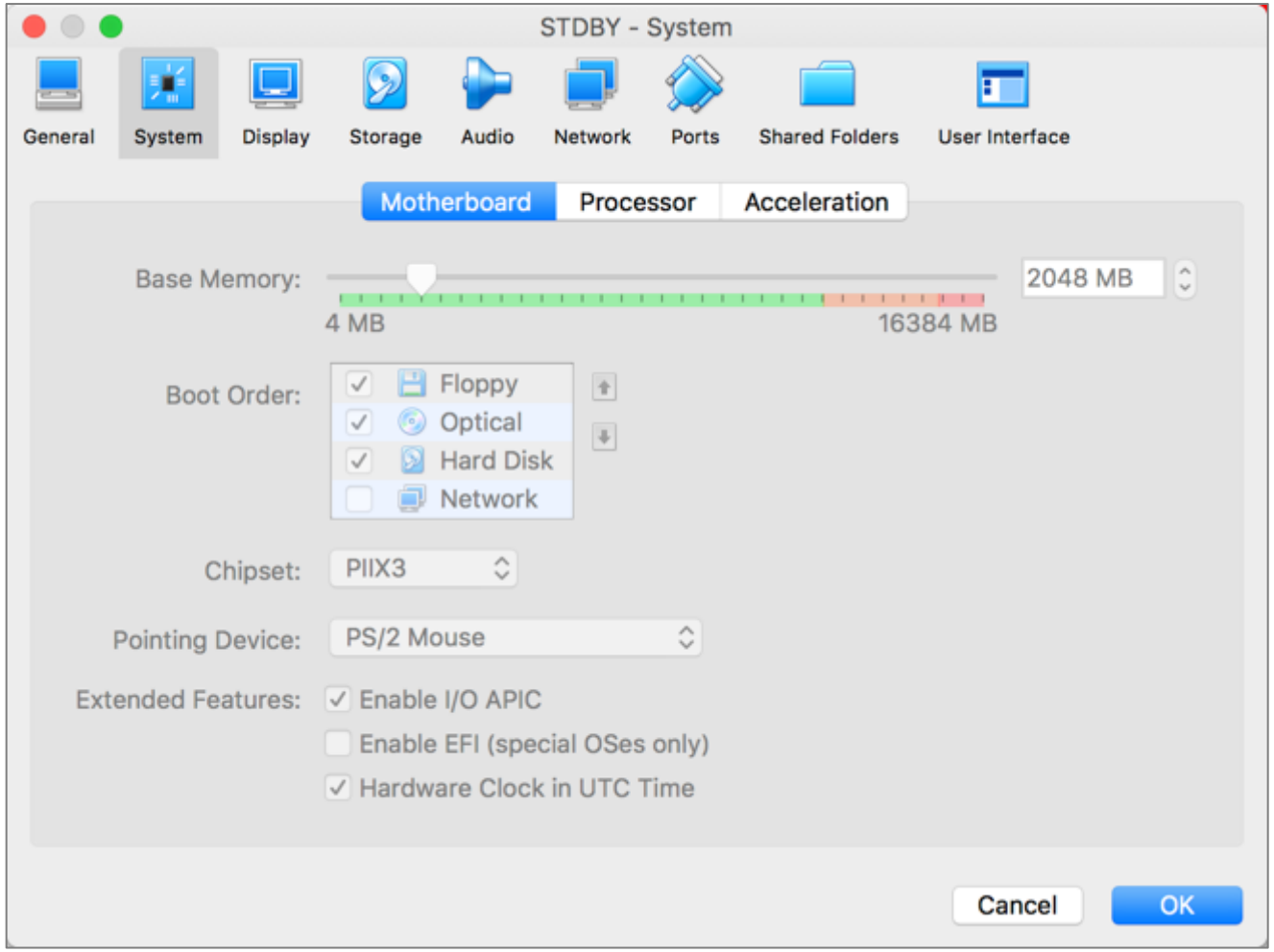
Import VirtualBox



Enable

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

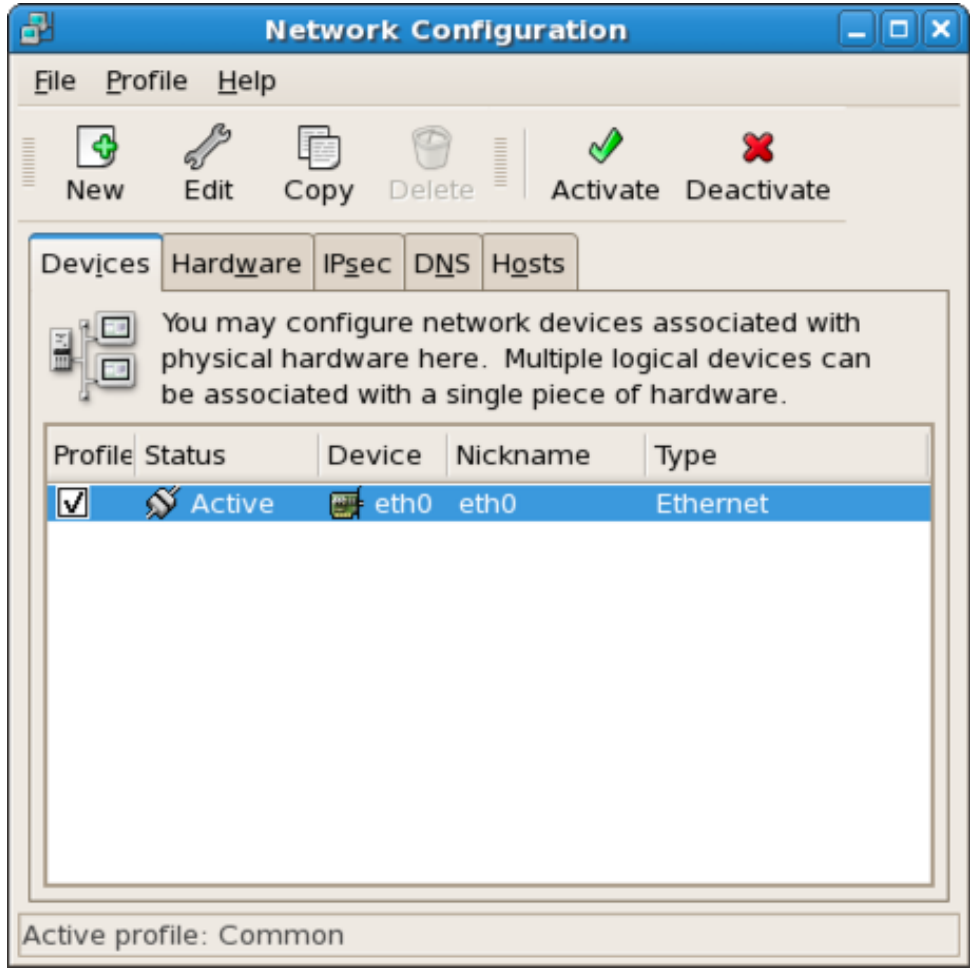
Memory Settings



PRMY	2048MB
STDBY	2048MB
DGOBSERVER	2048MB

Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Linux Network Settings



Delete Backup Device
Configure IP Address – 192.168.1.40/50/30
Configure Sub-Net – 255.255.255.0
Activate

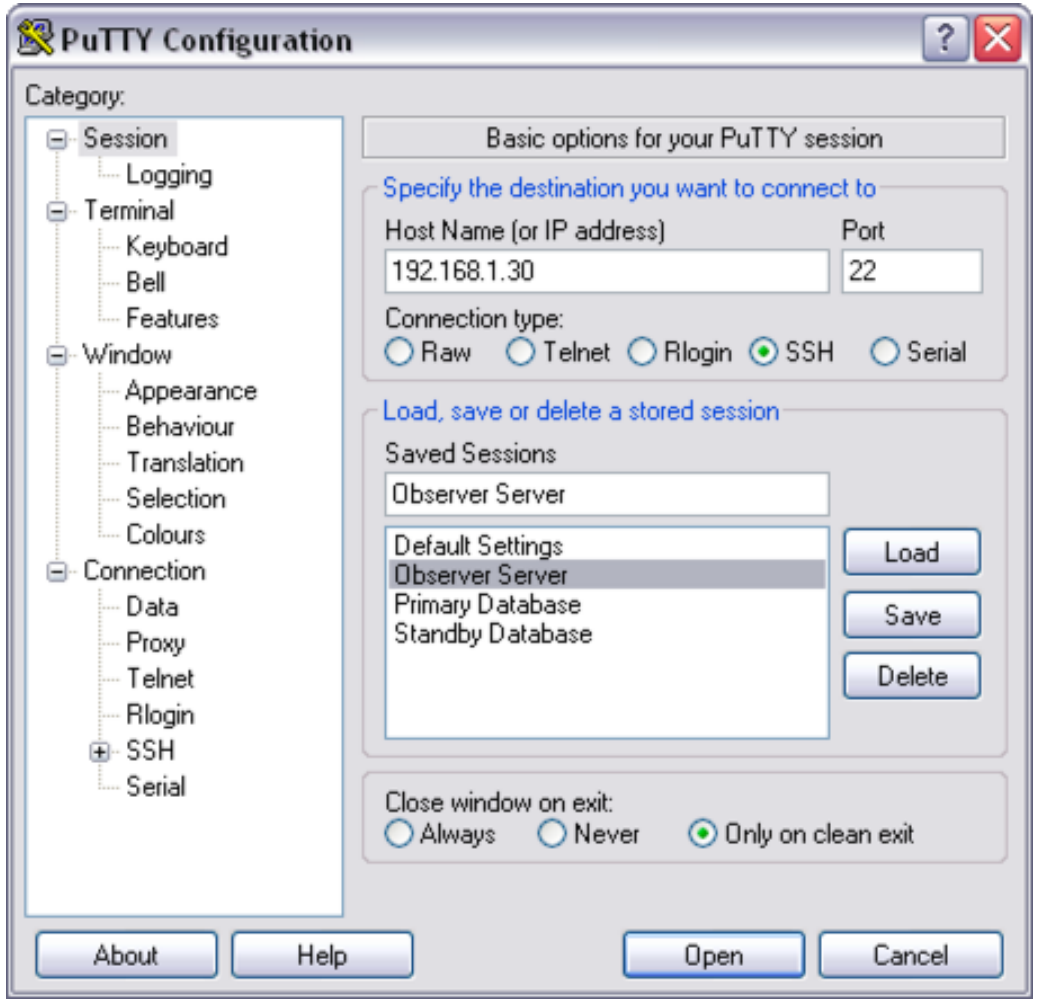
Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

Configure PuTTY Terminals

Primary Database: 192.168.1.40

Standby Database: 192.168.1.50

Observer Server: 192.168.1.30



Copyright © 2015 Oracle and/or its affiliates. All rights reserved. |

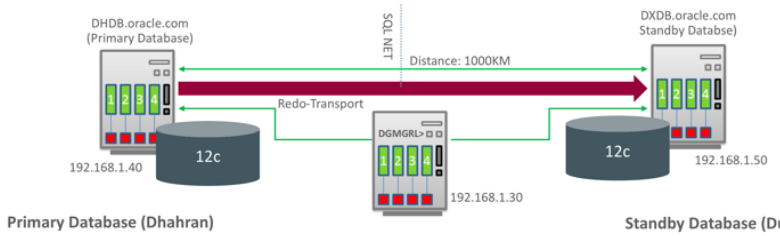

ORACLE®

Oracle Data Guard FAST-START Failover Workshop, Activity Guide

APPENDIX A

An Overview

This document outlines the steps to deploy Oracle Active Data Guard (ADG), and one of the unique ADG add value features, the automation of failover using FAST-START failover and re-instatement. Figure 2 shows the Hands-On lab for a Data Guard VirtualBox machines environment.



Primary Database (Dhahran)

Name	DHDB.oracle.com
Version	12.1
RAC	No
ASM	No
SYS	oracle_4U
SYSTEM	oracle_4U
Home	/u01/app/oracle/product/12.1.0/dbhome_1

Standby Database (Dubai)

Name	DXDB.oracle.com
Version	12.1
RAC	No
ASM	No
SYS	oracle_4U
SYSTEM	oracle_4U
Home	/u01/app/oracle/product/12.1.0/dbhome_1

Figure 2: Hands-On Lab environment

The Data Guard deployment is delivered using Oracle Recovery Manager (RMAN). The primary database (DHDB.oracle.com) continuously sends the transaction logs to the standby database on the protection mode configured. The Oracle Data Guard Observer monitor and manages the Data Guard configuration and based on default and configured properties settings and thresholds, an automated failover and re-instatement internally processed without the DBA manual intervention, eliminating the human error and increasing business continuity.

The workshop allows the data guard configuration to perform *Switchover* to the current standby database. Enabling of the primary and the standby database Flashback achieves automatic FAST-START failover and automatic re-instatement. RMAN is configured to execute backup on the standby database, Active Data Guard for query offloading is configured for enabling efficient resource utilization.

Environment Configuration

Table 1, 2 and 3 show the details for the systems depicted on Figure 1. The primary database already configured on Archive Log mode. Review the properties for each of the system listed below.

Primary System

Hostname	dhdb01.oracle.com
IP Address	192.168.1.40
OS root password	oracle
OS oracle password	oracle
Global Name	DHDB.oracle.com
SYS	oracle_4U
SYSTEM	oracle_4U
Archive Mode	Enabled

Table 1: Primary System details

Standby System

Hostname	dxdb01.oracle.com
IP Address	192.168.1.50
OS root password	oracle
OS oracle password	oracle
Global Name	DXDB.oracle.com
SYS	oracle_4U
SYSTEM	oracle_4U
Archive Mode	Enabled

Table 2: Standby System details

Primary System

Hostname	observer.oracle.com
IP Address	192.168.1.30
OS root password	oracle
OS oracle password	oracle
Global Name	
SYS	
SYSTEM	
Archive Mode	

Table 3: Observer System details

Linux Basic Commands for MS Windows Users

Table 4 lists Linux command commonly executed when managing Oracle Database running on Linux operating system.

Linux Command	Command Description
`$mkdir –p /u01/app/oracle`	Create full path directory
`$chown –R oracle:oinstall /u01`	Change the directory owner recursively
`$ls-ltr`	List files sorted by the creation date
`$du .`	Display disk usage
`$id`	Display current user details
`$chmod 777 .bash_profile`	Change permission
`$scp * root@192.168.1.30:/tmp`	Remote secure copy
`$touch file1`	Create a file
`$su`	Switch to super user
`$rm file1`	Remote file
`#reboot`	Reboot the operating system
`$df -k`	Report file system usage
`$ps –edf | grep pmon`	Display running processes
`#sysctl -p`	Re-activate the current kernel parameters
`$export DISPLAY=192.168.1.20:0.0`	Enable shell environment variable
`$hostname`	Display the host name
`$/sbin/ifconfig`	Display network interface
`$cp f1 f2`	Copy file
`$cd`	Change to home directory
`$vi`	Start the editor

Table 4: Linux Commands

Linux/UNIX Editor vi

Table 5 lists key-combination is for opening, modifying and saving files.

Keys	Description
`vi`	Open the editor
`i`	Insert
`Save`	:wq!
`:wq! f1.txt`	Create the file f1.txt
`:%s/DHDB/DXDB/`	Search and replace for a string
`Small k`	Go up
`Small j`	Go down
`Small l`	Go right
`Small h`	Go left

Table 5: Linux Editor vi

Database Design

The database schema OSM_REPAPI owns the tables used by the workshop. The CUSTOMERS and POLICIES are mapped with 1:1 constraint. The POLICY_TYPES and DISCOUNT_TYPES are lookup tables. The workload simulator generates the primary keys for the CUSTOMERS and POLICIES tables. Figure 3 illustrates the data model used by the workload simulator.

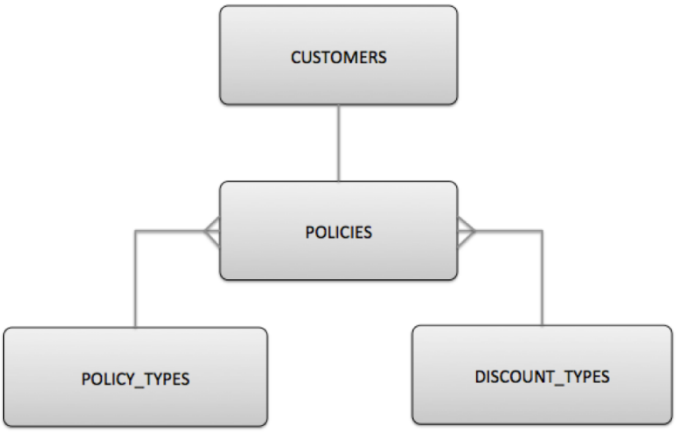


Figure 3: Workload Simulator Database

Workload Simulator

A workload simulators DBMS_OGG and stand-alone procedure are developed using PL/SQL. The database procedure TRANS_LOAD is invoked, passing the workload parameters value as explained below. Use the default parameters, unless otherwise instructed to use different values.

Parameter	Description
N	The workload level, default value 500, which inserts 5000 rows
S	The wait second between two transactions, default value is 5 seconds.

```
SQL>desc trans_load
PROCEDURE trans_load
Argument Name                    Type In/Out Default?
 ------------------------- ------------------ ------ --------
 PLOAD_LEVEL                    NUMBER    IN     DEFAULT
 PSTARTFROM                     NUMBER    IN     DEFAULT

SQL>desc call_trans_load
PROCEDURE call_trans_load
Argument Name                    Type In/Out Default?
```

```
 ------------------------- ------------------ ------ --------
 N                                            NUMBER    IN     DEFAULT
 S                                            NUMBER    IN      DEFAULT

SQL>
```

```
create or replace procedure call_trans_load(n number default 500,
                                            s number default 5) is
vsid number;
vserial# number;
   cursor c1 is select sid, serial# into vsid, vserial#
                from v$sql s1, v$session s2
                where sql_text like '%sys.dbms_lock.sleep(s);%' and
                      s1.sql_id = s2.sql_id;
begin
   open c1;
   loop
      fetch c1 into vsid, vserial#;
      exit when c1%notfound;
dbms_output.put_line('-- alter system kill session
'||''''||vsid||',''||vserial#||''''||' immediate;');
   end loop;
   close c1;
   --
   for i in 1..n loop
       trans_load;
       sys.dbms_lock.sleep(s);
   end loop;
end;
/
```

Run the workload simulator using the default values. The actual parameters uses the default values; N=500 and S=5.

```
SQL> exec call_trans_load;
```

Run the workload simulator using N=2000, this will sustain transactions load of approximately 4hrs.

```
SQL> exec call_trans_load(2000);
```

The following call is an extreme level, use this parameter setting when performing the workshop on 8-16 cores system.

```
SQL> exec call_trans_load(300000, 1);
```

Workshop Structure

The workshop composed of six parts as described below. Because of dependencies among the parts, the workshop's parts should be completed on the order outlined in Figure 4. The workshop assumes the delegate has Oracle database hands-on administration or has attended Oracle University database administrator training.

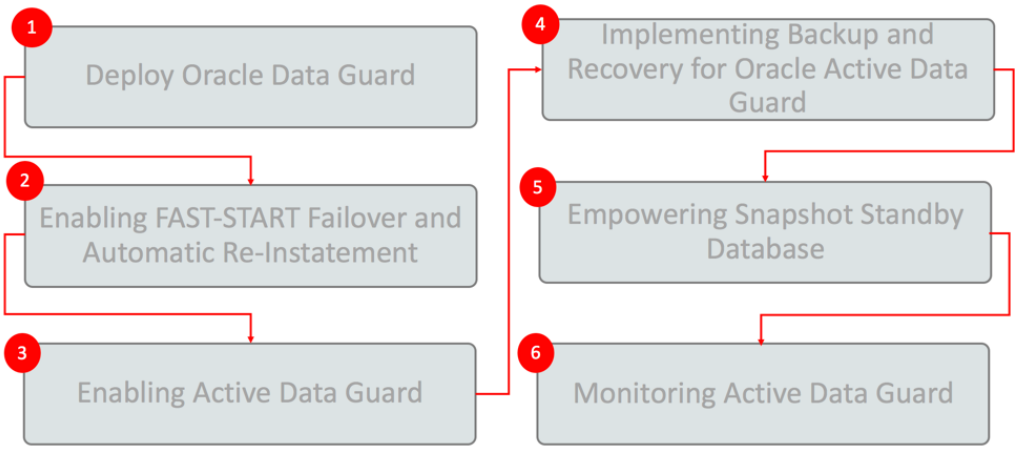


Figure 4: Workshop Structure

Part I: Deploying Oracle Data Guard

Part I deploys Oracle Data Guard by configuring the data guard configuration for the Primary database (DHDB) and the Auto-Managed standby database (DXDB). The preparation of the primary to support data guard redo-transport service. Using the Recovery Manager (RMAN), the standby database created online as an auxiliary database. The primary database and standby database configured to support manual switchover and failover operations. Setting the database parameters for the data guard broker and creating and enabling the data guard broker configuration using DGMGRL for DHDB and DXDB.

Part II: Enabling FAST-START Failover and Re-Instatement

This part configures the data guard FAST-START failover and re-instatement features. The purpose is to automate failover operation and primary database re-instatement. Configuring of Fast Recovery Area (FRA) and Flashback database is a pre-requisite. Using the data guard broker to designate the FAST-START failover physical standby database and the FAST-START failover threshold, database

properties and FAST-START conditions to enable manageable yet automated failover.

Part III: Enabling Active Data Guard

This part is simply enabling real-time query offloading on an Oracle data guard. The auto-managed standby database is open on read-only to enable running reports and RMAN backups, offloading the primary database from long-running operations. The Oracle Active Data Guard operates the standby database on read-only mode while its continuously synchronized through the data guard applies service.

Part IV: Implementing Backup and Recovery on Oracle Active Data Guard

This part demonstrates how to use Oracle Recovery Manager for backup and recovery operations for environment running Oracle Data Guard. Using the Oracle Recovery Manager (RMAN) catalog, RMAN backup, restore and recover scripts and RMAN persistent policies, the backing up of the primary database is performed from the physical standby database. Enabling the database change block tracking feature enable highly optimized incremental backup for VLDB.

Part V: Empowering Snapshot Standby Database

This part demonstrates highly flexible Data Guard feature through the use of Snapshot Standby database. The snapshot database is a temporary read-write database that is mostly used by the database replay for performance analysis and application testing. The snapshot standby database requires enabling the flashback database feature to automatically revert the database backup to physical standby database.

Part VI: Monitoring Active Data Guard

This part is for analyzing and troubleshooting database status running on Oracle data guard configuration. Monitor the Redo-Transport and the Redo-Apply services. Finding out the cause of FAST-START failover and the various areas to troubleshoot, and fix database errors running on Oracle Active Data Guard Environment.

Conventions

There are three symbolic conversion used through out the workshops:



Designate the Primary Database System



Designate the Standby Database System



Designate the Observer Server

The workshop commands are executed interactively or via a Linux shell scripts. For interactive mode, execute the command in bold only.




```
[oracle@prmy01 ~]$ dgmgrl
DGMGRL for Linux: Version 12.1.0.1.0 - 64bit Production

Copyright (c) 2000, 2012, Oracle. All rights reserved.

Welcome to DGMGRL, type "help" for information.
DGMGRL> connect sys/oracle
Connected as SYSDG.
DGMGRL> create configuration 'DG_CONF' as primary database is 'DHDB' connect
identifier is 'DHDB';
Configuration "DG CONF" created with primary database "DHDB"
DGMGRL> enable configuration;
Enabled.
DGMGRL> add database 'DXDB' as connect identifier is 'DXDB';
Database "DXDB" added
DGMGRL> enable database 'DHDB';
Enabled.
DGMGRL> enable database 'DXDB';
Enabled.
DGMGRL>
```



Successful command indicator

The interactive utilities used throughout the workshop:

SQL>	SQL*Plus
RMAN>	Oracle Recovery Manager
DGMGRL>	Data Guard
vi	Linux Editor
#	Root prompt
$	Oracle prompt

Importing the VirtualBox Machines

Import the VirtualBox machines must be imported using the VirtualBox Manager. Figure 5 and Figure 6 shows the import steps. Select File->Import Appliance.

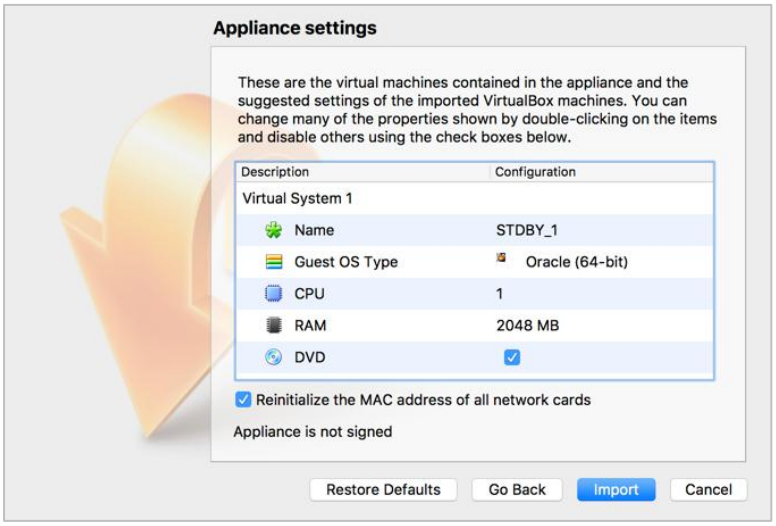


Figure 5: VirtualBox Import

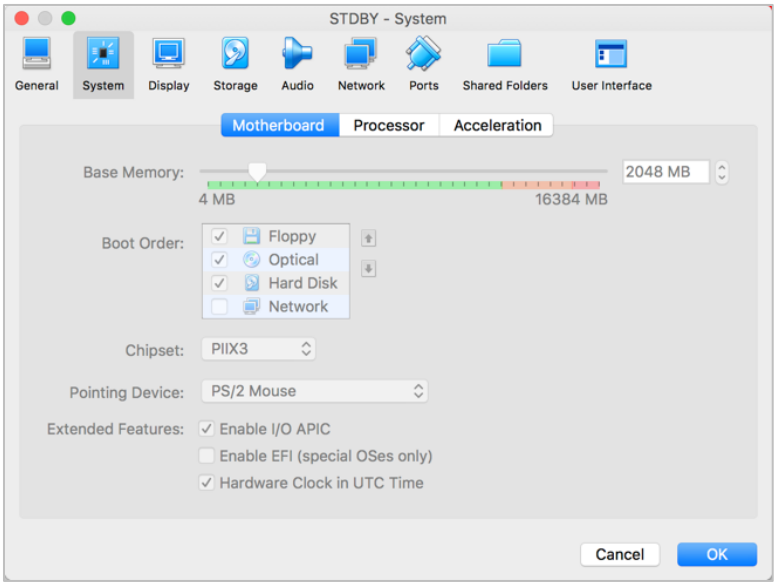


Figure 6: Adjust Memory Settings

Connect to Oracle Linux

The workshops require three VirtualBox machines as depicted on Figure 7. The machines are configured as follows:

IP Address	Hostname
192.168.1.40	prmy01.oracle.com
192.168.1.50	stdby01.oracle.com
192.168.1.30	dgobserver.oracle.com

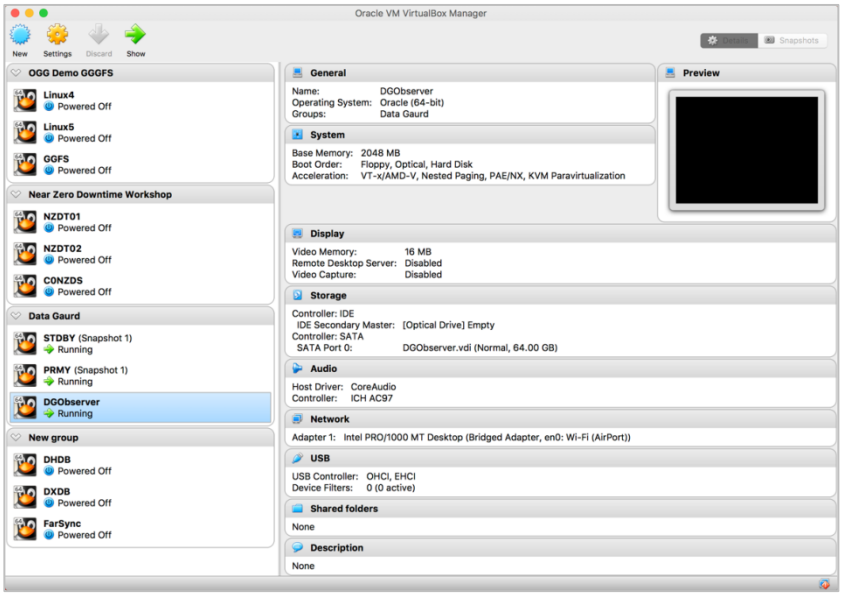


Figure 7: VirtualBox Machine Started

Connectivity Options

There are two options to connect a PuTTY terminal to the VirtualBox machines, which are:

- Virtual Loopback Network
- Wired or wireless network

For this workshop, connect to Wi-Fi OracleGoldenGateW. The password of this network is: Welcome10.

Configure PuTTY Terminals

Configure PuTTY as shown on Figure 8. It's configured to connect to:

The primary database system – 192.168.1.40
The Standby database system – 192.168.1.50
The observer system – 192.168.1.30

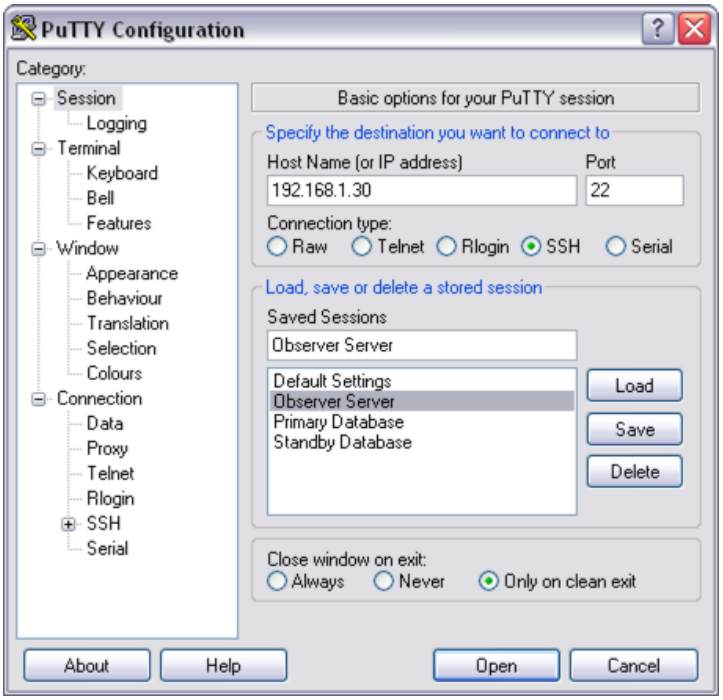


Figure 8: PuTTY Terminals

Open terminal sessions by selecting from the saved session and click on Open.
Repeat the terminal session for the Primary, Standby and Observer systems.

Login credentials:

The password for OS root user: oracle
The password for OS oracle user: oracle



Please, do not proceed further if you're unable to create three Linux terminal sessions.

Part I: Deploy Oracle Data Guard

This workshop deploys Oracle Data Guard by configuring the data guard configuration for the Primary database (DHDB) and Auto-Managed standby database (DXDB). Using the Recovery Manager, the standby database created online as an auxiliary database. The primary database and standby database configured to support manual switchover and failover operation.

Tasks

Table 6 lists the tasks associated with Part I, Deploy Oracle Data Guard.

Task#	Description
1	Configure Primary Database Force Logging
2	Update /etc/oratab file
3	Re-Create the Password File
4	Configure SQLNET Static Service Name Registration
5	Stop and Restart the Listener
6	Configure SQLNET Clients
7	Configure SQLNET Static Service Name Registration
8	Test SQLNET Services
9	Test SQLNET Services
10	Prepare the NOMOUNT initDXDB.ora Parameter File
11	Create the Auxiliary Instance Sub-Directories
12	Startup the Auxiliary Instance in NOMOUNT State
13	Create the Auxiliary Instance Database
14	Verify the Cloned Physical Standby Database is Successful
15	Configure the Primary Instance
16	Configure the Physical Standby Database Instance
17	Add standby Redo Log to the Standby Instance
18	Add Standby Redo Log to the Primary Instance
19	Clear Archive Destination
20	Clear Archive Destination
21	Create and Enable Data Guard Broker Configuration
22	Display and Review Data Guard Configuration Details for DHDB
23	Display and Review Data Guard Configuration Details for DXDB
24	Perform Planned Switchover to Current Standby Site
25	Perform Planned Switchover to Current Standby Site
26	Verify the Primary Database Role
27	Verify the Standby Database Role

Table 6: Deploy Oracle Data Guard Tasks

Architecture

Oracle Data Guard is an integral part of Oracle Database (RDBMS). Figure 8 shows Oracle Data Guard components. Depending on the protection mode and the apply mode, the process flow may slightly altered. The architecture depicts the primary database (DHDB) is protected by one standby database (DXDB).

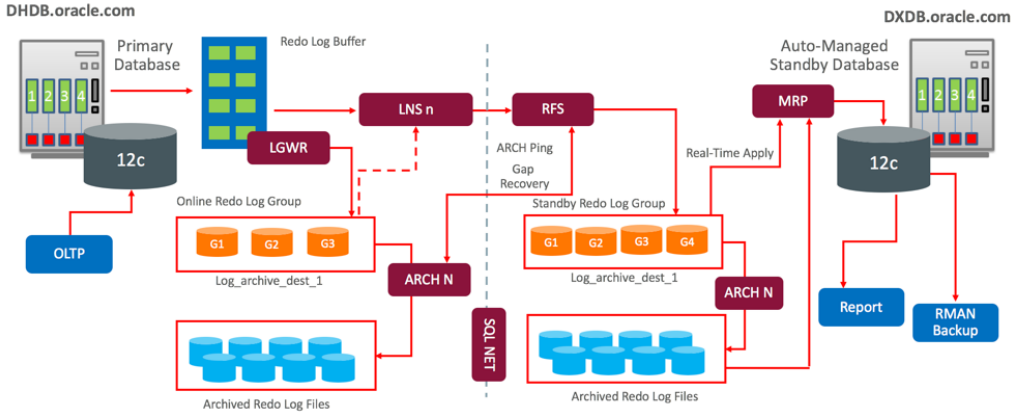


Figure 8: Oracle Data Guard Architecture

- The Primary Database (DHDB) runs OLTP applications; the LGWR writes committed transactions to the database online redo log files.
- The archive processes create the archived redo log file.
- For each destination configured for redo-transport, an LNS process is spawn to send data to the remote RFS process; this can be SYNC or ASYNC mode.
- The RFS writes to the standby online redo log files.
- For real-time apply, the MRP process applies the transactions to the standby database, for non real-time, the archive processes write to the standby archived redo log files, which are applied to the standby database by the MPR process.

The physical auto-managed standby database is enabled for Active Data Guard (ADG); the instance is able to perform query offloading for running reports application and enabled for optimized RMAN backup and automatic block repair.

Because this is a physical standby database, backup of the primary database can be executed on the standby database. Performing backup on standby database system using RMAN is a valid backup for the primary database.

The redo-Transport service uses TCP/IP to ship the transaction logs to the standby database standby database (DXDB) for supporting the Data Guard configuration.

P Configure Primary Database Force Logging

Enabling Force logging at the database level overrides database logging at the tablespace and table levels. Setting force logging writes the database change-vector to the redo logs steam required by the data guard. The primary database is configured to operate on archive log mode.

View and run the script $HOME/admin/s1_force_logging.sh.

Complete

```
[oracle@prmy01 ~]$ sqlplus /nolog

SQL*Plus: Release 12.1.0.1.0 Production on Fri Nov 3 14:38:49 2017

Copyright (c) 1982, 2013, Oracle.  All rights reserved.

SQL>conn / as sysdba
Connected.
SQL> select log_mode from v$database;

LOG MODE
------------
ARCHIVELOG

SQL> alter database force logging;

Database altered.

SQL> select force_logging from v$database;

FORCE_LOGGING
---------------------------------------
YES

SQL> exit
Disconnected from Oracle Database 12c Enterprise Edition Release 12.1.0.1.0
- 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real Application Testing
options
[oracle@prmy01 ~]$
```

S Update /etc/oratab File

Prior to database cloning by RMAN to create the standby database, add the instance DXDB name, home directory and auto-startup flag to the end of/etc/oratab file.

View and run the script $HOME/admin/s2_oratab_edit.sh

Complete

```
[oracle@stdby01 ~]$ vi /etc/oratab
```

```
# This file is used by ORACLE utilities.  It is created by root.sh
```

```
# and updated by either Database Configuration Assistant while creating
# a database or ASM Configuration Assistant while creating ASM instance.

# A colon, ':', is used as the field terminator.  A new line terminates
# the entry.  Lines beginning with a pound sign, '#', are comments.
#
# Entries are of the form:
#   $ORACLE_SID:$ORACLE_HOME:<N|Y>:
#
# The first and second fields are the system identifier and home
# directory of the database respectively.  The third field indicates
# to the dbstart utility that the database should , "Y", or should not,
# "N", be brought up at system boot time.
#
# Multiple entries with the same $ORACLE_SID are not allowed.
#
DXDB:/u01/app/oracle/product/12.1.0/dbhome_1:Y
```

P Re-Create the Password File

Because the data guard services and data guard broker need to login to all participating instances, the password file must identical for instances. Use the Oracle utility orapwd to re-create the password file and copy across to the standby system.

View and run the script $HOME/admin/s3_create_PWD.sh

Complete

```
[oracle@prmy01 ~]$ id
uid=54321(oracle) gid=54321(oinstall) groups=54321(oinstall),54322(dba)
```

```
[oracle@prmy01 ~]$ cd $ORACLE_HOME/dbs
[oracle@prmy01 dbs]$ ls -al orapwDHDB
-rw-r----- 1 oracle oinstall 17920 Nov  3 14:39 orapwDHDB
[oracle@prmy01 dbs]$ rm orapwDHDB
[oracle@prmy01 dbs]$ orapwd file=$ORACLE_HOME/dbs/orapwDHDB password=oracle
ignorecase=y force=y entries=30
[oracle@prmy01 dbs]$ ls -al orapwDHDB
-rw-r----- 1 oracle oinstall 17920 Nov  3 14:40 orapwDHDB
```

```
[oracle@prmy01 dbs]$ scp $ORACLE_HOME/dbs/orapwDHDB
root@192.168.1.50:$ORACLE_HOME/dbs/orapwDXDB
The authenticity of host '192.168.1.50 (192.168.1.50)' can't be established.
RSA key fingerprint is 88:c5:9b:0b:de:f3:0f:cb:ea:62:e5:30:37:c0:1f:8e.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.1.50' (RSA) to the list of known hosts.
root@192.168.1.50's password: <oracle>
orapwDHDB
100%   18KB  17.5KB/s   00:00
[oracle@prmy01 dbs]$
```

S On the standby system, change the password owner to 'oracle'.

```
[oracle@stdby01 ~]$ cd $ORACLE_HOME/dbs
[oracle@stdby01 dbs]$ su
Password: <oracle>
[root@stdby01 dbs]# ls -al orapwDXDB
-rw-r----- 1 root root 17920 Nov  3 15:13 orapwDXDB
[root@stdby01 dbs]# chown oracle:oinstall orapwDXDB
[root@stdby01 dbs]# exit
exit
[oracle@stdby01 dbs]$ ls -al orapwDXDB
-rw-r----- 1 oracle oinstall 17920 Nov  3 15:13 orapwDXDB
[oracle@stdby01 dbs]$
```



Configure SQLNET Static Service Name Registration

It a must to setup two static database instance registration services. One to accept connections from the application and one dedicated for the data guard broker. Stop and start the listener after making the modifications.

View and run the script $HOME/admin/s4_copy_sqlnet.sh

$ORACLE_HOME/network/admin/listener.ora




```
[oracle@prmy01 ~]$ cd $ORACLE_HOME/network/admin
[oracle@prmy01 admin]$ vi listener.ora
```

```
SID_LIST_LISTENER =
  (SID_LIST =
    (SID_DESC =
      (GLOBAL DBNAME = DHDB.oracle.com)
      (ORACLE HOME = /u01/app/oracle/product/12.1.0/dbhome 1)
      (SID_NAME = DHDB)
    )
    (SID_DESC =
      (GLOBAL_DBNAME = DHDB_DGMGRL.oracle.com)
      (ORACLE_HOME = /u01/app/oracle/product/12.1.0/dbhome_1)
      (SID_NAME = DHDB)
    )
  )

LISTENER =
  (DESCRIPTION_LIST =
    (DESCRIPTION =
      (ADDRESS = (PROTOCOL = TCP)(HOST = prmy01.oracle.com)(PORT = 1521))
    )
    (DESCRIPTION =
      (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC1521))
    )
  )

ADR_BASE_LISTENER = /u01/app/oracle
```

P Stop and Restart the Listener

Stop and start the listener, make sure the DHDB_DGMGLR.oracle.com is listed.

Complete

```
[oracle@prmy01 admin]$ lsnrctl

LSNRCTL for Linux: Version 12.1.0.1.0 - Production on 29-NOV-2017 09:23:19

Copyright (c) 1991, 2013, Oracle.  All rights reserved.

Welcome to LSNRCTL, type "help" for information.

LSNRCTL> stop
Connecting to
(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=prmy01.oracle.com)(PORT=1521)))
The command completed successfully
LSNRCTL> start
Starting /u01/app/oracle/product/12.1.0/dbhome_1/bin/tnslsnr: please wait...

TNSLSNR for Linux: Version 12.1.0.1.0 - Production
System parameter file is
/u01/app/oracle/product/12.1.0/dbhome 1/network/admin/listener.ora
Log messages written to
/u01/app/oracle/diag/tnslsnr/prmy01/listener/alert/log.xml
Listening on:
(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=prmy01.oracle.com)(PORT=1521)))
Listening on: (DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=EXTPROC1521)))

Connecting to
(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=prmy01.oracle.com)(PORT=1521)))
STATUS of the LISTENER
------------------------
Alias                     LISTENER
Version                   TNSLSNR for Linux: Version 12.1.0.1.0 - Production
Start Date                29-NOV-2017 09:23:25
Uptime                    0 days 0 hr. 0 min. 0 sec
Trace Level               off
Security                  ON: Local OS Authentication
SNMP                      OFF
Listener Parameter File
/u01/app/oracle/product/12.1.0/dbhome_1/network/admin/listener.ora
Listener Log File
/u01/app/oracle/diag/tnslsnr/prmy01/listener/alert/log.xml
Listening Endpoints Summary...
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=prmy01.oracle.com)(PORT=1521)))
  (DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=EXTPROC1521)))
Services Summary...
Service "DHDB.oracle.com" has 1 instance(s).
  Instance "DHDB", status UNKNOWN, has 1 handler(s) for this service...
Service "DHDB_DGMGRL.oracle.com" has 1 instance(s).
  Instance "DHDB", status UNKNOWN, has 1 handler(s) for this service...
The command completed successfully
LSNRCTL> exit
[oracle@prmy01 admin]$
```

P Configure SQLNET Clients

Setup SQLNET clients by modifying tnsnames.ora file; include DXDB and DHDB TNS names to $ORACLE_HOME/network/admin/tnsnames.ora.

Complete

```
[oracle@prmy01 admin]$ vi tnsnames.ora
```

```
DHDB =
  (DESCRIPTION =
    (ADDRESS LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = prmy01.oracle.com)(PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = DHDB.oracle.com)
    )
  )

DXDB =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = stdby01.oracle.com)(PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = DXDB.oracle.com)
    )
  )
```

S Configure SQLNET Static Service Name Registration

It a must to setup two static database instance registration services. One to accept connections from the application and one dedicated for the data guard broker. Stop and start the listener after making the modifications.

View and run the script $HOME/admin/s5_copy_sqlnet.sh

$ORACLE_HOME/network/admin/listener.ora

Complete

```
[oracle@stdby01 dbs]$ cd $ORACLE_HOME/network/admin
[oracle@stdby01 admin]$ vi listener.ora
```

```
SID_LIST_LISTENER =
  (SID_LIST =
    (SID_DESC =
      (GLOBAL_DBNAME = DXDB.oracle.com)
      (ORACLE_HOME = /u01/app/oracle/product/12.1.0/dbhome_1)
      (SID_NAME = DXDB)
    )
    (SID_DESC =
```

```
      (GLOBAL_DBNAME = DXDB_DGMGRL.oracle.com)
      (ORACLE_HOME = /u01/app/oracle/product/12.1.0/dbhome_1)
      (SID_NAME = DXDB)
    )
  )

LISTENER =
  (DESCRIPTION_LIST =
    (DESCRIPTION =
      (ADDRESS = (PROTOCOL = TCP)(HOST = stdby01.oracle.com)(PORT = 1521))
    )
    (DESCRIPTION =
      (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC1521))
    )
  )

ADR_BASE_LISTENER = /u01/app/oracle
```

Start the listener on the standby system, if not already started.

```
[oracle@stdby01 admin]$ lsnrctl

LSNRCTL for Linux: Version 12.1.0.1.0 - Production on 03-NOV-2017 15:47:36

Copyright (c) 1991, 2013, Oracle.  All rights reserved.

Welcome to LSNRCTL, type "help" for information.

LSNRCTL> start
Starting /u01/app/oracle/product/12.1.0/dbhome_1/bin/tnslsnr: please wait...

TNSLSNR for Linux: Version 12.1.0.1.0 - Production
System parameter file is
/u01/app/oracle/product/12.1.0/dbhome_1/network/admin/listener.ora
Log messages written to
/u01/app/oracle/diag/tnslsnr/stdby01/listener/alert/log.xml
Listening on:
(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=stdby01.oracle.com)(PORT=1521)))
Listening on: (DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=EXTPROC1521)))

Connecting to
(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=stdby01.oracle.com)(PORT=1521)))
STATUS of the LISTENER
------------------------
Alias                     LISTENER
Version                   TNSLSNR for Linux: Version 12.1.0.1.0 - Production
Start Date                03-NOV-2017 15:47:41
Uptime                    0 days 0 hr. 0 min. 0 sec
Trace Level               off
Security                  ON: Local OS Authentication
SNMP                      OFF
Listener Parameter File
/u01/app/oracle/product/12.1.0/dbhome_1/network/admin/listener.ora
Listener Log File
/u01/app/oracle/diag/tnslsnr/stdby01/listener/alert/log.xml
Listening Endpoints Summary...
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=stdby01.oracle.com)(PORT=1521)))
```

```
  (DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=EXTPROC1521)))
Services Summary...
Service "DXDB.oracle.com" has 1 instance(s).
  Instance "DXDB", status UNKNOWN, has 1 handler(s) for this service...
Service "DXDB DGMGRL.oracle.com" has 1 instance(s).
  Instance "DXDB", status UNKNOWN, has 1 handler(s) for this service...
The command completed successfully
LSNRCTL> exit
[oracle@stdby01 admin]$
```

Setup SQLNET clients by creating tnsnames.ora file; include DXDB and DHDB TNS names to $ORACLE_HOME/network/admin/tnsnames.ora

Complete

```
[oracle@stdby01 admin]$ pwd
/u01/app/oracle/product/12.1.0/dbhome_1/network/admin
[oracle@stdby01 admin]$
[oracle@stdby01 admin]$ vi tnsnames.ora
```

```
DXDB =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = stdby01.oracle.com)(PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = DXDB.oracle.com)
    )
  )

DHDB =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = prmy01.oracle.com)(PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = DHDB.oracle.com)
    )
  )
```

P Test SQLNET Services

Test the service names are setup correctly. Use the Oracle utility tnsping to verify service names configuration is successful.

Complete

```
[oracle@prmy01 admin]$ tnsping DHDB

TNS Ping Utility for Linux: Version 12.1.0.1.0 - Production on 12-NOV-2017
06:07:55

Copyright (c) 1997, 2013, Oracle.  All rights reserved.
```

```
Used parameter files:


Used TNSNAMES adapter to resolve the alias
Attempting to contact (DESCRIPTION = (ADDRESS LIST = (ADDRESS = (PROTOCOL =
TCP)(HOST = prmy01.oracle.com)(PORT = 1521))) (CONNECT_DATA = (SERVICE_NAME
= DHDB.oracle.com)))
OK (0 msec)
[oracle@prmy01 admin]$ tnsping DXDB

TNS Ping Utility for Linux: Version 12.1.0.1.0 - Production on 12-NOV-2017
06:08:02

Copyright (c) 1997, 2013, Oracle.  All rights reserved.

Used parameter files:


Used TNSNAMES adapter to resolve the alias
Attempting to contact (DESCRIPTION = (ADDRESS_LIST = (ADDRESS = (PROTOCOL =
TCP)(HOST = stdby01.oracle.com)(PORT = 1521))) (CONNECT_DATA = (SERVICE_NAME
= DXDB.oracle.com)))
OK (0 msec)
[oracle@prmy01 admin]$
```



Test SQLNET Services

Test the service names are setup correctly. Use the Oracle utility tnsping to verify service names configuration is successful.




```
[oracle@stdby01 admin]$ tnsping DXDB

TNS Ping Utility for Linux: Version 12.1.0.1.0 - Production on 12-NOV-2017
06:05:10

Copyright (c) 1997, 2013, Oracle.  All rights reserved.

Used parameter files:


Used TNSNAMES adapter to resolve the alias
Attempting to contact (DESCRIPTION = (ADDRESS_LIST = (ADDRESS = (PROTOCOL =
TCP)(HOST = stdby01.oracle.com)(PORT = 1521))) (CONNECT_DATA = (SERVICE_NAME
= DXDB.oracle.com)))
OK (10 msec)
[oracle@stdby01 admin]$ tnsping DHDB

TNS Ping Utility for Linux: Version 12.1.0.1.0 - Production on 12-NOV-2017
06:06:19

Copyright (c) 1997, 2013, Oracle.  All rights reserved.

Used parameter files:
```

```
Used TNSNAMES adapter to resolve the alias
Attempting to contact (DESCRIPTION = (ADDRESS_LIST = (ADDRESS = (PROTOCOL =
TCP)(HOST = prmy01.oracle.com)(PORT = 1521))) (CONNECT_DATA = (SERVICE_NAME
= DHDB.oracle.com)))
OK (0 msec)
[oracle@stdby01 admin]$
```

Prepare the NOMOUNT initDXDB.ora Parameter File

The parameter file (PFILE) is prepared to startup the standby instance (auxiliary) in NOMOUNT state. This step creates the control file specified by the parameter CONTROL_FILES by RMAN DUPLICATE command.

View and run the script $HOME/admin/s6_copy_initDXDB.sh

Complete

```
[oracle@stdby01 ~]$ cd $ORACLE_HOME/dbs
[oracle@stdby01 dbs]$ vi initDXDB.ora
```

```
db_name=DXDB
control_files='/u01/app/oracle/oradata/DXDB/control01.ctl'
log_file_name_convert='/u01/app/oracle/oradata/DHDB',
'/u01/app/oracle/oradata/DXDB'
db_file_name_convert='/u01/app/oracle/oradata/DHDB',
'/u01/app/oracle/oradata/DXDB'
```

Create the Auxiliary Instance Sub-Directories

Prior to running RMAN DUPLICATE command to create the standby database, create the directories structure. You must be login as 'oracle' to avoid permission related issues.

View and run the script $HOME/admin/s7_create_dir.sh

Complete

```
$mkdir -p /u01/app/oracle/oradata/DXDB
$mkdir -p /u01/app/oracle/oradata/DXDB/arch
$mkdir -p /u01/app/oracle/admin/DXDB/adump
$mkdir -p /u01/app/oracle/admin/DXDB/bdump
$mkdir -p /u01/app/oracle/admin/DXDB/cdump
$mkdir -p /u01/app/oracle/admin/DXDB/dpdump
$mkdir -p /u01/app/oracle/admin/DXDB/pfile
$mkdir -p /u01/app/oracle/admin/DXDB/scripts
$mkdir -p /u01/app/oracle/admin/DXDB/udump
$mkdir -p /u01/app/oracle/fast_recovery_area
```

Startup the Auxiliary Instance in NOMOUNT State

Before running RMAN DUPLICATE command, startup the auxiliary instance in NOMOUNT state using the PFILE – initDXDB.ora.

View and run the script $HOME/admin/s8_startup_nomount.sh

Complete

```
[oracle@stdby01 ~]$ . oraenv
ORACLE_SID = [oracle] ? DXDB
The Oracle base remains unchanged with value /u01/app/oracle
[oracle@stdby01 ~]$ sqlplus /nolog

SQL*Plus: Release 12.1.0.1.0 Production on Sat Nov 4 00:05:05 2017

Copyright (c) 1982, 2013, Oracle.  All rights reserved.

SQL> conn / as sysdba
Connected to an idle instance.
SQL> startup pfile='$ORACLE_HOME/dbs/initDXDB.ora' nomount
ORACLE instance started.

Total System Global Area          217157632 bytes
Fixed Size                          2286656 bytes
Variable Size                     159386560 bytes
Database Buffers                   50331648 bytes
Redo Buffers                        5152768 bytes
SQL> exit
Disconnected from Oracle Database 12c Enterprise Edition Release 12.1.0.1.0
- 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real Application Testing
options
[oracle@stdby01 dbs]$
```

Create the Auxiliary Instance Database

This is executed as two steps. First create RMAN script for the DUPLICATE command. Then, run the shell script to execute RMAN script. RMAN script must allocate an auxiliary channel. The allocation of multiple channels enhances the DUPLICTE command performance through parallelism.

View and run the script $HOME/admin/s10_dupDB.sh

Complete

```
[oracle@stdby01 dbs]$ cd $HOME/admin
[oracle@stdby01 admin]$ vi s9_dupDB.rmn
```

```
run{
allocate channel c1 type disk;
allocate channel c2 type disk;
allocate channel c3 type disk;
allocate channel c4 type disk;
```

```
allocate auxiliary channel c5 type disk;
duplicate target database for standby from active database
spfile
   parameter_value_convert '/DHDB', '/DXDB'
   set db_unique_name='DXDB'
   set db_file_name_convert='/DHDB/', '/DXDB/'
   set log_file_name_convert='/DHDB/', '/DXDB/'
   set control_files='/u01/app/oracle/oradata/DXDB/control01.ctl'
   set log_archive_max_processes='5'
   set fal_client='DXDB'
   set fal_server='DHDB'
   set standby_file_management='AUTO'
   set log_archive_config='dg_config=(DHDB, DXDB)'
   set log_archive_dest_2='service=DHDB ASYNC valid_for=(online_logfile,
primary_role) db_unique_name=DHDB';
}
```

Run the script, using RMAN command line while connected to the target database and the auxiliary instance.

Complete

```
[oracle@stdby01 admin]$ rman target=sys/oracle@DHDB
auxiliary=sys/oracle@DXDB @/home/oracle/admin/s9_dupDB.rmn

Recovery Manager: Release 12.1.0.1.0 - Production on Wed Nov 15 21:28:21
2017

Copyright (c) 1982, 2013, Oracle and/or its affiliates.  All rights
reserved.

connected to target database: DHDB (DBID=2763595033)
connected to auxiliary database: DXDB (not mounted)

RMAN> run{
2> allocate channel c1 type disk;
3> allocate channel c2 type disk;
4> allocate channel c3 type disk;
5> allocate channel c4 type disk;
6> allocate auxiliary channel c5 type disk;
7> duplicate target database for standby from active database
8> spfile
9>    parameter_value_convert '/DHDB', '/DXDB'
10>    set db_unique_name='DXDB'
11>    set db_file_name_convert='/DHDB/', '/DXDB/'
12>    set log_file_name_convert='/DHDB/', '/DXDB/'
13>    set control_files='/u01/app/oracle/oradata/DXDB/control01.ctl'
14>    set log_archive_max_processes='5'
15>    set fal_client='DXDB'
16>    set fal_server='DHDB'
17>    set standby_file_management='AUTO'
18>    set log_archive_config='dg_config=(DHDB, DXDB)'
19>    set log_archive_dest_2='service=DHDB ASYNC valid_for=(online_logfile,
primary_role) db_unique_name=DHDB';
20> }
21>
22>
```

```
using target database control file instead of recovery catalog
allocated channel: c1
channel c1: SID=44 device type=DISK

allocated channel: c2
channel c2: SID=45 device type=DISK

allocated channel: c3
channel c3: SID=46 device type=DISK

allocated channel: c4
channel c4: SID=50 device type=DISK

allocated channel: c5
channel c5: SID=20 device type=DISK

Starting Duplicate Db at 15-NOV-17

contents of Memory Script:
{
   backup as copy reuse
   targetfile  '/u01/app/oracle/product/12.1.0/dbhome_1/dbs/orapwDHDB'
auxiliary format
 '/u01/app/oracle/product/12.1.0/dbhome 1/dbs/orapwDXDB'   targetfile
 '/u01/app/oracle/product/12.1.0/dbhome_1/dbs/spfileDHDB.ora' auxiliary
format
 '/u01/app/oracle/product/12.1.0/dbhome_1/dbs/spfileDXDB.ora'   ;
   sql clone "alter system set spfile=
''/u01/app/oracle/product/12.1.0/dbhome_1/dbs/spfileDXDB.ora''";
}
executing Memory Script

Starting backup at 15-NOV-17
Finished backup at 15-NOV-17

sql statement: alter system set spfile=
''/u01/app/oracle/product/12.1.0/dbhome_1/dbs/spfileDXDB.ora''

contents of Memory Script:
{
   sql clone "alter system set  audit file dest =
 ''/u01/app/oracle/admin/DXDB/adump'' comment=
 '''' scope=spfile";
   sql clone "alter system set  db_unique_name =
 ''DXDB'' comment=
 '''' scope=spfile";
   sql clone "alter system set  db file name convert =
 ''/DHDB/'', ''/DXDB/'' comment=
 '''' scope=spfile";
   sql clone "alter system set  log_file_name_convert =
 ''/DHDB/'', ''/DXDB/'' comment=
 '''' scope=spfile";
   sql clone "alter system set  control files =
 ''/u01/app/oracle/oradata/DXDB/control01.ctl'' comment=
 '''' scope=spfile";
   sql clone "alter system set  log_archive_max_processes =
 5 comment=
 '''' scope=spfile";
   sql clone "alter system set  fal_client =
```

```
  ''DXDB'' comment=
  '''' scope=spfile";
    sql clone "alter system set  fal_server =
  ''DHDB'' comment=
  '''' scope=spfile";
    sql clone "alter system set  standby_file_management =
  ''AUTO'' comment=
  '''' scope=spfile";
    sql clone "alter system set  log_archive_config =
  ''dg_config=(DHDB, DXDB)'' comment=
  '''' scope=spfile";
    sql clone "alter system set  log_archive_dest_2 =
  ''service=DHDB ASYNC valid for=(online logfile, primary role)
db_unique_name=DHDB'' comment=
  '''' scope=spfile";
    shutdown clone immediate;
    startup clone nomount;
}
executing Memory Script

sql statement: alter system set  audit_file_dest =
''/u01/app/oracle/admin/DXDB/adump'' comment= '''' scope=spfile

sql statement: alter system set  db unique name =  ''DXDB'' comment= ''''
scope=spfile

sql statement: alter system set  db_file_name_convert =  ''/DHDB/'',
''/DXDB/'' comment= '''' scope=spfile

sql statement: alter system set  log file name convert =  ''/DHDB/'',
''/DXDB/'' comment= '''' scope=spfile

sql statement: alter system set  control_files =
''/u01/app/oracle/oradata/DXDB/control01.ctl'' comment= '''' scope=spfile

sql statement: alter system set  log archive max processes =  5 comment=
'''' scope=spfile

sql statement: alter system set  fal_client =  ''DXDB'' comment= ''''
scope=spfile

sql statement: alter system set  fal server =  ''DHDB'' comment= ''''
scope=spfile

sql statement: alter system set  standby_file_management =  ''AUTO''
comment= '''' scope=spfile

sql statement: alter system set  log_archive_config =  ''dg_config=(DHDB,
DXDB)'' comment= '''' scope=spfile

sql statement: alter system set  log archive dest 2 =  ''service=DHDB ASYNC
valid_for=(online_logfile, primary_role) db_unique_name=DHDB'' comment= ''''
scope=spfile

Oracle instance shut down

connected to auxiliary database (not started)
Oracle instance started
```

```
Total System Global Area     839282688 bytes

Fixed Size                     2293928 bytes
Variable Size                599789400 bytes
Database Buffers             234881024 bytes
Redo Buffers                   2318336 bytes
allocated channel: c5
channel c5: SID=20 device type=DISK

contents of Memory Script:
{
   backup as copy current controlfile for standby auxiliary format
'/u01/app/oracle/oradata/DXDB/control01.ctl';
}
executing Memory Script

Starting backup at 15-NOV-17
channel c1: starting datafile copy
copying standby control file
output file name=/u01/app/oracle/product/12.1.0/dbhome_1/dbs/snapcf_DHDB.f
tag=TAG20171115T212832
channel c1: datafile copy complete, elapsed time: 00:00:01
Finished backup at 15-NOV-17

contents of Memory Script:
{
   sql clone 'alter database mount standby database';
}
executing Memory Script

sql statement: alter database mount standby database

contents of Memory Script:
{
   set newname for tempfile  1 to
 "/u01/app/oracle/oradata/DXDB/datafile/o1 mf temp dzpdqcfp .tmp";
   switch clone tempfile all;
   set newname for datafile  1 to
 "/u01/app/oracle/oradata/DXDB/datafile/o1_mf_system_dzpdocm7_.dbf";
   set newname for datafile  3 to
 "/u01/app/oracle/oradata/DXDB/datafile/o1 mf sysaux dzpdn84c .dbf";
   set newname for datafile  4 to
 "/u01/app/oracle/oradata/DXDB/datafile/o1_mf_undotbs1_dzpdptgh_.dbf";
   set newname for datafile  6 to
 "/u01/app/oracle/oradata/DXDB/datafile/o1_mf_users_dzpdps52_.dbf";
   backup as copy reuse
   datafile  1 auxiliary format
 "/u01/app/oracle/oradata/DXDB/datafile/o1_mf_system_dzpdocm7_.dbf"
datafile
 3 auxiliary format
 "/u01/app/oracle/oradata/DXDB/datafile/o1 mf sysaux dzpdn84c .dbf"
datafile
 4 auxiliary format
 "/u01/app/oracle/oradata/DXDB/datafile/o1_mf_undotbs1_dzpdptgh_.dbf"
datafile
 6 auxiliary format
 "/u01/app/oracle/oradata/DXDB/datafile/o1 mf users dzpdps52 .dbf"   ;
   sql 'alter system archive log current';
}
```

```
executing Memory Script

executing command: SET NEWNAME

renamed tempfile 1 to
/u01/app/oracle/oradata/DXDB/datafile/o1_mf_temp_dzpdqcfp_.tmp in control
file

executing command: SET NEWNAME

executing command: SET NEWNAME

executing command: SET NEWNAME

executing command: SET NEWNAME

Starting backup at 15-NOV-17
channel c1: starting datafile copy
input datafile file number=00001
name=/u01/app/oracle/oradata/DHDB/datafile/o1_mf_system_dzpdocm7_.dbf
channel c2: starting datafile copy
input datafile file number=00003
name=/u01/app/oracle/oradata/DHDB/datafile/o1 mf sysaux dzpdn84c .dbf
channel c3: starting datafile copy
input datafile file number=00006
name=/u01/app/oracle/oradata/DHDB/datafile/o1_mf_users_dzpdps52_.dbf
channel c4: starting datafile copy
input datafile file number=00004
name=/u01/app/oracle/oradata/DHDB/datafile/o1_mf_undotbs1_dzpdptgh_.dbf
output file
name=/u01/app/oracle/oradata/DXDB/datafile/o1_mf_undotbs1_dzpdptgh_.dbf
tag=TAG20171115T212838
channel c4: datafile copy complete, elapsed time: 00:00:07
output file
name=/u01/app/oracle/oradata/DXDB/datafile/o1_mf_system_dzpdocm7_.dbf
tag=TAG20171115T212838
channel c1: datafile copy complete, elapsed time: 00:00:15
output file
name=/u01/app/oracle/oradata/DXDB/datafile/o1_mf_sysaux_dzpdn84c_.dbf
tag=TAG20171115T212838
channel c2: datafile copy complete, elapsed time: 00:00:15
output file
name=/u01/app/oracle/oradata/DXDB/datafile/o1_mf_users_dzpdps52_.dbf
tag=TAG20171115T212838
channel c3: datafile copy complete, elapsed time: 00:00:15
Finished backup at 15-NOV-17

sql statement: alter system archive log current

contents of Memory Script:
{
   switch clone datafile all;
}
executing Memory Script

datafile 1 switched to datafile copy
input datafile copy RECID=1 STAMP=960154133 file
name=/u01/app/oracle/oradata/DXDB/datafile/o1_mf_system_dzpdocm7_.dbf
datafile 3 switched to datafile copy
```

```
input datafile copy RECID=2 STAMP=960154133 file
name=/u01/app/oracle/oradata/DXDB/datafile/o1_mf_sysaux_dzpdn84c_.dbf
datafile 4 switched to datafile copy
input datafile copy RECID=3 STAMP=960154133 file
name=/u01/app/oracle/oradata/DXDB/datafile/o1 mf undotbs1 dzpdptgh .dbf
datafile 6 switched to datafile copy
input datafile copy RECID=4 STAMP=960154133 file
name=/u01/app/oracle/oradata/DXDB/datafile/o1_mf_users_dzpdps52_.dbf
Finished Duplicate Db at 15-NOV-17
released channel: c1
released channel: c2
released channel: c3
released channel: c4
released channel: c5

Recovery Manager complete.
[oracle@stdby01 admin]$
```



Verify the Cloned Physical Standby Database is Successful

Verify the cloning is successful by logging to the standby database and query V$DATABASE dictionary view to verify the database role and the open mode.

View and run the script $HOME/admin/s11_verify.sh




```
[oracle@stdby01 admin]$ sqlplus /nolog

SQL*Plus: Release 12.1.0.1.0 Production on Wed Nov 15 21:31:01 2017

Copyright (c) 1982, 2013, Oracle.  All rights reserved.

SQL> conn / as sysdba
Connected.
SQL> select database_role from v$database;

DATABASE_ROLE
----------------
PHYSICAL STANDBY

SQL> select open_mode from v$database;

OPEN_MODE
--------------------
MOUNTED

SQL>
```

P Configure the Primary Instance

Add the instance database parameters to support Oracle Data Guard configuration. The configuration parameters enable the following:

- Automatic data guard broker startup
- Broker configuration
- Specify the primary database (DHDB) and the standby database (DXDB) managed by the broker
- Log archive settings for redo transport and protection mode
- File conversion parameters

View and run the script $HOME/admin/s12_config_prmy.sh

Complete

```
[oracle@prmy01 admin]$ sqlplus /nolog

SQL*Plus: Release 12.1.0.1.0 Production on Wed Nov 29 09:48:29 2017

Copyright (c) 1982, 2013, Oracle.  All rights reserved.

SQL> conn / as sysdba
Connected.
SQL> alter system set dg_broker_start=true scope=both;

System altered.

SQL> alter system set
dg_broker_config_file1='/u01/app/oracle/product/12.1.0/dbhome_1/dbs/DHDB_dg1
.conf' scope=spfile;

System altered.

SQL> alter system set
dg_broker_config_file2='/u01/app/oracle/product/12.1.0/dbhome_1/dbs/DHDB_dg2
.conf' scope=spfile;

System altered.

SQL> alter system set log_archive_format='DHDB_%t_%s_%r.arc' scope=spfile;

System altered.

SQL> alter system set log_archive_config='DG_CONFIG=(DHDB, DXDB)'
scope=both;

System altered.

SQL> alter system set log_archive_max_processes=4 scope=both;

System altered.

SQL> alter system set fal_server=DXDB scope=spfile;
```

```
System altered.

SQL> alter system set fal_client=DHDB scope=spfile;

System altered.

SQL> alter system set log_file_name_convert='/u01/app/oracle/oradata/DXDB',
'/u01/app/oracle/oradata/DHDB' scope=spfile;

System altered.

SQL> alter system set db_file_name_convert='/u01/app/oracle/oradata/DXDB',
'/u01/app/oracle/oradata/DHDB' scope=spfile;

System altered.

SQL> alter system set standby_file_management=auto scope=spfile;

System altered.

SQL>
```

Restart the database instance to enable static parameters.

```
SQL> shutdown immediate
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> startup
ORACLE instance started.

Total System Global Area          839282688 bytes
Fixed Size                          2293928 bytes
Variable Size                     599789400 bytes
Database Buffers                  234881024 bytes
Redo Buffers                        2318336 bytes
Database mounted.
Database opened.
SQL>
```



Configure the Physical Standby Database Instance

Add the instance parameters to support Oracle Data Guard configuration. The configuration parameters enable the following:

- Automatic data guard broker startup
- Broker configuration
- Specify the primary database (DHDB) and the standby database (DXDB) managed by the broker
- Log archive settings
- File conversion parameters

View and run the script $HOME/admin/s13_config_stdby.sh

Complete

```
SQL> conn / as sysdba
Connected.
SQL> alter system set dg_broker_start=true scope=both;

System altered.

SQL> alter system set
dg_broker_config_file1='/u01/app/oracle/product/12.1.0/dbhome_1/dbs/DXDB_dg1
.conf' scope=spfile;

System altered.

SQL> alter system set
dg_broker_config_file2='/u01/app/oracle/product/12.1.0/dbhome_1/dbs/DXDB_dg2
.conf' scope=spfile;

System altered.

SQL> alter system set log_archive_format='DXDB_%t_%s_%r.arc' scope=spfile;

System altered.

SQL> alter system set log_archive_config='DG_CONFIG=(DHDB, DXDB)'
scope=both;

System altered.

SQL> alter system set log_archive_dest_2='service=DHDB
valid_for=(online_logfile, primary_role) db_unique_name=DHDB' scope=both;

System altered.

SQL> alter system set log_archive_dest_state_2=enable scope=both;

System altered.

SQL> alter system set log_archive_max_processes=4 scope=both;

System altered.

SQL> alter system set fal_server=DHDB scope=spfile;

System altered.

SQL> alter system set fal_client=DXDB scope=spfile;

System altered.

SQL> alter system set
log_file_name_convert='/u01/app/oracle/oradata/DHDB','/u01/app/oracle/oradat
a/DXDB' scope=spfile;

System altered.
```

```
SQL> alter system set db_file_name_convert='/u01/app/oracle/oradata/DHDB',
'/u01/app/oracle/oradata/DXDB' scope=spfile;

System altered.

SQL> alter system set standby_file_management=auto scope=spfile;

System altered.

SQL>
```

Restart the physical standby database and place it in recovery mode with disconnect option.

```
SQL> shutdown immediate
ORA-01109: database not open


Database dismounted.
ORACLE instance shut down.
SQL> startup
ORACLE instance started.

Total System Global Area          839282688 bytes
Fixed Size                          2293928 bytes
Variable Size                     599789400 bytes
Database Buffers                  234881024 bytes
Redo Buffers                        2318336 bytes
Database mounted.
Database opened.
SQL> alter database recover managed standby database disconnect from
session;

Database altered.

SQL>
```



Add standby Redo Log to the Standby Instance

To support real-time apply and bi-directional switchover and failover commands create the standby online redo log files. The standby redo log files should be slightly larger than the corresponding online redo log files. The number of standby redo log groups should be one more than the redo log groups. For example, if the primary database has 3 groups, the standby should have 4 groups. Query v$log to verify the number and size of online redo log groups.

```
SQL> select group#, bytes/1024/1024 "SIZE MB"
  2  from v$log;

    GROUP#    SIZE MB
---------- ----------
         1         50
```

```
         2         50
         3         50

SQL>
```

View and run the script $HOME/admin/s14_create_stdby_redo.sh

Complete

```
SQL> conn / as sysdba
Connected.
SQL> recover managed standby database cancel;
Media recovery complete.
SQL> alter database add standby logfile group 4
'/u01/app/oracle/oradata/DXDB/redo04.log' size 60M;

Database altered.

SQL> alter database add standby logfile group 5
'/u01/app/oracle/oradata/DXDB/redo05.log' size 60M;

Database altered.

SQL> alter database add standby logfile group 6
'/u01/app/oracle/oradata/DXDB/redo06.log' size 60M;

Database altered.

SQL> alter database add standby logfile group 7
'/u01/app/oracle/oradata/DXDB/redo07.log' size 60M;

Database altered.

SQL> alter database recover managed standby database disconnect from
session;

Database altered.

SQL>
```

Add Standby Redo Log to the Primary Instance

To support real-time apply and bi-directional switchover and failover commands create the standby online redo log files. The standby redo log files should be slightly larger than the corresponding online redo log files. The number of standby redo log groups should be one more than the redo log groups. For example, if the primary database has 3 groups, the standby should have 4 groups. Query v$log to verify the number and size of online redo log groups.

```
SQL> select group#, bytes/1024/1024 "SIZE MB"
  2  from v$log;

    GROUP#    SIZE MB
---------- ----------
         1         50
```

```
         2          50
         3          50

SQL>
```

View and run the script $HOME/admin/s15_create_stdby_redo.sh

Complete

```
SQL> conn / as sysdba
Connected.
SQL> alter database add standby logfile group 4
'/u01/app/oracle/oradata/DHDB/redo04.log' size 60M;

Database altered.

SQL> alter database add standby logfile group 5
'/u01/app/oracle/oradata/DHDB/redo05.log' size 60M;

Database altered.

SQL> alter database add standby logfile group 6
'/u01/app/oracle/oradata/DHDB/redo06.log' size 60M;

Database altered.

SQL> alter database add standby logfile group 7
'/u01/app/oracle/oradata/DHDB/redo07.log' size 60M;

Database altered.

SQL>
```

P Clear Archive Destination

Because of the UNREGISTER attribute of the LOG_ARCHIVE_DEST_2 parameter, clear the archive log destination.

View and run the script $HOME/admin/s18_alter.sh

Complete

```
SQL> conn / as sysdba
Connected.
SQL> alter system set log_archive_dest_2='' scope=both;

System altered.

SQL> exit
Disconnected from Oracle Database 12c Enterprise Edition Release 12.1.0.1.0
- 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real Application Testing
options
[oracle@prmy01 admin]$
```

S Clear Archive Destination

Because of the UNREGISTER attribute of the LOG_ARCHIVE_DEST_2 parameter, clear the archive log destination.

View and run the script $HOME/admin/s17_alter.sh

Complete

```
SQL> conn / as sysdba
Connected.
SQL> alter system set log_archive_dest_2='' scope=both;

System altered.

SQL>
```

P Create and Enable Data Guard Broker Configuration

The data guard broker is configured and enabled via DGMGRL command line. The configuration name DG_CONF, which include the primary database (DHDB) and the standby database (DXDB).

View and run the script $HOME/admin/s16_dg_broker_conf.sh

Complete

```
[oracle@prmy01 ~]$ dgmgrl
DGMGRL for Linux: Version 12.1.0.1.0 - 64bit Production

Copyright (c) 2000, 2012, Oracle. All rights reserved.

Welcome to DGMGRL, type "help" for information.
DGMGRL> connect sys/oracle
Connected as SYSDG.
DGMGRL> create configuration 'DG_CONF' as primary database is 'DHDB' connect
identifier is 'DHDB';
Configuration "DG_CONF" created with primary database "DHDB"
DGMGRL> enable configuration;
Enabled.
DGMGRL> add database 'DXDB' as connect identifier is 'DXDB';
Database "DXDB" added
DGMGRL> enable database 'DHDB';
Enabled.
DGMGRL> enable database 'DXDB';
Enabled.
DGMGRL>
```

P Display and Review Data Guard Configuration Details for DHDB

Ensure the database properties are correct. Test the connection using SQL*Plus to ensure the StaticConnectIdentifier is configured correctly. Copy-Paste the connect string with database user SYSTEM.

```
SQL> conn
system/oracle_4U@(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=prmy01.oracle.com
)(PORT=1521))(CONNECT_DATA=(SERVICE_NAME=DHDB_DGMGRL.oracle.com)(INSTANCE_NA
ME=DHDB)(SERVER=DEDICATED)))
Connected.
SQL>
```

Complete

```
DGMGRL> show database verbose 'DHDB';

Database - DHDB

  Role:              PRIMARY
  Intended State:    TRANSPORT-ON
  Instance(s):
    DHDB

  Properties:
    DGConnectIdentifier             = 'DHDB'
    ObserverConnectIdentifier       = ''
    LogXptMode                      = 'ASYNC'
    RedoRoutes                      = ''
    DelayMins                       = '0'
    Binding                         = 'optional'
    MaxFailure                      = '0'
    MaxConnections                  = '1'
    ReopenSecs                      = '300'
    NetTimeout                      = '30'
    RedoCompression                 = 'DISABLE'
    LogShipping                     = 'ON'
    PreferredApplyInstance          = ''
    ApplyInstanceTimeout            = '0'
    ApplyLagThreshold               = '0'
    TransportLagThreshold           = '0'
    TransportDisconnectedThreshold  = '30'
    ApplyParallel                   = 'AUTO'
    StandbyFileManagement           = 'AUTO'
    ArchiveLagTarget                = '0'
    LogArchiveMaxProcesses          = '4'
    LogArchiveMinSucceedDest        = '1'
    DbFileNameConvert               = '/u01/app/oracle/oradata/DXDB,
/u01/app/oracle/oradata/DHDB'
    LogFileNameConvert              = '/u01/app/oracle/oradata/DXDB,
/u01/app/oracle/oradata/DHDB'
    FastStartFailoverTarget         = ''
    InconsistentProperties          = '(monitor)'
    InconsistentLogXptProps         = '(monitor)'
```

```
    SendQEntries                    = '(monitor)'
    LogXptStatus                    = '(monitor)'
    RecvQEntries                    = '(monitor)'
    StaticConnectIdentifier         =
'(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=prmy01.oracle.com)(PORT=1521))(CO
NNECT_DATA=(SERVICE_NAME=DHDB_DGMGRL.oracle.com)(INSTANCE_NAME=DHDB)(SERVER=
DEDICATED)))'
    StandbyArchiveLocation          = 'USE_DB_RECOVERY_FILE_DEST'
    AlternateLocation               = ''
    LogArchiveTrace                 = '0'
    LogArchiveFormat                = 'DHDB %t %s %r.arc'
    TopWaitEvents                   = '(monitor)'

Database Status:
SUCCESS

DGMGRL>
```

Display and Review Data Guard Configuration Details for DXDB

Ensure the database properties are correct. Ensure the database properties are correct. Test the connection using SQL*Plus to ensure the StaticConnectIdentifier is configured correctly. Copy-Paste the connect string with database user SYSTEM.

```
SQL> conn
system/oracle_4U@(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=stdby01.oracle.co
m)(PORT=1521))(CONNECT_DATA=(SERVICE_NAME=DXDB_DGMGRL.oracle.com)(INSTANCE_N
AME=DXDB)(SERVER=DEDICATED)))
Connected.
SQL>
```

Complete

```
DGMGRL> show database verbose 'DXDB';

Database - DXDB

  Role:               PHYSICAL STANDBY
  Intended State:     APPLY-ON
  Transport Lag:      0 seconds (computed 0 seconds ago)
  Apply Lag:          0 seconds (computed 0 seconds ago)
  Apply Rate:         0 Byte/s
  Real Time Query:    ON
  Instance(s):
    DXDB

  Properties:
    DGConnectIdentifier             = 'DXDB'
    ObserverConnectIdentifier       = ''
    LogXptMode                      = 'ASYNC'
    RedoRoutes                      = ''
    DelayMins                       = '0'
    Binding                         = 'optional'
```

```
    MaxFailure                      = '0'
    MaxConnections                  = '1'
    ReopenSecs                      = '300'
    NetTimeout                      = '30'
    RedoCompression                 = 'DISABLE'
    LogShipping                     = 'ON'
    PreferredApplyInstance          = ''
    ApplyInstanceTimeout            = '0'
    ApplyLagThreshold               = '0'
    TransportLagThreshold           = '0'
    TransportDisconnectedThreshold  = '30'
    ApplyParallel                   = 'AUTO'
    StandbyFileManagement           = 'AUTO'
    ArchiveLagTarget                = '0'
    LogArchiveMaxProcesses          = '4'
    LogArchiveMinSucceedDest        = '1'
    DbFileNameConvert               = '/u01/app/oracle/oradata/DHDB,
/u01/app/oracle/oradata/DXDB'
    LogFileNameConvert              = '/u01/app/oracle/oradata/DHDB,
/u01/app/oracle/oradata/DXDB'
    FastStartFailoverTarget         = ''
    InconsistentProperties          = '(monitor)'
    InconsistentLogXptProps         = '(monitor)'
    SendQEntries                    = '(monitor)'
    LogXptStatus                    = '(monitor)'
    RecvQEntries                    = '(monitor)'
    StaticConnectIdentifier         =
'(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=stdby01.oracle.com)(PORT=1521))(C
ONNECT_DATA=(SERVICE_NAME=DXDB_DGMGRL.oracle.com)(INSTANCE_NAME=DXDB)(SERVER
=DEDICATED)))'
    StandbyArchiveLocation          = 'USE_DB_RECOVERY_FILE_DEST'
    AlternateLocation               = ''
    LogArchiveTrace                 = '0'
    LogArchiveFormat                = 'DXDB %t %s %r.arc'
    TopWaitEvents                   = '(monitor)'

Database Status:
SUCCESS

DGMGRL>
```

P Perform Planned Switchover to Current Standby Site

While connected to the data guard using DGMGRL, perform a switchover to the current standby database – DXDB. In this case DXDB role is the primary and DHDB role is the standby.

View and run the script $HOME/admin/s19_switchover.sh

Complete

```
DGMGRL> connect sys/oracle
Connected as SYSDG.
DGMGRL> switchover to 'DXDB';
Performing switchover NOW, please wait...
Operation requires a connection to instance "DXDB" on database "DXDB"
```

```
Connecting to instance "DXDB"...
Connected as SYSDBA.
New primary database "DXDB" is opening...
Operation requires startup of instance "DHDB" on database "DHDB"
Starting instance "DHDB"...
ORACLE instance started.
Database mounted.
Database opened.
Switchover succeeded, new primary is "DXDB"
DGMGRL>
```

Perform Planned Switchover to Current Standby Site

While connected to the data guard configuration using DGMGRL, perform a switchover to the current standby database – DHDB. In this case DHDB role is the primary and DXDB role is the standby.

View and run the script $HOME/admin/s20_switchover.sh

Complete

```
DGMGRL> switchover to 'DHDB';
Performing switchover NOW, please wait...
Operation requires a connection to instance "DHDB" on database "DHDB"
Connecting to instance "DHDB"...
Connected as SYSDBA.
New primary database "DHDB" is opening...
Operation requires startup of instance "DXDB" on database "DXDB"
Starting instance "DXDB"...
ORACLE instance started.
Database mounted.
Database opened.
Switchover succeeded, new primary is "DHDB"
DGMGRL> exit
[oracle@prmy01 admin]$
```

Verify the Primary Database Role

At this point; the primary database role is PRIMARY. Verify this by querying the data dictionary view v$database.

View and run the script $HOME/admin/s20a_verify_prmy.sh

Complete

```
[oracle@prmy01 admin]$ sqlplus /nolog

SQL*Plus: Release 12.1.0.1.0 Production on Wed Nov 29 10:06:41 2017

Copyright (c) 1982, 2013, Oracle.  All rights reserved.

SQL> conn / as sysdba
Connected.
SQL> select database_role from v$database;
```

```
DATABASE_ROLE
----------------
PRIMARY

SQL>
```

Verify the Standby Database Role

At this point; the standby database role is PHYSICAL STANDBY. Verify this by querying the data dictionary view v$database.

View and run the script $HOME/admin/s20b_verify_stdby.sh

Complete

```
SQL> conn / as sysdba
Connected.
SQL> select database_role from v$database;

DATABASE_ROLE
----------------
PHYSICAL STANDBY

SQL>
```

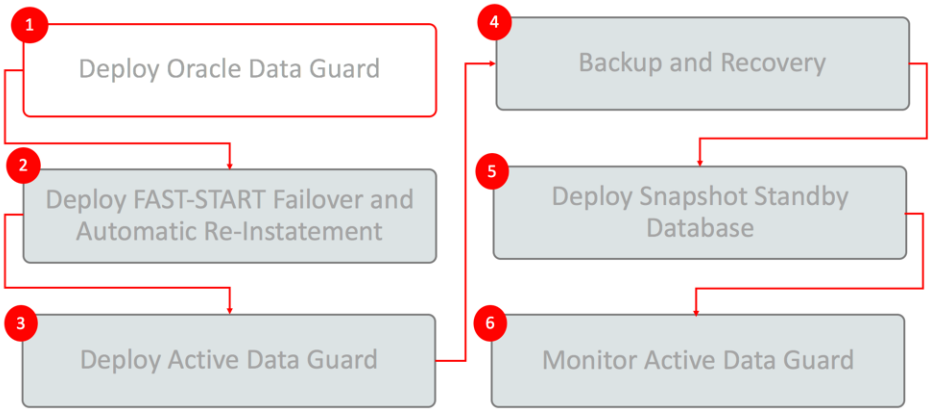
Warning

If you're unable to successfully switchover bi-directionally, please DO NOT continues and seeks help from the instructor. The next sections depend entirely on successful completion of this section.

Good luck…

Congratulation!

You've successfully completed Part I: Deploy Oracle Data Guard Workshop. This is a pre-requisite to the next workshop – Deploy FAST-START failover and Automatic Re-Instatement.



Part II: Deploy FAST-START Failover and Re-Instatement

This part configures the data guard FAST-START failover and re-instatement feature. The purpose is to automate failover operation and primary database re-instatement. Configuring of Fast Recovery Area and Flashback database is a pre-requisite.

Tasks

Table 7 lists the tasks associated with Part II, Deploy FAST-START Failover and Re-Instatement.

Task#	Description
1	Configure Fast Recovery Area
2	Configure Fast Recovery Area
3	Enable Database Flashback
4	Enable Database Flashback
5	Verify Data Guard Broker Configuration
6	Configure SQLNET Service
7	Verify Static Instance Registration
8	Verify Static Instance Registration
9	Display Default FAST-START Failover Settings
10	Specify FAST-START Target Standby Database
11	Set the Protection Mode
12	Set the FastStartFailoverThreshold Property
13	Set Additional Database Properties
14	Set Additional Fast-Start Failover Conditions
15	Display and Verify Current FAST-START Failover Properties
16	Enable the FAST_START Failover
17	Start the Observer Server
18	Shutdown ABORT Primary Database
19	Monitor the Observer Terminal
20	Display Current Data Guard Status
21	Automatic Re-Instatement of DHDB Database
22	Show Current Data Guard Status
23	Stop the Observer Server and Disable FAST-START Failover
24	Switchover to DHDB
25	Show Current Data Guard Status

Table 7: FAST-START failover and re-instatement tasks

Architecture

Figure 9 shows the FAST-START failover building blocks. It composed of three major components:

- The primary database
- The standby database
- The observer Server

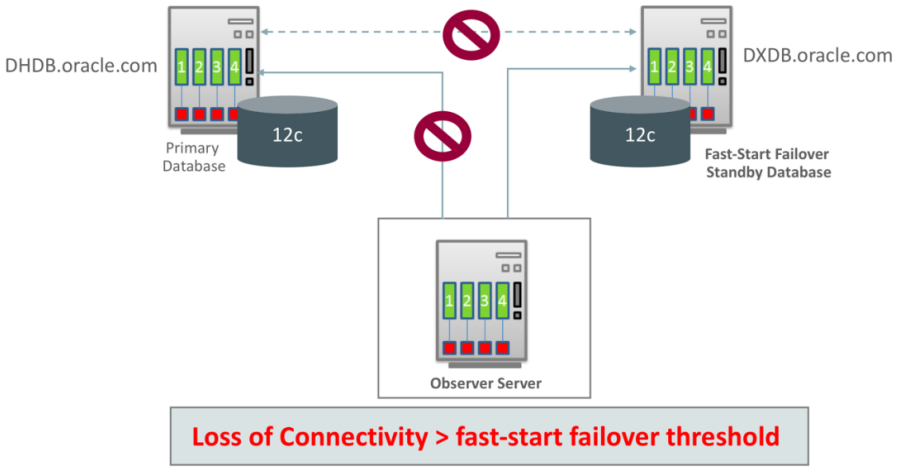


Figure 9: FAST-START failover Architecture

Fast-start failover enables Data Guard to rapidly and automatically fail over to a previously chosen standby database without requiring manual intervention. This feature increases the availability of your database in the event of a disaster by reducing the need for you to perform a failover operation manually. Fast-start failover occurs when any of the following conditions are met:

Loss of connectivity between both the primary database and the observer—and between the primary database and the fast-start failover target standby database—exceeds the fast-start failover threshold.

The database health-check mechanism determines any of the following:

- A data file is offline because of a write error.
- Dictionary corruption of a critical database object occurs.
- Control file is permanently damaged because of a disk failure.
- LGWR is unable to write to any member of the log group due to I/O error.
- Archive is unable to archive a redo log because the device is full or unavailable

P Configure Fast Recovery Area

The Fast Recovery Area (FRA) is a unified storage location for all recovery-related files and activities in an Oracle database. All files that are needed to completely recover a database from a media failure are part of the Fast Recovery Area.

View and run the script $HOME/admin/s21_setFRA.sh

☐ Complete

```
[oracle@prmy01 ~]$ sqlplus /nolog

SQL*Plus: Release 12.1.0.1.0 Production on Wed Nov 29 11:18:31 2017

Copyright (c) 1982, 2013, Oracle.  All rights reserved.

SQL> conn / as sysdba
Connected.
SQL> show parameters db_recovery_file_dest

NAME                                 TYPE        VALUE
------------------------------------ ----------- ------------------------------
---
db_recovery_file_dest                     string
/u01/app/oracle/fast_recovery_
                                                 area
db recovery file dest size      big integer 4800M
SQL> alter system set db_recovery_file_dest_size=9000M scope=both;

System altered.

SQL>
```

S Configure Fast Recovery Area

The Fast Recovery Area (FRA) is a unified storage location for all recovery-related files and activities in an Oracle database. All files that are needed to completely recover a database from a media failure are part of the Fast Recovery Area.

View and run the script $HOME/admin/s22_setFRA.sh

☐ Complete

```
SQL> show parameters db_recovery_file_dest

NAME     TYPE VALUE
------------------------------------ ----------- -------------------------
db recovery file dest             string
     /u01/app/oracle/fast_recovery_area
db recovery file dest size        big integer    4800M

SQL> alter system set db_recovery_file_dest_size=9000M scope=both;

System altered.
```

```
SQL>
```

P Enable Database Flashback

With Flashback Database, you can quickly bring your database to an earlier point in time by undoing all the changes that have taken place since that time. This operation is fast because you do not need to restore backups. You can use this feature to undo changes that have resulted in logical data corruptions.

View and run the script $HOME/admin/s23_setFRA.sh

Complete

```
SQL> show parameters DB_FLASHBACK_RETENTION_TARGET

NAME     TYPE VALUE
------------------------------------      -----------    ------------------
db flashback retention target      integer        2880
SQL> alter system set DB_FLASHBACK_RETENTION_TARGET=5000 scope=both;

System altered.


SQL> alter database flashback on;

Database altered.

SQL> select flashback_on from v$database;

FLASHBACK_ON
------------------
YES

SQL> exit
Disconnected from Oracle Database 12c Enterprise Edition Release 12.1.0.1.0
- 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real Application Testing
options
[oracle@prmy01 ~]$
```

S Enable Database Flashback

With Flashback Database, you can quickly bring your database to an earlier point in time by undoing all the changes that have taken place since that time. This operation is fast because you do not need to restore backups. You can use this feature to undo changes that have resulted in logical data corruptions.

View and run the script $HOME/admin/s24_setFRA.sh

Complete

```
SQL> show parameters DB_FLASHBACK_RETENTION_TARGET
```

```
NAME     TYPE VALUE
------------------------------------      -----------    ------------------
db_flashback_retention_target     integer        2880
SQL> alter system set DB_FLASHBACK_RETENTION_TARGET=5000 scope=both;

System altered.


SQL> alter database recover managed standby database cancel;

Database altered.

SQL> alter database flashback on;

Database altered.

SQL> alter database recover managed standby database disconnect from
session;

Database altered.

SQL>
```



Verify Data Guard Broker Configuration

Show the current Data Guard Configuration. It must show a healthy status with DHDB is the primary and DXDB is the standby database.

View and run the script $HOME/admin/s25_setFRA.sh




```
[oracle@prmy01 ~]$ dgmgrl
DGMGRL for Linux: Version 12.1.0.1.0 - 64bit Production

Copyright (c) 2000, 2012, Oracle. All rights reserved.

Welcome to DGMGRL, type "help" for information.
DGMGRL> connect sys/oracle
Connected as SYSDG.
DGMGRL> show configuration

Configuration - DG_CONF

  Protection Mode: MaxPerformance
  Databases:
  DHDB - Primary database
    DXDB - Physical standby database

Fast-Start Failover: DISABLED

Configuration Status:
SUCCESS

DGMGRL>
```

O Configure SQLNET Service

Ensure the observer server is able to connect to the Primary database and Standby database.

View and run the script $HOME/admin/s26_ConfSQLNET.sh

Complete

```
[oracle@dgobserver ~]$ cd $ORACLE_HOME/network/admin
[oracle@dgobserver admin]$ vi tnsnames.ora
```

```
DXDB =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = stdby01.oracle.com)(PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = DXDB.oracle.com)
    )
  )

DHDB =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = prmy01.oracle.com)(PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = DHDB.oracle.com)
    )
  )
```

P Verify Static Instance Registration

This is required by the observer server in order to automatically start the database registered in the listener configuration.

View and run the script $HOME/admin/s26_ConfSQLNET.sh

Complete

```
(SID DESC =
      (GLOBAL_DBNAME = DHDB.oracle.com)
      (ORACLE_HOME = /u01/app/oracle/product/12.1.0/dbhome_1)
      (SID_NAME = DHDB)
)
```

Run the listener STATUS command.

```
[oracle@prmy01 ~]$ lsnrctl

LSNRCTL for Linux: Version 12.1.0.1.0 - Production on 17-NOV-2017 13:12:07
```

```
Copyright (c) 1991, 2013, Oracle.  All rights reserved.

Welcome to LSNRCTL, type "help" for information.

LSNRCTL> status
Connecting to
(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=prmy01.oracle.com)(PORT=1521)))
STATUS of the LISTENER
------------------------
Alias                     LISTENER
Version                   TNSLSNR for Linux: Version 12.1.0.1.0 - Production
Start Date                17-NOV-2017 07:41:25
Uptime                    0 days 5 hr. 30 min. 43 sec
Trace Level               off
Security                  ON: Local OS Authentication
SNMP                      OFF
Listener Parameter File
/u01/app/oracle/product/12.1.0/dbhome 1/network/admin/listener.ora
Listener Log File
/u01/app/oracle/diag/tnslsnr/prmy01/listener/alert/log.xml
Listening Endpoints Summary...
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=prmy01.oracle.com)(PORT=1521)))
  (DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=EXTPROC1521)))

(DESCRIPTION=(ADDRESS=(PROTOCOL=tcps)(HOST=prmy01.oracle.com)(PORT=5500))(Se
curity=(my_wallet_directory=/u01/app/oracle/admin/DHDB/xdb_wallet))(Presenta
tion=HTTP)(Session=RAW))
Services Summary...
Service "DHDB.oracle.com" has 2 instance(s).
  Instance "DHDB", status UNKNOWN, has 1 handler(s) for this service...
  Instance "DHDB", status READY, has 1 handler(s) for this service...
Service "DHDBXDB.oracle.com" has 1 instance(s).
  Instance "DHDB", status READY, has 1 handler(s) for this service...
Service "DHDB_DGB.oracle.com" has 1 instance(s).
  Instance "DHDB", status READY, has 1 handler(s) for this service...
Service "DHDB_DGMGRL.oracle.com" has 1 instance(s).
  Instance "DHDB", status UNKNOWN, has 1 handler(s) for this service...
The command completed successfully
LSNRCTL>
```

Verify Static Instance Registration

This is required by the observer server in order to automatically start the database registered in the listener configuration.

View and run the script $HOME/admin/s27_ConfSQLNET.sh

☐ Complete

```
(SID_DESC =
      (GLOBAL_DBNAME = DXDB.oracle.com)
      (ORACLE_HOME = /u01/app/oracle/product/12.1.0/dbhome_1)
      (SID_NAME = DXDB)
)
```

Run the listener STATUS command.

```
[oracle@stdby01 ~]$ lsnrctl

LSNRCTL for Linux: Version 12.1.0.1.0 - Production on 17-NOV-2017 13:13:49

Copyright (c) 1991, 2013, Oracle.  All rights reserved.

Welcome to LSNRCTL, type "help" for information.

LSNRCTL> status
Connecting to
(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=stdby01.oracle.com)(PORT=1521)))
STATUS of the LISTENER
------------------------
Alias                     LISTENER
Version                   TNSLSNR for Linux: Version 12.1.0.1.0 - Production
Start Date                17-NOV-2017 12:42:12
Uptime                    0 days 0 hr. 31 min. 38 sec
Trace Level               off
Security                  ON: Local OS Authentication
SNMP                      OFF
Listener Parameter File
/u01/app/oracle/product/12.1.0/dbhome 1/network/admin/listener.ora
Listener Log File
/u01/app/oracle/diag/tnslsnr/stdby01/listener/alert/log.xml
Listening Endpoints Summary...
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=stdby01.oracle.com)(PORT=1521)))
  (DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=EXTPROC1521)))

(DESCRIPTION=(ADDRESS=(PROTOCOL=tcps)(HOST=stdby01.oracle.com)(PORT=5500))(S
ecurity=(my wallet directory=/u01/app/oracle/admin/DXDB/xdb wallet))(Present
ation=HTTP)(Session=RAW))
Services Summary...
Service "DHDBXDB.oracle.com" has 1 instance(s).
  Instance "DXDB", status READY, has 1 handler(s) for this service...
Service "DXDB.oracle.com" has 2 instance(s).
  Instance "DXDB", status UNKNOWN, has 1 handler(s) for this service...
  Instance "DXDB", status READY, has 1 handler(s) for this service...
Service "DXDB_DGB.oracle.com" has 1 instance(s).
  Instance "DXDB", status READY, has 1 handler(s) for this service...
Service "DXDB_DGMGRL.oracle.com" has 1 instance(s).
  Instance "DXDB", status UNKNOWN, has 1 handler(s) for this service...
The command completed successfully
LSNRCTL>
```

Display Default FAST-START Failover Settings

The FAST-START failover comes with default parameters settings. These parameters are very generic and needs adjustment to suite the specific site requirements.

View and run the script $HOME/admin/s28_FSF_Defaults.sh

Complete

```
[oracle@dgobserver admin]$ dgmgrl
```

```
DGMGRL for Linux: Version 12.1.0.1.0 - 64bit Production

Copyright (c) 2000, 2012, Oracle. All rights reserved.

Welcome to DGMGRL, type "help" for information.
DGMGRL> connect sys/oracle@DHDB
Connected as SYSDBA.
DGMGRL> show fast_start failover

Fast-Start Failover: DISABLED

  Threshold:          30 seconds
  Target:             (none)
  Observer:           (none)
  Lag Limit:          30 seconds
  Shutdown Primary:   TRUE
  Auto-reinstate:     TRUE
  Observer Reconnect: (none)
  Observer Override:  FALSE

Configurable Failover Conditions
  Health Conditions:
    Corrupted Controlfile          YES
    Corrupted Dictionary           YES
    Inaccessible Logfile            NO
    Stuck Archiver                  NO
    Datafile Offline               YES

  Oracle Error Conditions:
    (none)

DGMGRL>
```

0 Specify FAST-START Target Standby Database

In a Data Guard configuration with multiple standby databases, you must set the FastStartFailoverTarget database property on the primary database and your chosen target standby database before enabling fast-start failover.

View and run the script $HOME/admin/s29_FSF_Target.sh

Complete

```
DGMGRL> edit database 'DHDB'
> set property FastStartFailoverTarget = 'DXDB';
Property "faststartfailovertarget" updated
DGMGRL>


DGMGRL> edit database 'DXDB'
> set property FastStartFailoverTarget = 'DHDB';
Property "faststartfailovertarget" updated
DGMGRL>
```

Set the Protection Mode

You can enable fast-start failover when the configuration is in maximum availability mode or maximum performance mode.

View and run the script $HOME/admin/s30_FSF_mode.sh

Complete

```
DGMGRL> EDIT DATABASE 'DHDB'
> set property LogXptMode='ASYNC';
Property "logxptmode" updated
DGMGRL> EDIT DATABASE 'DXDB'
> set property LogXptMode='ASYNC';
Property "logxptmode" updated
DGMGRL> EDIT CONFIGURATION
> SET PROTECTION MODE AS MaxPerformance;
Succeeded.
DGMGRL>
```

Set the FastStartFailoverThreshold Property

The fast-start failover threshold specifies how long the observer and target standby database should simultaneously wait to hear from the primary database before initiating a fast-start failover. The threshold value is specified as a positive, nonzero number of seconds. The default value for the FastStartFailoverThreshold property is 30 seconds.

View and run the script $HOME/admin/s31_FSF_TH.sh

Complete

```
DGMGRL> EDIT CONFIGURATION
> SET PROPERTY FastStartFailoverThreshold = 45;
Property "faststartfailoverthreshold" updated
DGMGRL>
```

Set Additional Database Properties

Set additional properties to control fast-start failover operation.

The parameter FastStartFailoverLagLimit: Establishes an acceptable length of time for the standby to fall behind the primary database with respect to applied redo.

The parameter FastStartFailoverPmyShutdown: Determines whether the primary database shuts down if redo generation has stalled and the primary database has lost connectivity with the observer and target standby database for a longer time than FastStartFailoverThreshold

View and run the script $HOME/admin/s31_FSF_DBP.sh

Complete

```
DGMGRL> edit configuration set property
> FastStartFailoverLagLimit = 45;
Property "faststartfailoverlaglimit" updated
DGMGRL>

DGMGRL> edit configuration set property
> FastStartFailoverPmyShutdown = false;
Property "faststartfailoverpmyshutdown" updated
DGMGRL>
```

Set Additional Fast-Start Failover Conditions

Use the ENABLE/DISABLE FAST_START FAILOVER broker commands to specify the conditions under which a fast-start failover should occur. The Oracle Database server detects when a specified condition occurs and informs the observer.

View and run the script $HOME/admin/s32_FSF_COND.sh

Complete

```
DGMGRL> enable fast_start failover condition '1578';
Succeeded.
DGMGRL>

DGMGRL> enable fast_start failover condition "Inaccessible Logfile";
Succeeded.
DGMGRL>
```

Display and Verify Current FAST-START Failover Properties

Verify the current FAST-START failover settings are reflected by the commands executed.

View and run the script $HOME/admin/s33_FSF_Verify.sh

Complete

```
DGMGRL> show fast_start failover

Fast-Start Failover: DISABLED

  Threshold:          45 seconds
  Target:             (none)
  Observer:           (none)
  Lag Limit:          45 seconds
  Shutdown Primary:   FALSE
  Auto-reinstate:     TRUE
  Observer Reconnect: (none)
  Observer Override:  FALSE
```

```
Configurable Failover Conditions
  Health Conditions:
    Corrupted Controlfile          YES
    Corrupted Dictionary           YES
    Inaccessible Logfile           YES
    Stuck Archiver                  NO
    Datafile Offline               YES

  Oracle Error Conditions:
    ORA-01578: ORACLE data block corrupted (file # %s, block # %s)

DGMGRL>
```

O Enable the FAST_START Failover

Enable the FAST-START failover using the DGMGRL command ENABLE.

View and run the script $HOME/admin/s34_FSF_Enable.sh

☐ Complete

```
DGMGRL> enable fast_start failover
Enabled.
DGMGRL>
```

O Start the Observer Server

Start the observer Server for FAST-START failover using the command START. The terminal will run on foreground mode. *Do not terminate the terminal.*

View and run the script $HOME/admin/s34_FSF_Start.sh

☐ Complete

```
DGMGRL> start observer
Observer started
```

P Shutdown ABORT Primary Database

Testing FAST-START failover by perform shutdown of the primary database and monitoring the Observer Server terminal behavior.

View and run the script $HOME/admin/s35_DB_SHUT.sh

☐ Complete

```
SQL> conn / as sysdba
Connected.
SQL> shutdown abort;
ORACLE instance shut down.
SQL> exit
```

```
Disconnected from Oracle Database 12c Enterprise Edition Release 12.1.0.1.0
- 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real Application Testing
options
[oracle@prmy01 admin]$
```

O Monitor the Observer Terminal

Monitor the Observer Server currently processing. It's performing automatic failover to the target standby database – DXDB.

Wait for 45-Seconds for the observer to begin working…

Complete

```
DGMGRL> start observer
Observer started

14:13:09.22  Friday, November 17, 2017
Initiating Fast-Start Failover to database "DXDB"...
Performing failover NOW, please wait...
Failover succeeded, new primary is "DXDB"
14:13:11.36  Friday, November 17, 2017
```

P Display Current Data Guard Status

Show the current Data Guard configuration using DGMGRL of the current primary database – DXDB.

View and run the script $HOME/admin/s36_DG_Config.sh

Complete

```
[oracle@prmy01 admin]$ dgmgrl
DGMGRL for Linux: Version 12.1.0.1.0 - 64bit Production

Copyright (c) 2000, 2012, Oracle. All rights reserved.

Welcome to DGMGRL, type "help" for information.
DGMGRL> connect sys/oracle@DXDB
Connected as SYSDBA.
DGMGRL> show configuration

Configuration - DG CONF

  Protection Mode: MaxPerformance
  Databases:
  DXDB - Primary database
    Warning: ORA-16829: fast-start failover configuration is lagging

    DHDB - (*) Physical standby database (disabled)
      ORA-16661: the standby database needs to be reinstated

Fast-Start Failover: ENABLED
```

```
Configuration Status:
WARNING

DGMGRL> exit
[oracle@prmy01 admin]$
```

S Automatic Re-Instatement of DHDB Database

To reinstate the database, you must start the instance and mount the database. Then monitor the Observer Server terminal, it's re-instating the database automatically.

View and run the script $HOME/admin/s37_Reinsatate.sh




```
[oracle@prmy01 admin]$ sqlplus /nolog

SQL*Plus: Release 12.1.0.1.0 Production on Wed Nov 29 11:40:28 2017

Copyright (c) 1982, 2013, Oracle.  All rights reserved.

SQL> connect / as sysdba
Connected to an idle instance.
SQL> startup mount
ORACLE instance started.

Total System Global Area   839282688 bytes
Fixed Size                   2293928 bytes
Variable Size              620760920 bytes
Database Buffers           213909504 bytes
Redo Buffers                 2318336 bytes
Database mounted.
SQL>
```

The observer terminal current display is trying to re-instate the original primary to become the current standby database.

```
DGMGRL> start observer
Observer started

14:13:09.22  Friday, November 17, 2017
Initiating Fast-Start Failover to database "DXDB"...
Performing failover NOW, please wait...
Failover succeeded, new primary is "DXDB"
14:13:11.36  Friday, November 17, 2017

14:20:53.09  Friday, November 17, 2017
Initiating reinstatement for database "DHDB"...
Reinstating database "DHDB", please wait...
Reinstatement of database "DHDB" succeeded
14:21:06.48  Friday, November 17, 2017
```

Show Current Data Guard Status

This should display the primary as 'DXDB' and the standby 'DHDB.

View and run the script $HOME/admin/s38_DG_Status.sh

Complete

```
[oracle@prmy01 admin]$ dgmgrl
DGMGRL for Linux: Version 12.1.0.1.0 - 64bit Production

Copyright (c) 2000, 2012, Oracle. All rights reserved.

Welcome to DGMGRL, type "help" for information.
DGMGRL> connect sys/oracle
Connected as SYSDG
DGMGRL> show configuration

Configuration - DG_CONF

  Protection Mode: MaxPerformance
  Databases:
  DXDB - Primary database
    DHDB - (*) Physical standby database

Fast-Start Failover: ENABLED

Configuration Status:
SUCCESS

DGMGRL>
```

Stop the Observer Server and Disable FAST-START Failover

Stop the Observer Server and disable FAST_START failover in preparation of manual switchover to the original primary database – DHDB.

View and run the script $HOME/admin/s39_DG_Stop.sh

Complete

```
DGMGRL> stop observer;
Done.
DGMGRL> disable fast_start failover;
Disabled.
DGMGRL>
```

Switchover to DHDB

The FAST-START failover is configured and tested to perform automatic failover and automatic re-instatement. Prior to continuing the next workshops, let's go back to

the original data guard configuration, where the primary database is DHDB and the standby database DXDB.

View and run the script $HOME/admin/s40_DG_Switchover.sh

Complete

```
DGMGRL> switchover to 'DHDB';
Performing switchover NOW, please wait...
Operation requires a connection to instance "DHDB" on database "DHDB"
Connecting to instance "DHDB"...
Connected as SYSDBA.
New primary database "DHDB" is opening...
Operation requires startup of instance "DXDB" on database "DXDB"
Starting instance "DXDB"...
ORACLE instance started.
Database mounted.
Database opened.
Switchover succeeded, new primary is "DHDB"
DGMGRL>
```

P Show Current Data Guard Status

This should display the primary as 'DHDB' and the standby 'DXDB.

View and run the script $HOME/admin/s41_DG_Status.sh

Complete

```
DGMGRL> show configuration

Configuration - DG CONF

  Protection Mode: MaxPerformance
  Databases:
  DHDB - Primary database
    DXDB - Physical standby database

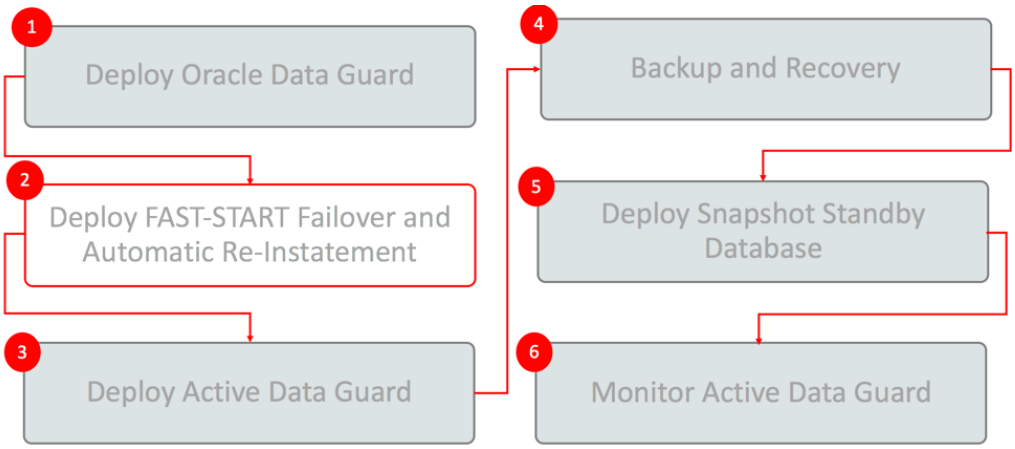
Fast-Start Failover: DISABLED

Configuration Status:
SUCCESS

DGMGRL> exit
[oracle@prmy01 admin]$
```

Congratulation!

Well-done, you have successfully completed Part II: Deploying FAST-START Failover and Re-Instatement workshop.



Part III: Deploy Active Data Guard

This part is simply enabling query offloading on an Oracle data guard. The auto-managed standby database is open on read-only to enable running report and RMAN backup, offloading the primary database from long-running operations.

Tasks

Table 8 lists the tasks associated with Part III, Deploy Oracle Active Data Guard.

Task#	Description
1	Start the Workload Simulator
2	Show Current Standby Database Prosperities
3	Disable Real-Time Query
4	Run Query Test
5	Enable Real-Time Query
6	Run Query Test
7	Disable Real-Time Query
8	Stop the Workload Simulator

Table 8: Deploy Active Data Guard tasks

Architecture

The Active Data Guard (ADG) option enables the auto-managed standby database to be opened in read-only mode. There are two ways to enable ADG:

- SQL command-line
- DGMGRL command-line

The primary database perform redo-transport, the standby database perform redo-apply as shown in Figure 10.

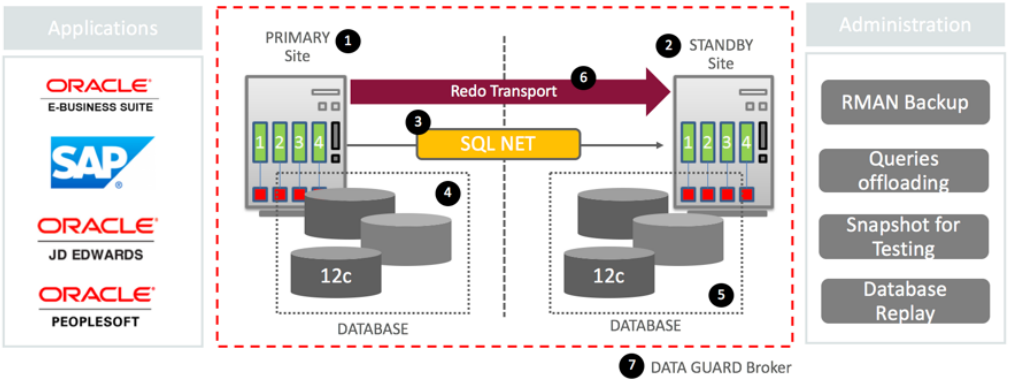


Figure 10: Active Data Guard Architecture

Oracle Active Data Guard enables the following features:

- Database Block Change Tracking feature for RMAN incremental backup
- Real-Time Query for query offloading

P Start the Workload Simulator

Begin by starting the workload simulator. The workload is executed on the primary database and synchronized on the standby database.

Complete

```
[oracle@prmy01 admin]$ sqlplus /nolog

SQL*Plus: Release 12.1.0.1.0 Production on Wed Nov 29 12:00:15 2017

Copyright (c) 1982, 2013, Oracle.  All rights reserved.

SQL> conn osm_repapi/oracle
Connected.
SQL> exec call_trans_load;
```

Leave the terminal running. Do not terminate it.

S Show Current Standby Database Prosperities

Display the database properties, by default the real-time apply query is enabled – ON. This indicates the database is mounted and available for read-only and specific maintenance tasks such as RMAN commands.

.

Complete

```
[oracle@stdby01 admin]$ dgmgrl
DGMGRL for Linux: Version 12.1.0.1.0 - 64bit Production

Copyright (c) 2000, 2012, Oracle. All rights reserved.

Welcome to DGMGRL, type "help" for information.
DGMGRL> connect sys/oracle
Connected as SYSDG.
DGMGRL> show database 'DXDB';

Database - DXDB

  Role:              PHYSICAL STANDBY
  Intended State:    APPLY-ON
  Transport Lag:     0 seconds (computed 1 second ago)
  Apply Lag:         0 seconds (computed 1 second ago)
  Apply Rate:        0 Byte/s
  Real Time Query:   ON
  Instance(s):
    DXDB

Database Status:
SUCCESS

DGMGRL>
```

S Disable Real-Time Query

Disable Real-Time Query setting on the standby database.

Complete

```
DGMGRL> edit database 'DXDB' set state='apply-off';
Succeeded.
DGMGRL> show database 'DXDB';

Database - DXDB

  Role:               PHYSICAL STANDBY
  Intended State:     APPLY-OFF
  Transport Lag:      0 seconds (computed 0 seconds ago)
  Apply Lag:          12 seconds (computed 0 seconds ago)
  Apply Rate:         (unknown)
  Real Time Query:    OFF
  Instance(s):
    DXDB

Database Status:
SUCCESS

DGMGRL> exit
[oracle@stdby01 admin]$
```

S Run Query Test

Query the CUSTOMERS table for row count. Row count is not changing; this is because Real-Time Query Active Data Guard (ADG) is disabled. Keep re-executing the query several times to verify ADG is disabled.

Complete

```
[oracle@stdby01 admin]$ sqlplus /nolog

SQL*Plus: Release 12.1.0.1.0 Production on Wed Nov 29 12:05:52 2017

Copyright (c) 1982, 2013, Oracle.  All rights reserved.

SQL> conn osm_repapi/oracle
Connected.
SQL> show user
USER is "OSM_REPAPI"
SQL> select count(*) from customers;

  COUNT(*)
----------
   1911547

SQL> /

  COUNT(*)
----------
   1911547
```

```
SQL> exit
Disconnected from Oracle Database 12c Enterprise Edition Release 12.1.0.1.0
- 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real Application Testing
options
[oracle@stdby01 admin]$
```

S Enable Real-Time Query

Using DGMGRL, enable Real-Time Query. This means the Redo-Transport is applied to the standby and it's available for read-only operations.

Complete

```
[oracle@stdby01 admin]$ dgmgrl
DGMGRL for Linux: Version 12.1.0.1.0 - 64bit Production

Copyright (c) 2000, 2012, Oracle. All rights reserved.

Welcome to DGMGRL, type "help" for information.
DGMGRL> connect sys/oracle
Connected as SYSDG.
DGMGRL> edit database 'DXDB' set state='apply-on';
Succeeded.
DGMGRL> show database 'DXDB';

Database - DXDB

  Role:               PHYSICAL STANDBY
  Intended State:     APPLY-ON
  Transport Lag:      0 seconds (computed 1 second ago)
  Apply Lag:          0 seconds (computed 1 second ago)
  Apply Rate:         (unknown)
  Real Time Query:    ON
  Instance(s):
    DXDB

Database Status:
SUCCESS

DGMGRL> exit
[oracle@stdby01 admin]$
```

S Run Query Test

Query the CUSTOMERS table for row count. Row count keeps increasing. The Redo-Transport is applied and the database is available for read-only operations.

Complete

```
[oracle@stdby01 admin]$ sqlplus /nolog

SQL*Plus: Release 12.1.0.1.0 Production on Wed Nov 29 12:09:41 2017
```

```
Copyright (c) 1982, 2013, Oracle.  All rights reserved.

SQL>conn osm_repapi/oracle
Connected.
SQL> select count(*) from customers;

  COUNT(*)
----------
   1912263

SQL> /

  COUNT(*)
----------
   1912267

SQL> exit
Disconnected from Oracle Database 12c Enterprise Edition Release 12.1.0.1.0
- 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real Application Testing
options
[oracle@stdby01 admin]$
```

S Disable Real-Time Query

Disable Real-Time Query option. The role of the database remains PHYSICAL STANDBY and the Real-Time Query is OFF.

Complete

```
[oracle@stdby01 admin]$ dgmgrl
DGMGRL for Linux: Version 12.1.0.1.0 - 64bit Production

Copyright (c) 2000, 2012, Oracle. All rights reserved.

Welcome to DGMGRL, type "help" for information.
DGMGRL> connect sys/oracle
Connected as SYSDG.
DGMGRL> edit database 'DXDB' set state='apply-off';
Succeeded.
DGMGRL> show database 'DXDB';

Database - DXDB

  Role:               PHYSICAL STANDBY
  Intended State:     APPLY-OFF
  Transport Lag:      0 seconds (computed 1 second ago)
  Apply Lag:          8 seconds (computed 1 second ago)
  Apply Rate:         (unknown)
  Real Time Query:    OFF
  Instance(s):
    DXDB

Database Status:
SUCCESS
```

```
DGMGRL> exit
[oracle@stdby01 admin]$
```

S Stop the Workload Simulator

Stop the workload simulator by performing the following steps:

- Determine the workload simulator session's SID and SERIAL#
- Issue the KILL SESSION SQL command
- Supply the correct SID and SERIAL# values

Complete

```
[oracle@stdby01 admin]$ sqlplus /nolog

SQL*Plus: Release 12.1.0.1.0 Production on Wed Nov 29 12:13:53 2017

Copyright (c) 1982, 2013, Oracle.  All rights reserved.

SQL> conn system/oracle_4U@DHDB
Connected.
SQL> select sid, serial# from v$session where username='OSM_REPAPI';

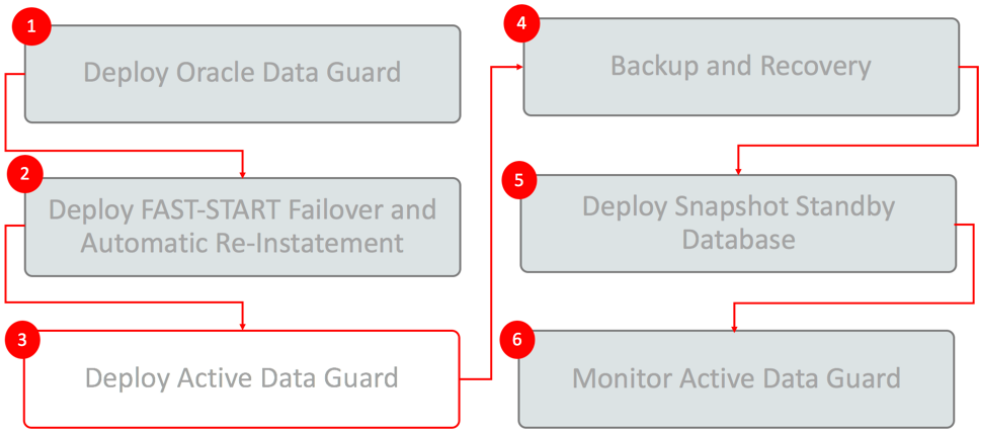
       SID    SERIAL#
---------- ----------
     61        879

SQL> alter system kill session '61, 879' immediate;
alter system kill session '61, 879' immediate
*
ERROR at line 1:
ORA-00031: session marked for kill


SQL>
```

Congratulation!

You've successfully finished Part III: Deploy Oracle Active Data Guard.



Part IV: Implementing Backup and Recovery

This workshop demonstrates how to use Oracle Recovery Manager for backup and restore operations for environment running Oracle Data Guard. Using Oracle Recovery Manager (RMAN) backup/restore scripts and RMAN policies empowers Oracle Best Practices for Oracle Data Guard running Oracle Database 12c.

Tasks

Table 9 lists the tasks associated with Part IV, Backup and Recovery.

Task#	Description
1	Creating the RCAT Database
2	Create the Recovery Catalog
3	Create the Backup Script
4	Create the Database Block Change Tracking Feature
5	Start the Workload Simulator
6	Enable Redo Apply
7	Perform Database Backup
8	Delete obsolete Backup

Table 9: Backup and Recovery tasks

Architecture

The aim of the workshop is to offload backup from the primary database and move it to the standby database. Figure 11 shows the components use to develop the Recovery Manager (RMAN) for performing backup on the physical standby database.

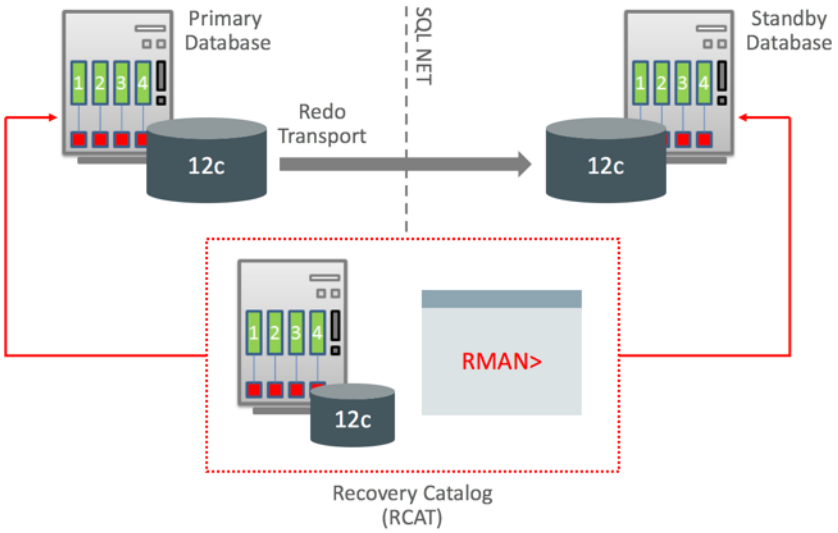


Figure 11: Backup and Recovery using RMAN

In order for RMAN to perform optimized backup, the database block change track feature is enabled. The Recovery Manager catalog is created on the observer server, this allows us to use stored scripts to perform backup.

RMAN catalog database (RCAT) is created manually, which require performing the following tasks:

- Prepare the parameter file initRCAT.ora
- Create the Password file
- Create the instance handler (RCAT)
- Run the command CREATE DATABASE
- Run the Data Dictionary SQL scripts
- Create SPFILE
- Create Temporary, UNDO and USERS tablespaces

Creating the RCAT Database

This database will be dedicated for the Recovery Manager (RMAN) catalog. To create the database from SQL*PLUS command line, you need to prepare the following:

- Prepare the parameter file initRCAT.ora
- Create the Password file
- Create the instance handler (RCAT)
- Run the command CREATE DATABASE
- Run the Data Dictionary SQL scripts
- Create SPFILE
- Create Temporary, UNDO and USERS tablespaces

Prepare initRCAT.ora parameter file. Complete

```
[oracle@dgobserver ~]$ cd $ORACLE_HOME/dbs
[oracle@dgobserver dbs]$ vi initRCAT.ora
```

```
*.audit_file_dest='/u01/app/oracle/admin/RCAT/adump'
*.audit_trail='db'
*.compatible='12.1.0.0.0'
*.db_block_size=8192
*.db_create_file_dest='/u01/app/oracle/oradata'
*.db_create_online_log_dest_1='/u01/app/oracle/oradata'
*.db_domain='oracle.com'
*.db_name='RCAT'
*.db_recovery_file_dest='/u01/app/oracle/fast_recovery_area'
*.db_recovery_file_dest_size=4294967296
*.diagnostic_dest='/u01/app/oracle'
*.dispatchers='(PROTOCOL=TCP) (SERVICE=RCATXDB)'
*.memory_target=801m
*.open_cursors=3000
*.processes=300
*.recyclebin='OFF'
*.remote_login_passwordfile='EXCLUSIVE'
```

Create the password file. Use the Oracle utility orapwd. Complete

```
[oracle@dgobserver dbs]$ orapwd file=$ORACLE_HOME/dbs/orapwRCAT
password=oracle ignorecase=y force=y entries=30
[oracle@dgobserver dbs]$ ls -al orapwRCAT
-rw-r----- 1 oracle oinstall 17920 Nov 17 19:29 orapwRCAT
[oracle@dgobserver dbs]$
```

Create sub-directories structure to support database auxiliary settings. Complete

```
[oracle@dgobserver ~]$ mkdir -p /u01/app/oracle/oradata/RCAT
[oracle@dgobserver ~]$ mkdir -p /u01/app/oracle/oradata/RCAT/arch
[oracle@dgobserver ~]$ mkdir -p /u01/app/oracle/admin/RCAT/adump
```

```
[oracle@dgobserver ~]$ mkdir -p /u01/app/oracle/admin/RCAT/bdump
[oracle@dgobserver ~]$ mkdir -p /u01/app/oracle/admin/RCAT/cdump
[oracle@dgobserver ~]$ mkdir -p /u01/app/oracle/admin/RCAT/dpdump
[oracle@dgobserver ~]$ mkdir -p /u01/app/oracle/admin/RCAT/pfile
[oracle@dgobserver ~]$ mkdir -p /u01/app/oracle/admin/RCAT/scripts
[oracle@dgobserver ~]$ mkdir -p /u01/app/oracle/admin/RCAT/udump
[oracle@dgobserver ~]$ mkdir -p /u01/app/oracle/fast_recovery_area
```

Set the instance Linux environment variables.




```
[oracle@dgobserver ~]$ export ORACLE_SID=RCAT
[oracle@dgobserver ~]$ export
ORACLE_HOME=/u01/app/oracle/product/12.1.0/dbhome_1
```

Start the instance in NOMOUNT state




```
[oracle@dgobserver ~]$ . oraenv
ORACLE_SID = [RCAT] ?
The Oracle base remains unchanged with value /u01/app/oracle
[oracle@dgobserver ~]$ sqlplus /nolog

SQL*Plus: Release 12.1.0.1.0 Production on Fri Nov 17 20:06:14 2017

Copyright (c) 1982, 2013, Oracle.  All rights reserved.

SQL> conn / as sysdba
Connected to an idle instance.
SQL> startup nomount pfile='$ORACLE_HOME/dbs/initRCAT.ora'
ORACLE instance started.

Total System Global Area          839282688 bytes
Fixed Size                          2293928 bytes
Variable Size                     591400792 bytes
Database Buffers                  243269632 bytes
Redo Buffers                        2318336 bytes
SQL>
SQL> CREATE DATABASE;

Database created.

SQL>
```

Create the data dictionary metadata and API.




```
SQL> @$ORACLE_HOME/rdbms/admin/catalog.sql
SQL> @$ORACLE_HOME/rdbms/admin/catproc.sql
```

Create SPFILE from PFILE




```
SQL> create spfile from pfile;

File created.
```

```
SQL>
```

Update the location of the Controlfile.

Complete

```
SQL> select value from v$parameter where name='control_files';

VALUE
--------------------------------------------------------------------------------
/u01/app/oracle/oradata/RCAT/controlfile/o1 mf f1x06vo1 .ctl

1 row selected.


SQL> shutdown immediate
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> startup nomount
ORACLE instance started.

Total System Global Area  839282688 bytes
Fixed Size                  2293928 bytes
Variable Size             591400792 bytes
Database Buffers          243269632 bytes
Redo Buffers                2318336 bytes
SQL> alter system set
control_files='/u01/app/oracle/oradata/RCAT/controlfile/o1_mf_f0y76q7v_.ctl'
scope=spfile;

System altered.

SQL>
SQL> startup force
ORACLE instance started.

Total System Global Area  839282688 bytes
Fixed Size                  2293928 bytes
Variable Size             591400792 bytes
Database Buffers          243269632 bytes
Redo Buffers                2318336 bytes
Database mounted.
Database opened.
SQL>
```

Create the Temporary tablespace and assign it as the database-wide default Temporary tablespace.

Complete

```
SQL> create tablespace temp
  2  temporary datafile '/u01/app/oracle/oradata/RCAT/datafile/temp01.dbf'
size 100M;

Tablespace created.

SQL> alter database default temporary tablespace temp;
```

```
Database altered.

SQL>
```

Configure automatic undo management.

Complete

```
SQL> show parameters undo

NAME                                 TYPE        VALUE
------------------------------------ ----------- ------------------------------
temp undo enabled                    boolean     FALSE
undo_management                      string      MANUAL
undo_retention                       integer     900
undo tablespace                      string

SQL> create undo tablespace undotbs01
  2  datafile '/u01/app/oracle/oradata/RCAT/datafile/undotbs01.dbf' size
100M;

Tablespace created.

SQL> alter system set undo_management=auto scope=spfile;

System altered.

SQL> alter system set undo_tablespace='undotbs01' scope=spfile;

System altered.

SQL> startup force
ORACLE instance started.

Total System Global Area  839282688 bytes
Fixed Size                  2293928 bytes
Variable Size             599789400 bytes
Database Buffers          234881024 bytes
Redo Buffers                2318336 bytes
Database mounted.
Database opened.
SQL>
```

Create the USERS tablespace. Assign it as the database-wide default tablespace.

Complete

```
SQL> create tablespace users
  2  datafile '/u01/app/oracle/oradata/RCAT/datafile/users01.dbf' size 100M
  3  autoextend on;

Tablespace created.

SQL> alter database default tablespace users;

Database altered.
```

```
SQL>
```



Create the Recovery Catalog

In order to support several Oracle Data Guard functionalities, an RMAN recovery catalog is required. The following steps are executed on the database:

- Create the default recovery catalog tablespace
- Create the recovery catalog schema
- Assign database privileges
- Start RMAN and connect to CATALOG and TARGET databases
- Create the catalog
- Register the Database (Primary)
- Perform full and incremented backups
- Enable block tracking feature
- Create stored backup script
- Perform incremental backup

Create recovery catalog tablespace – RCAT




```
SQL> create tablespace rcat
  2  datafile '/u01/app/oracle/oradata/RCAT/datafile/rcat01.dbf' size 100M
  3  autoextend on;

Tablespace created.

SQL>
```

Create the recovery catalog owner - RCATMGR




```
SQL> create user rcatmgr
  2  identified by oracle_4U
  3  default tablespace rcat
  4  quota unlimited on rcat;

User created.

SQL>
```

Grant data dictionary privileges




```
SQL> grant recovery_catalog_owner to rcatmgr;

Grant succeeded.

SQL> exit
```

```
Disconnected from Oracle Database 12c Enterprise Edition Release 12.1.0.1.0
- 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real Application Testing
options
[oracle@dgobserver dbs]$
```

Start RMAN, connect to the primary database

Complete

```
[oracle@dgobserver ~]$ rman

Recovery Manager: Release 12.1.0.1.0 - Production on Fri Nov 17 23:12:08
2017

Copyright (c) 1982, 2013, Oracle and/or its affiliates.  All rights
reserved.

RMAN> connect catalog rcatmgr/oracle_4U

connected to recovery catalog database

RMAN> connect target sys/oracle@DHDB

connected to target database: DHDB (DBID=2763595033)

RMAN>
```

Create the recovery catalog

Complete

```
RMAN> create catalog;

recovery catalog created

RMAN>
```

Register the database

Complete

```
RMAN> register database;

database registered in recovery catalog
starting full resync of recovery catalog
full resync complete

RMAN> exit


Recovery Manager complete
```

Show current RMAN settings. Connect to the standby database.
‘

Complete

```
[oracle@dgobserver ~]$ rman

Recovery Manager: Release 12.1.0.1.0 - Production on Fri Nov 17 23:13:25
2017
```

```
Copyright (c) 1982, 2013, Oracle and/or its affiliates.  All rights
reserved.

RMAN> connect catalog rcatmgr/oracle_4U

connected to recovery catalog database

RMAN> connect target sys/oracle@DXDB

connected to target database: DHDB (DBID=2763595033)

RMAN> show all;

RMAN configuration parameters for database with db_unique_name DXDB are:
CONFIGURE RETENTION POLICY TO REDUNDANCY 1; # default
CONFIGURE BACKUP OPTIMIZATION OFF; # default
CONFIGURE DEFAULT DEVICE TYPE TO DISK; # default
CONFIGURE CONTROLFILE AUTOBACKUP OFF; # default
CONFIGURE CONTROLFILE AUTOBACKUP FORMAT FOR DEVICE TYPE DISK TO '%F'; #
default
CONFIGURE DEVICE TYPE DISK PARALLELISM 1 BACKUP TYPE TO BACKUPSET; # default
CONFIGURE DATAFILE BACKUP COPIES FOR DEVICE TYPE DISK TO 1; # default
CONFIGURE ARCHIVELOG BACKUP COPIES FOR DEVICE TYPE DISK TO 1; # default
CONFIGURE MAXSETSIZE TO UNLIMITED; # default
CONFIGURE ENCRYPTION FOR DATABASE OFF; # default
CONFIGURE ENCRYPTION ALGORITHM 'AES128'; # default
CONFIGURE COMPRESSION ALGORITHM 'BASIC' AS OF RELEASE 'DEFAULT' OPTIMIZE FOR
LOAD TRUE ; # default
CONFIGURE RMAN OUTPUT TO KEEP FOR 7 DAYS; # default
CONFIGURE ARCHIVELOG DELETION POLICY TO NONE; # default
CONFIGURE SNAPSHOT CONTROLFILE NAME TO
'/u01/app/oracle/product/12.1.0/dbhome_1/dbs/snapcf_DXDB.f'; # default

RMAN>
```

Configure auto-backup for the Controlfile and SPFILE

Complete

```
RMAN> CONFIGURE CONTROLFILE AUTOBACKUP ON;

new RMAN configuration parameters:
CONFIGURE CONTROLFILE AUTOBACKUP ON;
new RMAN configuration parameters are successfully stored

RMAN>
```

Perform full database backup

Complete

```
RMAN> backup database plus archivelog;


Starting backup at 17-NOV-17
current log archived at primary database
allocated channel: ORA DISK 1
channel ORA_DISK_1: SID=24 device type=DISK
channel ORA_DISK_1: starting archived log backup set
```

```
channel ORA DISK 1: specifying archived log(s) in backup set
input archived log thread=1 sequence=19 RECID=1 STAMP=959095968
input archived log thread=1 sequence=20 RECID=2 STAMP=959095969
input archived log thread=1 sequence=21 RECID=3 STAMP=959096106
input archived log thread=1 sequence=22 RECID=4 STAMP=959096984
input archived log thread=1 sequence=23 RECID=5 STAMP=959097036
input archived log thread=1 sequence=24 RECID=6 STAMP=959097038
input archived log thread=1 sequence=25 RECID=7 STAMP=959097047
input archived log thread=1 sequence=26 RECID=8 STAMP=959097048
input archived log thread=1 sequence=27 RECID=11 STAMP=959097213
input archived log thread=1 sequence=28 RECID=13 STAMP=959097228
input archived log thread=1 sequence=29 RECID=16 STAMP=959097253
input archived log thread=1 sequence=30 RECID=17 STAMP=959097257
input archived log thread=1 sequence=31 RECID=18 STAMP=959097257
input archived log thread=1 sequence=32 RECID=19 STAMP=959097272
input archived log thread=1 sequence=33 RECID=20 STAMP=959097334
input archived log thread=1 sequence=34 RECID=21 STAMP=959097341
input archived log thread=1 sequence=35 RECID=22 STAMP=959097342
input archived log thread=1 sequence=36 RECID=24 STAMP=959097574
input archived log thread=1 sequence=37 RECID=27 STAMP=959097592
input archived log thread=1 sequence=38 RECID=29 STAMP=959097984
input archived log thread=1 sequence=39 RECID=32 STAMP=959098209
input archived log thread=1 sequence=40 RECID=34 STAMP=959098210
input archived log thread=1 sequence=41 RECID=33 STAMP=959098210
input archived log thread=1 sequence=42 RECID=35 STAMP=959098213
input archived log thread=1 sequence=43 RECID=36 STAMP=959098237
input archived log thread=1 sequence=44 RECID=37 STAMP=959099333
input archived log thread=1 sequence=45 RECID=38 STAMP=959099335
input archived log thread=1 sequence=46 RECID=39 STAMP=959099344
input archived log thread=1 sequence=47 RECID=40 STAMP=959099345
input archived log thread=1 sequence=48 RECID=43 STAMP=959099368
input archived log thread=1 sequence=49 RECID=46 STAMP=959099412
input archived log thread=1 sequence=50 RECID=47 STAMP=959099416
input archived log thread=1 sequence=51 RECID=48 STAMP=959099416
input archived log thread=1 sequence=52 RECID=49 STAMP=959099428
input archived log thread=1 sequence=53 RECID=53 STAMP=960295380
input archived log thread=1 sequence=54 RECID=51 STAMP=960295380
input archived log thread=1 sequence=55 RECID=52 STAMP=960295380
input archived log thread=1 sequence=56 RECID=50 STAMP=960295377
input archived log thread=1 sequence=57 RECID=54 STAMP=960296237
input archived log thread=1 sequence=58 RECID=55 STAMP=960296239
input archived log thread=1 sequence=59 RECID=56 STAMP=960296248
input archived log thread=1 sequence=60 RECID=57 STAMP=960296249
input archived log thread=1 sequence=61 RECID=60 STAMP=960296273
input archived log thread=1 sequence=62 RECID=63 STAMP=960296498
input archived log thread=1 sequence=63 RECID=65 STAMP=960296503
input archived log thread=1 sequence=64 RECID=64 STAMP=960296502
input archived log thread=1 sequence=65 RECID=66 STAMP=960296516
input archived log thread=1 sequence=66 RECID=67 STAMP=960298735
input archived log thread=1 sequence=67 RECID=68 STAMP=960300820
channel ORA DISK 1: starting piece 1 at 17-NOV-17
channel ORA_DISK_1: finished piece 1 at 17-NOV-17
piece
handle=/u01/app/oracle/fast_recovery_area/DXDB/backupset/2017_11_17/o1_mf_an
nnn_TAG20171117T231529_f0yjv18r_.bkp tag=TAG20171117T231529 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:01
channel ORA DISK 1: starting archived log backup set
channel ORA_DISK_1: specifying archived log(s) in backup set
input archived log thread=1 sequence=1 RECID=70 STAMP=960301258
```

```
input archived log thread=1 sequence=2 RECID=72 STAMP=960301265
input archived log thread=1 sequence=3 RECID=74 STAMP=960302079
input archived log thread=1 sequence=4 RECID=77 STAMP=960302098
input archived log thread=1 sequence=5 RECID=79 STAMP=960302102
input archived log thread=1 sequence=6 RECID=78 STAMP=960302101
input archived log thread=1 sequence=7 RECID=80 STAMP=960302115
input archived log thread=1 sequence=8 RECID=82 STAMP=960319872
input archived log thread=1 sequence=9 RECID=81 STAMP=960319871
input archived log thread=1 sequence=10 RECID=83 STAMP=960320784
input archived log thread=1 sequence=11 RECID=84 STAMP=960320853
input archived log thread=1 sequence=12 RECID=85 STAMP=960333325
channel ORA_DISK_1: starting piece 1 at 17-NOV-17
channel ORA DISK 1: finished piece 1 at 17-NOV-17
piece
handle=/u01/app/oracle/fast_recovery_area/DXDB/backupset/2017_11_17/o1_mf_an
nnn_TAG20171117T231529_f0yjv2cb_.bkp tag=TAG20171117T231529 comment=NONE
channel ORA DISK 1: backup set complete, elapsed time: 00:00:01
Finished backup at 17-NOV-17

Starting backup at 17-NOV-17
using channel ORA_DISK_1
channel ORA_DISK_1: starting full datafile backup set
channel ORA DISK 1: specifying datafile(s) in backup set
input datafile file number=00003
name=/u01/app/oracle/oradata/DXDB/datafile/o1_mf_sysaux_dzpdn84c_.dbf
input datafile file number=00001
name=/u01/app/oracle/oradata/DXDB/datafile/o1_mf_system_dzpdocm7_.dbf
input datafile file number=00006
name=/u01/app/oracle/oradata/DXDB/datafile/o1_mf_users_dzpdps52_.dbf
input datafile file number=00004
name=/u01/app/oracle/oradata/DXDB/datafile/o1_mf_undotbs1_dzpdptgh_.dbf
channel ORA_DISK_1: starting piece 1 at 17-NOV-17
channel ORA_DISK_1: finished piece 1 at 17-NOV-17
piece
handle=/u01/app/oracle/fast_recovery_area/DXDB/backupset/2017_11_17/o1_mf_nn
ndf TAG20171117T231531 f0yjv4ss .bkp tag=TAG20171117T231531 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:15
Finished backup at 17-NOV-17

Starting backup at 17-NOV-17
current log archived at primary database
using channel ORA DISK 1
channel ORA_DISK_1: starting archived log backup set
channel ORA_DISK_1: specifying archived log(s) in backup set
input archived log thread=1 sequence=13 RECID=86 STAMP=960333348
channel ORA DISK 1: starting piece 1 at 17-NOV-17
channel ORA DISK 1: finished piece 1 at 17-NOV-17
piece
handle=/u01/app/oracle/fast_recovery_area/DXDB/backupset/2017_11_17/o1_mf_an
nnn_TAG20171117T231551_f0yjvqdr_.bkp tag=TAG20171117T231551 comment=NONE
channel ORA DISK 1: backup set complete, elapsed time: 00:00:01
Finished backup at 17-NOV-17

Starting Control File and SPFILE Autobackup at 17-NOV-17
piece
handle=/u01/app/oracle/fast_recovery_area/DXDB/autobackup/2017_11_17/o1_mf_s
 960316720 f0yjvrrd .bkp comment=NONE
Finished Control File and SPFILE Autobackup at 17-NOV-17
```

```
RMAN>
```

Perform incremental Database backup. Use incremental level 1.

Complete

```
RMAN> backup incremental level 1 database;

Starting backup at 17-NOV-17
using channel ORA_DISK_1
no parent backup or copy of datafile 3 found
no parent backup or copy of datafile 1 found
no parent backup or copy of datafile 6 found
no parent backup or copy of datafile 4 found
channel ORA_DISK_1: starting incremental level 0 datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
input datafile file number=00003
name=/u01/app/oracle/oradata/DXDB/datafile/o1 mf sysaux dzpdn84c .dbf
input datafile file number=00001
name=/u01/app/oracle/oradata/DXDB/datafile/o1 mf system dzpdocm7 .dbf
input datafile file number=00006
name=/u01/app/oracle/oradata/DXDB/datafile/o1_mf_users_dzpdps52_.dbf
input datafile file number=00004
name=/u01/app/oracle/oradata/DXDB/datafile/o1_mf_undotbs1_dzpdptgh_.dbf
channel ORA_DISK_1: starting piece 1 at 17-NOV-17
channel ORA DISK 1: finished piece 1 at 17-NOV-17
piece
handle=/u01/app/oracle/fast_recovery_area/DXDB/backupset/2017_11_17/o1_mf_nn
nd0_TAG20171117T232243_f0yk8n3l_.bkp tag=TAG20171117T232243 comment=NONE
channel ORA DISK 1: backup set complete, elapsed time: 00:00:15
Finished backup at 17-NOV-17

Starting Control File and SPFILE Autobackup at 17-NOV-17
piece
handle=/u01/app/oracle/fast_recovery_area/DXDB/autobackup/2017_11_17/o1_mf_s
_960316720_f0yk93d4_.bkp comment=NONE
Finished Control File and SPFILE Autobackup at 17-NOV-17

RMAN>
```

9 Create the Backup Script

Stored script feature requires a recovery catalog. Execute the RMAN command CREATE SCRIPT. The script is stored in the recovery catalog and can be executed from any Oracle client configured to connect to the recovery catalog and target database.

Complete

```
RMAN> create script s1_bck {
2> backup incremental level 1 database;
3> }

created script s1_bck

RMAN>
```

O Create the Database Block Change Tracking Feature

By enabling the database block change tracking, it enables optimized backup operation. The backup uses an external lookup file for determining the database changes at the physical block level; achieving smaller backup set pieces at backup speed proportional to the database changes since the last incremental backup.

☐ Complete

```
RMAN> alter database enable block change tracking;

Statement processed

RMAN>
```

P Start the Workload Simulator

To avoid data file skip due to optimization settings, the workload simulator is started on the primary database and synchronized on the standby database.

☐ Complete

```
[oracle@prmy01 trace]$ sqlplus /nolog

SQL*Plus: Release 12.1.0.1.0 Production on Sat Nov 18 09:49:57 2017

Copyright (c) 1982, 2013, Oracle.  All rights reserved.

SQL> connect osm_repapi/oracle
Connected.
SQL> exec call_trans_load;
```

Leave the session running, do not interrupt it.

P Enable Redo Apply

Enable apply on the standby database. This allows the sync with the primary database.

☐ Complete

```
[oracle@prmy01 ~]$ dgmgrl
DGMGRL for Linux: Version 12.1.0.1.0 - 64bit Production

Copyright (c) 2000, 2012, Oracle. All rights reserved.

Welcome to DGMGRL, type "help" for information.
DGMGRL> connect sys/oracle
Connected as SYSDG.
DGMGRL> edit database 'DXDB' set state='apply-on';
Succeeded.
DGMGRL> show database 'DXDB';
```

```
Database - DXDB

  Role:              PHYSICAL STANDBY
  Intended State:    APPLY-ON
  Transport Lag:     0 seconds (computed 0 seconds ago)
  Apply Lag:         0 seconds (computed 0 seconds ago)
  Apply Rate:        0 Byte/s
  Real Time Query:   ON
  Instance(s):
    DXDB

Database Status:
SUCCESS

DGMGRL>
```

9 Perform Database Backup

Perform incremental backup by executing the stored backup script. You must be connected to the recovery catalog database and the target database – DXDB. The backup should finish quickly as its incremental backup with database change block tracking feature enabled.

☐ Complete

```
RMAN> run {execute script s1_bck;}

executing script: s1_bck

Starting backup at 17-NOV-17
using channel ORA_DISK_1
channel ORA_DISK_1: starting incremental level 1 datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
input datafile file number=00003
name=/u01/app/oracle/oradata/DXDB/datafile/o1_mf_sysaux_dzpdn84c_.dbf
input datafile file number=00001
name=/u01/app/oracle/oradata/DXDB/datafile/o1_mf_system_dzpdocm7_.dbf
input datafile file number=00006
name=/u01/app/oracle/oradata/DXDB/datafile/o1_mf_users_dzpdps52_.dbf
input datafile file number=00004
name=/u01/app/oracle/oradata/DXDB/datafile/o1_mf_undotbs1_dzpdptgh_.dbf
channel ORA_DISK_1: starting piece 1 at 17-NOV-17
channel ORA_DISK_1: finished piece 1 at 17-NOV-17
piece
handle=/u01/app/oracle/fast recovery area/DXDB/backupset/2017 11 17/o1 mf nn
nd1_TAG20171117T233908_f0yl7z19_.bkp tag=TAG20171117T233908 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:07
Finished backup at 17-NOV-17

Starting Control File and SPFILE Autobackup at 17-NOV-17
piece
handle=/u01/app/oracle/fast_recovery_area/DXDB/autobackup/2017_11_17/o1_mf_s
_960371639_f0yl82sw_.bkp comment=NONE
Finished Control File and SPFILE Autobackup at 17-NOV-17
```

```
RMAN>
```

Delete obsolete Backup

Deleting obsolete backup should be automated. It should occur after successfully completing the backup.

DO NOT delete backup files manually; they should be always deleted using RMAN. Otherwise the recovery catalog and data dictionary are out of sync with physical backup files.

Complete

```
RMAN> delete noprompt obsolete;

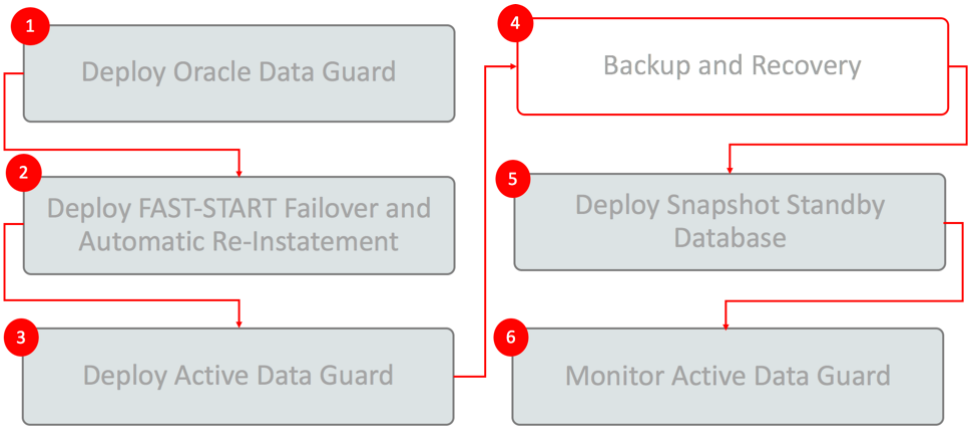
RMAN retention policy will be applied to the command
RMAN retention policy is set to redundancy 1
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=24 device type=DISK
Deleting the following obsolete backups and copies:
Type                 Key    Completion Time    Filename/Handle
-------------------- ------ ------------------ --------------------
Archive Log          503    03-NOV-17
/u01/app/oracle/fast_recovery_area/DXDB/archivelog/2017_11_03/o1_mf_1_19_dzr
rhjny_.arc
Archive Log          504    03-NOV-17
/u01/app/oracle/fast_recovery_area/DXDB/archivelog/2017_11_03/o1_mf_1_20_dzr
rhk3v .arc
Archive Log          505    03-NOV-17
/u01/app/oracle/fast recovery area/DXDB/archivelog/2017 11 03/o1 mf 1 21 dzr
rmt4x_.arc
Archive Log          506    03-NOV-17
/u01/app/oracle/fast_recovery_area/DXDB/archivelog/2017_11_03/o1_mf_1_22_dzr
sh8n7 .arc
Archive Log          507    03-NOV-17
/u01/app/oracle/fast recovery area/DXDB/archivelog/2017 11 03/o1 mf 1 23 dzr
sjwnh_.arc
Archive Log          508    03-NOV-17
/u01/app/oracle/fast_recovery_area/DXDB/archivelog/2017_11_03/o1_mf_1_24_dzr
sjyng .arc
Archive Log          509    03-NOV-17
/u01/app/oracle/fast_recovery_area/DXDB/archivelog/2017_11_03/o1_mf_1_25_dzr
sk7vy_.arc
Archive Log          510    03-NOV-17
/u01/app/oracle/fast recovery area/DXDB/archivelog/2017 11 03/o1 mf 1 26 dzr
sk8xt_.arc
Archive Log          511    03-NOV-17
/u01/app/oracle/fast_recovery_area/DXDB/archivelog/2017_11_03/o1_mf_1_27_dzr
spf5r_.arc
Archive Log          512    03-NOV-17
/u01/app/oracle/fast recovery area/DXDB/archivelog/2017 11 03/o1 mf 1 28 dzr
spw8q_.arc
Archive Log          513    03-NOV-17
/u01/app/oracle/fast_recovery_area/DXDB/archivelog/2017_11_03/o1_mf_1_29_dzr
sqogw_.arc
```

```
Archive Log          514    03-NOV-17
/u01/app/oracle/fast_recovery_area/DXDB/archivelog/2017_11_03/o1_mf_1_30_dzr
sqsos_.arc
Archive Log          515    03-NOV-17
/u01/app/oracle/fast recovery area/DXDB/archivelog/2017 11 03/o1 mf 1 31 dzr
sqspg_.arc
Archive Log          516    03-NOV-17
/u01/app/oracle/fast_recovery_area/DXDB/archivelog/2017_11_03/o1_mf_1_32_dzr
sr82t_.arc
Archive Log          517    03-NOV-17
/u01/app/oracle/fast recovery area/DXDB/archivelog/2017 11 03/o1 mf 1 33 dzr
st62q_.arc
Archive Log          518    03-NOV-17
/u01/app/oracle/fast_recovery_area/DXDB/archivelog/2017_11_03/o1_mf_1_34_dzr
stf8l_.arc
Archive Log          519    03-NOV-17
/u01/app/oracle/fast recovery area/DXDB/archivelog/2017 11 03/o1 mf 1 35 dzr
stg45_.arc
Archive Log          520    03-NOV-17
/u01/app/oracle/fast_recovery_area/DXDB/archivelog/2017_11_03/o1_mf_1_36_dzr
t1p99_.arc
Archive Log          521    03-NOV-17
/u01/app/oracle/fast recovery area/DXDB/archivelog/2017 11 03/o1 mf 1 37 dzr
t288p .arc
Archive Log          522    03-NOV-17
/u01/app/oracle/fast_recovery_area/DXDB/archivelog/2017_11_03/o1_mf_1_38_dzr
tgjqp_.arc
handle=/u01/app/oracle/fast recovery area/DXDB/autobackup/2017 11 17/o1 mf s
_960316720_f0yl1w7q_.bkp RECID=8 STAMP=960334572
deleted backup piece
backup piece
handle=/u01/app/oracle/fast_recovery_area/DXDB/autobackup/2017_11_17/o1_mf_s
_960316720_f0yl2yb2_.bkp RECID=9 STAMP=960334606
Deleted 39 objects


RMAN>
```

Congratulation!

You've successfully finished Part IV: Implementing Backup and Recovery Workshop.



Part V: Empowering Snapshot Standby Database

This workshop demonstrate highly flexible Data Guard feature through the use of using Snapshot Standby database. The snapshot database is a temporary read-write database that is mostly used by the database replay for performance analysis and application testing. The snapshot standby database requires enabling the flashback database feature.

Tasks

Table 10 lists the tasks associated with Part V, Empowering Snapshot Database

Task#	Description
1	Verify Flashback Database Enabled
2	Show Current Data Guard Configuration
3	Convert the Physical Standby to Snapshot Standby Database
4	Show Current Data Guard Configuration
5	Perform DML Test
6	Convert to Snapshot to Physical Standby Database
7	Show Current Data Guard Configuration
8	Verify DML Flashed Back

Table 10: Empowering Snapshot Standby Database

Architecture

One of the key features of Oracle Data Guard is the Snapshot standby database. The snapshot standby database is a read-write database with flashback database on. It's primarily used for quick testing environment such as database replay. Figure 12 shows the snapshot standby database components.

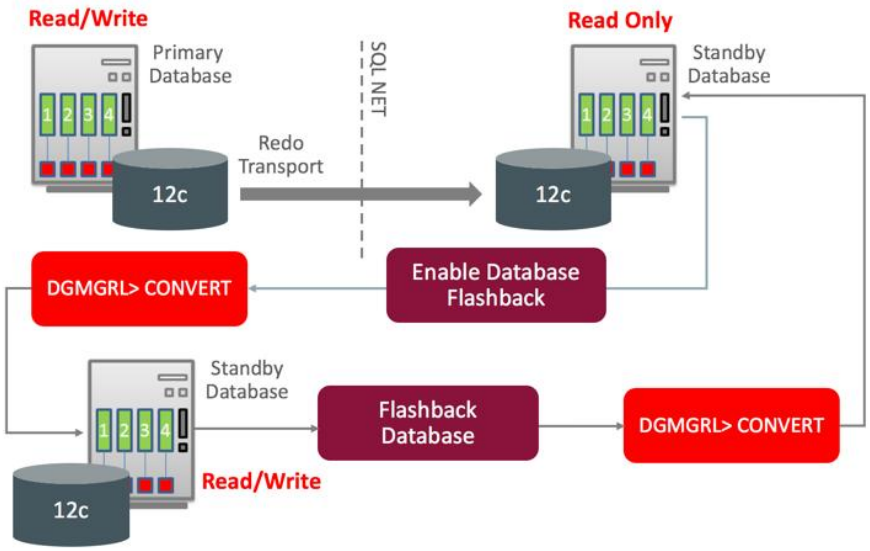


Figure 12: Snapshot Standby Database

The flashback database is enabled automatically when converting the physical standby database into snapshot standby database. As illustrated in Figure 11:

1. Flashback Database
2. Data Guard Convert the database into read/write
3. Apply database flashback
4. Convert the database back into physical standby database

S Verify Flashback Database Enabled

Snapshot standby database uses the database flashback to restore it self. Ensuring the flashback is enabled is essential.

☐ Complete

```
SQL> conn / as sysdba
Connected.
SQL> select flashback_on
   2  from   v$database;

FLASHBACK ON
------------------
YES

SQL>
```

O Show Current Data Guard Configuration

Verify the current state of the data guard. The configuration must indicate DHDB is the primary database and DXDB is the standby database.

☐ Complete

```
[oracle@dgobserver ~]$ dgmgrl
DGMGRL for Linux: Version 12.1.0.1.0 - 64bit Production

Copyright (c) 2000, 2012, Oracle. All rights reserved.

Welcome to DGMGRL, type "help" for information.
DGMGRL> connect sys/oracle@DHDB
Connected as SYSDBA.
DGMGRL> show configuration;

Configuration - DG_CONF

  Protection Mode: MaxPerformance
  Databases:
  DHDB - Primary database
    DXDB - Physical standby database

Fast-Start Failover: DISABLED

Configuration Status:
SUCCESS

DGMGRL>
```

O Convert the Physical Standby to Snapshot Standby Database

Use the data guard broker command CONVERT to convert the standby database to snapshot database.

Complete

```
DGMGRL> CONVERT DATABASE 'DXDB' TO SNAPSHOT STANDBY;
Converting database "DXDB" to a Snapshot Standby database, please wait...
Database "DXDB" converted successfully
DGMGRL>
```

O Show Current Data Guard Configuration

The current data guard configuration must indicate DHDB is the primary database and the DXDB is the snapshot database. This snapshot database is used for application testing and database replay.

‘

Complete

```
DGMGRL> show configuration

Configuration - DG CONF

  Protection Mode: MaxPerformance
  Databases:
  DHDB - Primary database
    DXDB - Snapshot standby database

Fast-Start Failover: DISABLED

Configuration Status:
SUCCESS

DGMGRL>
```

S Perform DML Test

A simple test is performed by inserting and committing one row to the DISCOUNT_TYPES table. Issue the SQL statements: SELECT, INSET and SELECT from DISCOUNT_TYPES table.

Complete

```
SQL> conn / as sysdba
Connected.
SQL> desc osm_repapi.discount_types
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------
 DT_CODE                                   NOT NULL VARCHAR2(8)
```

```
 DT_DESC                                   NOT NULL VARCHAR2(128)
 DT_PERC                                            NUMBER

SQL> select dt_code from  osm_repapi.discount_types;

DT_CODE
--------
SA
SAB
SCH

SQL> insert into  osm_repapi.discount_types values
  2  ('TST', 'TEST', null);

1 row created.

SQL> commit;

Commit complete.

SQL> select dt_code from  osm_repapi.discount_types;

DT CODE
--------
SA
SAB
SCH
TST

SQL>
```

Convert to Snapshot to Physical Standby Database

Issue the data guard command CONVERT to transform the snapshot database back to standby database. The COVERT command uses the database flashback logs to rewind the database to the restore point prior to the data guard COVERT to transform the database to snapshot standby database.

Complete

```
DGMGRL> connect sys/oracle@DXDB
Connected as SYSDBA.
DGMGRL> CONVERT DATABASE 'DXDB' TO PHYSICAL STANDBY;
Converting database "DXDB" to a Physical Standby database, please wait...
Database "DXDB" converted successfully
DGMGRL>
```

Show Current Data Guard Configuration

The current data guard configuration must indicate the primary database DHDB and the original snapshot database have been converted back to physical standby database.

Complete

```
DGMGRL> show configuration;

Configuration - DG_CONF

  Protection Mode: MaxPerformance
  Databases:
  DHDB - Primary database
    DXDB - Physical standby database

Fast-Start Failover: DISABLED

Configuration Status:
SUCCESS

DGMGRL>
```



Verify DML Flashed Back

Using the database flashback logs and the restore point, the inserted data is removed without performing FLASHBACK command.



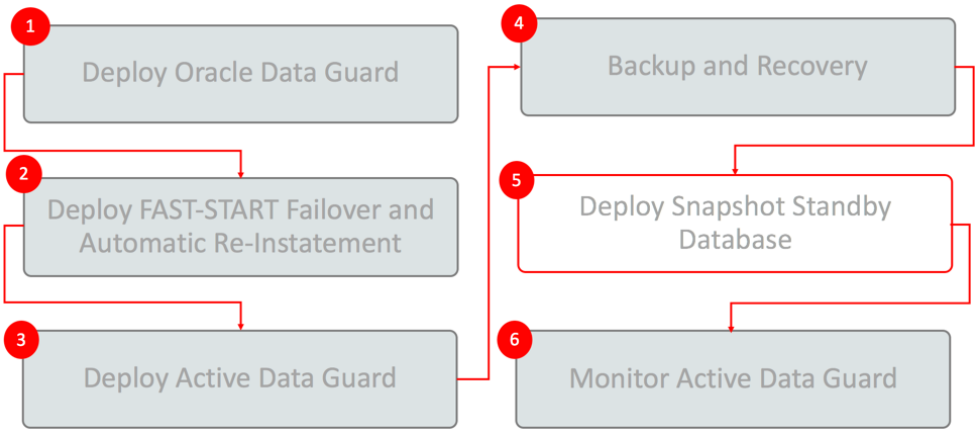

```
SQL> conn / as sysdba
Connected.
SQL> select dt_code from  osm_repapi.discount_types;

DT_CODE
--------
SA
SAB
SCH

SQL>
```

Congratulation!

You've successfully finished Part V: Configure Snapshot Standby Database Workshop



Part VI: Monitor Active Data Guard

This workshop is for looking into the database running within data guard configuration to monitor the Redo-Transport and the Redo-Apply services. Finding out the cause of FAST-START failover and the various areas to troubleshoot, and fix database errors running on Oracle Active Data Guard Environment.

Tasks

Table 11 lists the tasks associated with Part IV, Monitor Active Data Guard

Task#	Description
1	Verify Online Standby Redo Log Files
2	View Standby Redo Log Information
3	Show Data Guard Configuration
4	Show Database Status
5	Data Guard Reports
6	Enable Redo Transport Compression
7	Monitor Redo Apply
8	Monitor Apply Lag
9	Monitor Block Change Tracking

Table 11: Monitor Active Data Guard tasks

Architecture

Oracle Data Guard is an integral part of Oracle Database (RDBMS). Figure 13 shows Oracle Data Guard components. Depending on the protection mode and the apply mode, the process flow may slightly altered. The architecture depicts the primary database (DHDB) is protected by one standby database (DXDB).

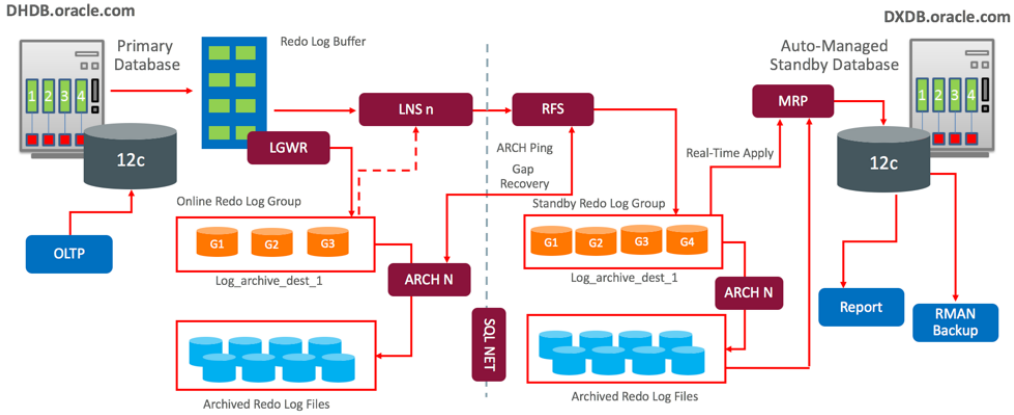


Figure 13: Oracle Data Guard Architecture

- The Primary Database (DHDB) runs OLTP applications; the LGWR writes committed transactions to the database online redo log files.
- The archive processes create the archived redo log file.
- For each destination configured for redo-transport, an LNS process is spawn to send data to the remote RFS process; this can be SYNC or ASYNC mode.
- The RFS writes to the standby online redo log files.
- For real-time apply, the MRP process applies the transactions to the standby database, for non real-time, the archive processes write to the standby archived redo log files, which are applied to the standby database by the MPR process.

The physical auto-managed standby database is enabled for Active Data Guard (ADG); the instance is able to perform query offloading for running reports application and enabled for optimized RMAN backup and automatic block repair.

Because this is a physical standby database, backup of the primary database can be executed on the standby database. Performing backup on standby database system using RMAN is a valid backup for the primary database.

The redo-Transport service uses TCP/IP to ship the transaction logs to the standby database standby database (DXDB) for supporting the Data Guard configuration.

P Verify Online Standby Redo Log Files

Verify the instance standby redo log files. Using real-time apply requires the addition of standby redo log files.

Complete

```
SQL> SELECT group#||' - '||member
  2  FROM v$logfile
  3  WHERE type = 'STANDBY';

GROUP#||'-'||MEMBER
--------------------------------------------------------------------------------
4 - /u01/app/oracle/oradata/DHDB/redo04.log
5 - /u01/app/oracle/oradata/DHDB/redo05.log
6 - /u01/app/oracle/oradata/DHDB/redo06.log
7 - /u01/app/oracle/oradata/DHDB/redo07.log

SQL>
```

S View Standby Redo Log Information

View the current redo log file information. It should indicate only one active group.

Complete

```
SQL> SELECT group#, dbid, archived, status
  2  FROM v$standby_log;

    GROUP# DBID                                 ARC STATUS
---------- ---------------------------------------- --- ----------
         4 2763595033                           YES ACTIVE
         5 UNASSIGNED                           NO  UNASSIGNED
         6 UNASSIGNED                           YES UNASSIGNED
         7 UNASSIGNED                           YES UNASSIGNED

SQL>
```

O Show Data Guard Configuration

Display the data guard configuration details. It should indicate the primary database, the standby database, the protection mode, and the configuration status.

Complete

```
DGMGRL> show configuration

Configuration - DG_CONF

  Protection Mode: MaxPerformance
  Databases:
  DHDB - Primary database
```

```
    DXDB - Physical standby database

Fast-Start Failover: DISABLED

Configuration Status:
SUCCESS

DGMGRL>
```

0 Show Database Status

Display the database status details for DXDB.

Complete

```
DGMGRL> show database 'DXDB';

Database - DXDB

  Role:               PHYSICAL STANDBY
  Intended State:     APPLY-ON
  Transport Lag:      0 seconds (computed 1 second ago)
  Apply Lag:          0 seconds (computed 1 second ago)
  Apply Rate:         0 Byte/s
  Real Time Query:    ON
  Instance(s):
    DXDB

Database Status:
SUCCESS

DGMGRL>
```

Display the database status details for DHDB.

Complete

```
DGMGRL> SHOW DATABASE 'DHDB'

Database - DHDB

  Role:               PRIMARY
  Intended State:     TRANSPORT-ON
  Instance(s):
    DHDB

Database Status:
SUCCESS

DGMGRL>
```

In-depth details for the database properties.

Complete

```
DGMGRL> show database verbose 'DXDB';

Database - DXDB
```

```
  Role:               PHYSICAL STANDBY
  Intended State:     APPLY-ON
  Transport Lag:      0 seconds (computed 0 seconds ago)
  Apply Lag:          0 seconds (computed 0 seconds ago)
  Apply Rate:         0 Byte/s
  Real Time Query:    ON
  Instance(s):
    DXDB

  Properties:
    DGConnectIdentifier             = 'DXDB'
    ObserverConnectIdentifier       = ''
    LogXptMode                      = 'ASYNC'
    RedoRoutes                      = ''
    DelayMins                       = '0'
    Binding                         = 'optional'
    MaxFailure                      = '0'
    MaxConnections                  = '1'
    ReopenSecs                      = '300'
    NetTimeout                      = '30'
    RedoCompression                 = 'DISABLE'
    LogShipping                     = 'ON'
    PreferredApplyInstance          = ''
    ApplyInstanceTimeout            = '0'
    ApplyLagThreshold               = '0'
    TransportLagThreshold           = '0'
    TransportDisconnectedThreshold  = '30'
    ApplyParallel                   = 'AUTO'
    StandbyFileManagement           = 'AUTO'
    ArchiveLagTarget                = '0'
    LogArchiveMaxProcesses          = '4'
    LogArchiveMinSucceedDest        = '1'
    DbFileNameConvert               = '/u01/app/oracle/oradata/DHDB,
/u01/app/oracle/oradata/DXDB'
    LogFileNameConvert              = '/u01/app/oracle/oradata/DHDB,
/u01/app/oracle/oradata/DXDB'
    FastStartFailoverTarget         = 'DHDB'
    InconsistentProperties          = '(monitor)'
    InconsistentLogXptProps         = '(monitor)'
    SendQEntries                    = '(monitor)'
    LogXptStatus                    = '(monitor)'
    RecvQEntries                    = '(monitor)'
    StaticConnectIdentifier         =
'(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=stdby01.oracle.com)(PORT=1521))(C
ONNECT_DATA=(SERVICE_NAME=DXDB_DGMGRL.oracle.com)(INSTANCE_NAME=DXDB)(SERVER
=DEDICATED)))'
    StandbyArchiveLocation          = 'USE DB RECOVERY FILE DEST'
    AlternateLocation               = ''
    LogArchiveTrace                 = '0'
    LogArchiveFormat                = 'DXDB_%t_%s_%r.arc'
    TopWaitEvents                   = '(monitor)'

Database Status:
SUCCESS

DGMGRL>
```

Data Guard Reports

The output of these reports should be empty, which indicate the data guard configuration is healthy, running without major issues.

Complete

```
DGMGRL> connect sys/oracle@DHDB
Connected as SYSDBA.
DGMGRL> SHOW DATABASE 'DHDB' 'StatusReport';
STATUS REPORT
       INSTANCE_NAME   SEVERITY ERROR_TEXT

DGMGRL> SHOW DATABASE 'DHDB' 'LogXptStatus';
LOG TRANSPORT STATUS
PRIMARY_INSTANCE_NAME STANDBY_DATABASE_NAME               STATUS
                DHDB                  DXDB

DGMGRL> SHOW DATABASE 'DHDB' 'InconsistentProperties';
INCONSISTENT PROPERTIES
   INSTANCE_NAME        PROPERTY_NAME         MEMORY_VALUE
SPFILE_VALUE         BROKER_VALUE

DGMGRL> SHOW DATABASE 'DHDB' 'InconsistentLogXptProps';
INCONSISTENT LOG TRANSPORT PROPERTIES
   INSTANCE_NAME         STANDBY_NAME         PROPERTY_NAME
MEMORY_VALUE         BROKER_VALUE

DGMGRL>
```

Enable Redo Transport Compression

The compression feature enables the optimized redo-transport. This feature requires Oracle advanced compression option.

Complete

```
DGMGRL> EDIT DATABASE 'DHDB'
> SET PROPERTY 'RedoCompression'='ENABLE';
Property "RedoCompression" updated
DGMGRL> EDIT DATABASE 'DXDB'
> SET PROPERTY 'RedoCompression'='ENABLE';
Property "RedoCompression" updated
DGMGRL>
```

Monitor Redo Apply

Monitor the data guard apply service. Investigate the MPR0 data guard apply process.

Complete

```
SQL> SELECT process, status, group#, thread#
```

```
  2  FROM v$managed_standby
  3  order by process, group#, thread#;

PROCESS   STATUS       GROUP#                                    THREAD#
--------- ------------ ----------------------------------------- ----------
ARCH   CLOSING      4                                           1
ARCH   CLOSING      5                                           1
ARCH   CLOSING      5                                           1
ARCH   CONNECTED    N/A                                         0
MRP0   APPLYING_LOG N/A                                         1
RFS    IDLE         2                                           1
RFS    IDLE         N/A                                         0
RFS    IDLE         N/A                                         0

8 rows selected.

SQL>
```



Monitor Apply Lag

Monitor the apply lag, which indicates how much the standby database behind the primary database.




```
SQL> SELECT name, value, datum_time, time_computed
  2  FROM v$dataguard_stats
  3  WHERE name like 'apply lag';

NAME
--------------------------------
VALUE
----------------------------------------------------------------
DATUM_TIME                 TIME_COMPUTED
-------------------------- ------------------------------
apply lag
+00 00:00:00
11/18/2017 02:34:59        11/18/2017 02:35:01


SQL>
```

Monitor the apply lag based on the database histogram.




```
SQL> SELECT * FROM V$STANDBY_EVENT_HISTOGRAM
  2  WHERE NAME = 'apply lag' AND COUNT > 0 and
  3  rownum < 6;

NAME                                                             TIME
---------------------------------------------------------------- ----------
UNIT              COUNT LAST TIME UPDATED      CON ID
----------------- ---------- -------------------- ----------
apply lag                                                         0
seconds           24770 11/18/2017 02:33:26        0
```

```
apply lag                                                                1
seconds               1 11/17/2017 23:51:35         0

apply lag                                                                2
seconds               1 11/17/2017 18:38:42         0


NAME                                                                  TIME
----------------------------------------------------------------- ----------
UNIT              COUNT LAST_TIME_UPDATED      CON_ID
---------------- ---------- -------------------- ----------
apply lag                                                                3
seconds               2 11/17/2017 18:38:43         0

apply lag                                                                4
seconds               2 11/17/2017 18:38:44         0


SQL>
```

Monitor Block Change Tracking

Monitor incremental backup on the standby database. This query shows the enhancement when using the database block change tracking feature.

Complete

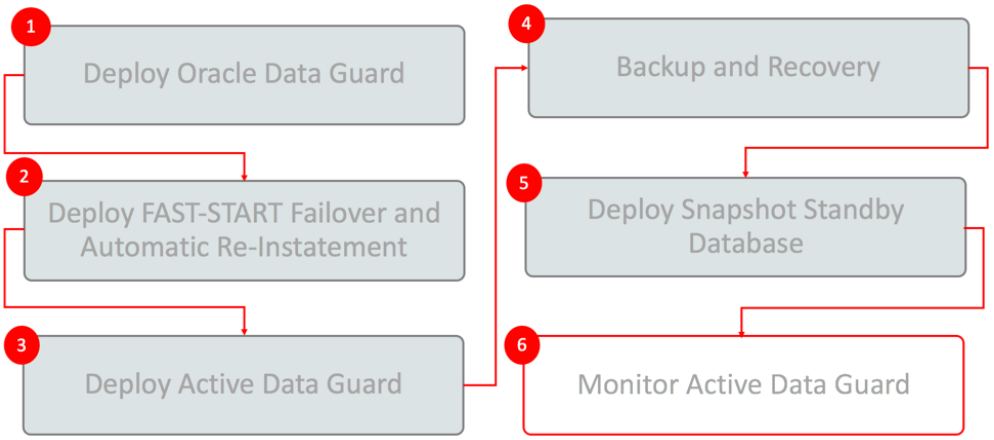
```
SQL> SELECT file#, avg(datafile_blocks),
  2  avg(blocks_read),
  3  avg(blocks_read/datafile_blocks)
  4  * 100 AS PCT_READ_FOR_BACKUP,
  5  avg(blocks)
  6  FROM   v$backup_datafile
  7  WHERE  used_change_tracking = 'YES'
  8  AND    incremental_level > 0
  9  GROUP  BY file#;

     FILE# AVG(DATAFILE_BLOCKS) AVG(BLOCKS_READ) PCT_READ_FOR_BACKUP
AVG(BLOCKS)
---------- ------------------- ---------------- ------------------- ------
         1              101120             9891         9.78144778       5087
         6               51360             1233         2.40070093        414
         4               18560             4171         22.4730603       4110
         3              102400            18459         18.0263672      10782

SQL>
```

Congratulation!

You've successfully finished Part VI: Monitor Data Guard Workshop



Bi-Directional Topology for Business Continuity

APPEXNDX

B

Overview

The objective of this chapter is to use Oracle GoldenGate 12c two-node configuration for building high availability infrastructure to support business applications maximum availability by minimizing disruption due to system failure. Oracle customers have been implementing Oracle real application clusters (RAC) for high availability and Oracle Data Guard (DG) for disaster recovery. The two solutions co-exist for best continuity. RAC and DG require an advanced level of skills to implement and manage together. RAC and DG are the major components of Oracle Maximum Availability Architecture (MAA) for sites running mission critical systems and requiring a maximum level of scalability, data availability, and data protection. For applications that may compromise RAC embedded scalability and DG fast start failover, Oracle GoldenGate achieves the requirements for business continuity with a sustainable budget, operational cost, and reasonable infrastructure complexity.

The flexibility of Oracle GoldenGate advanced configuration enables the software to provide business continuity similar to RAC and DG combined using active-passive and active-active bi-directional topology architecture. However, this specific implementation is not out-of-the-box and requires a considerable level of Oracle GoldenGate and database experience in respect to the database design and applications coding.

This chapter implements Oracle GoldenGate for business continuity using similarly designed architecture for deploying:

- Active-Passive Model for Disaster Recovery (DR)
- Active-Active Model for High Availability (HA)

Unlike Oracle Real Application Clusters (RAC) in terms of scalability, Oracle GoldenGate does not promote the built-in scalability provided by RAC, which is the core function of RAC to support linearly increasing transactions volume. When scalability is an important element of the application, sufficient evaluation is necessary before deploying Oracle GoldenGate for high availability. Also, with Oracle Real Application Clusters (RAC), increasing the number of nodes delivers higher scalability.

However, Oracle GoldenGate encourages the use of two-node architecture only for high-availability. This is recommended to avoid complex data conflict detection and resolutions in an active-active bi-directional topology.

For an enterprise, Oracle GoldenGate does not replace RAC or DG, rather it is implemented to complement and integrate with the Oracle clusterware and Oracle physical standby database. Refer to Chapter 12 to learn more about integrating Oracle GoldenGate with RAC and DG.

For sites running the Oracle Database Standard Edition, Oracle GoldenGate is an attractive option for implementing high availability (HA) and disaster recovery (DR). The option is cost-effective and highly suitable for in-house developed applications, but needs to be developed from the group-up so that the applications are replication-aware. In deploying Oracle GoldenGate on bi-directional active-active topology, the challenge is to handle data conflicts which are discussed using the following:

- Data and Applications Segmentation
- Oracle GoldenGate Built-In Conflict Detection and Resolution (CDR)
- Custom Conflict Detection and Resolution using PL/SQL Sub-Programs Development
- Oracle Virtual Private Database (VPD)

GoldenGate Business Continuity Architecture

Oracle GoldenGate bi-directional architecture achieves better performance by splitting the load across two instances. This sustains fault tolerance when one of the two instances fail by moving all processing to the surviving system. However, the top advantage of using Oracle GoldenGate bi-directional architecture is the geographically separated data centers that enable cost-effective business continuity.

Figure 5-1 depicts bi-directional topology components needed for business continuity. The data flow is from the primary system (the source) to the live standby system (the target) and vice versa. The primary system database S1E2 is on Oracle Database 11g and the live standby system database T1E2 is on Oracle Database 11g. The Oracle GoldenGate building blocks of extract groups, delivery groups, and related components for data movement are outlined below.

The Primary System (The Source) Components

- Extract groups, the primary extract ES1E203, the local trail ./dirdat/sc and the secondary extract PS1E203.

- The delivery group, the dynamic collector server, the remote trail ./dirdat/tc and the delivery process RS1E203.

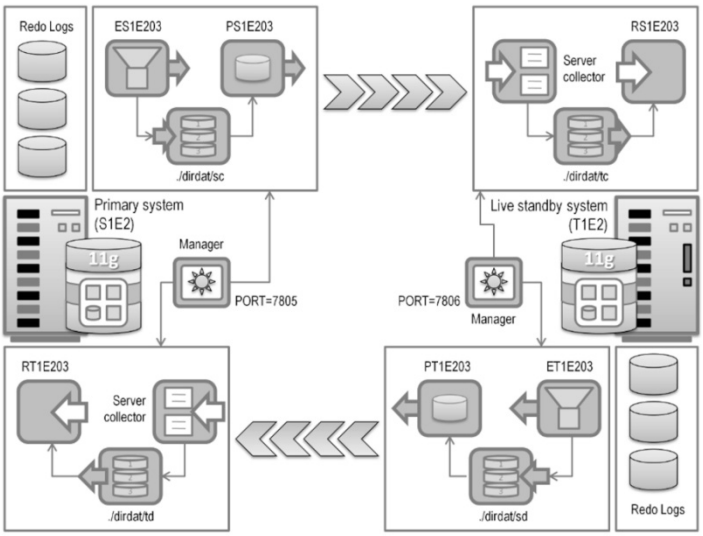


Figure 5-1: Bi-Directional Topology Building Blocks

The Live Standby System (The Target) Components

- Extract groups, the primary extract ET1E203, the local trail ./dirdat/sd and the secondary extract PT1E203.

- The delivery group, the dynamic collector server, the remote trail ./dirdat/td and the delivery process RT1E203.

Database Configuration

The primary system database and live standby database are operating on ARCHIVELOG mode. The transaction logs (online redo log files) structure of both databases are identical. Each has 3 groups used by the primary extract for performing change data capture (CDC). Online redo log files are archived and stored in the fast recovery area (FRA) for a recovery window of (7) days in case of being required by the primary extract or the recovery manager (RMAN) recover operation. The below RUN command backup the archived log files while maintaining a recovery window of (7) days of archived files in the FRA. Deleting files that are not required by the recovery window of (7) days.

```
RMAN> RUN {
2> BACKUP ARCHIVELOG ALL;
```

```
3> DELETE COPY OF ARCHIVELOG ALL COMPLETED BEFORE 'SYSDATE-7';
4> }
```

At minimum, set the size of the recovery window equal to or greater than the value of the database parameter CONTROL_FILE_RECORD_KEEP_TIME. The default is (7) days, unless a recovery catalog is configured. Also, ensuring FRA storage capacity allows retaining (7) days of archived log files by setting the database parameter DB_RECOVERY_FILE_DEST_SIZE as needed.

Extending Oracle GoldenGate for High Availability

When using Oracle GoldenGate for business continuity, the task of migrating users from the primary system to standby system and vice versa requires the database administrator (DBA) intervention and users reconnect to the surviving instance. However, for an active-active environment, the application needs the load balancing technique for connecting sessions to the database with minimal workload. Figure 5-2 shows the load balancer workflow that dynamically routes the application to the database with minimal workload.

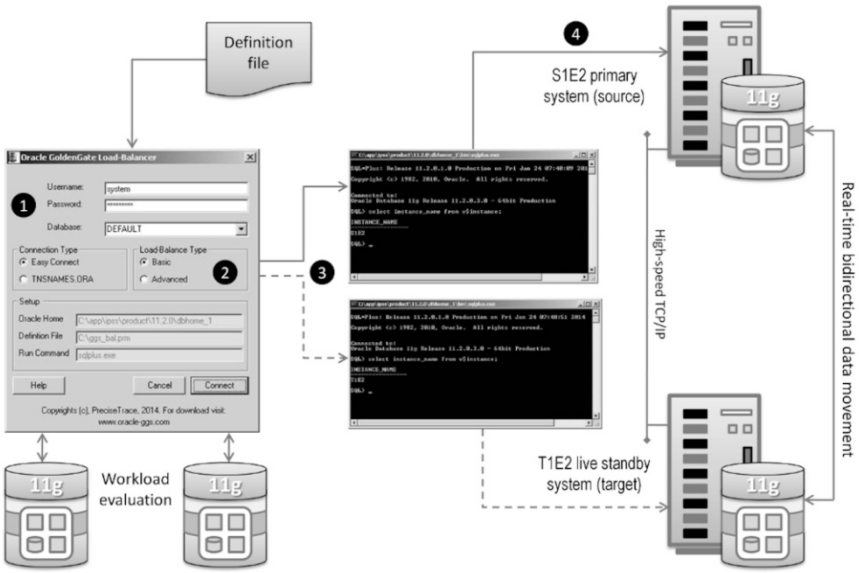


Figure 5-2: High Availability Load Balancer

The load balancer negotiates the primary and standby database to determine the preferred instance based on current workload. Then, it uses the selected connection type to establish the connection to the primary or standby systems – the database

with lesser workload. The load balancer uses a definition file that defines the primary and standby systems. Each line is composed of the host name or IP address, the listener port, the database service name, and the run command. The definition file c:\ggs-bal.prm is mandatory and the database credential must be identical for the primary system and live standby system database. The following lines define the primary and live standby systems configuration.

```
ggs-source   1521   s1e2.precisetrace.com    sqlplus.exe
ggs-target   1521   t1e2.precisetrace.com    sqlplus.exe
```

The above load balancer maintains the workload between the primary and the live-standby systems by performing the following:

1. Connecting to the primary and standby systems for evaluating the current database workload.
2. Comparing the two workloads to determine the preferred database instance – the one with lesser workload.
3. Using the selected connection type to compose the database connect string.
4. Routing and spawning the application to the preferred system.

When using Oracle GoldenGate for high-availability (active-active), the number of Oracle GoldenGate database instances should be limited to a maximum of (3) database instances to avoid complex conflicts resolution and configuration. To download the load balancer utility, visit http://www.oracle-ggs.com.

Disaster Recovery

When using Oracle GoldenGate to implement a disaster recovery (DR) site, it is developed as *active-passive*, bi-directional topology architecture. The source database is designated as the primary system and the target database is designated as the live-standby system. The live-standby system database remains on read-write mode. Applications users are connected to either the primary or the live-standby system, but not both. The live-standby system remains ready for the failover and switchover situations. Routing connection to one-and-only-one system is mostly handled from within the application servers as depicted on Figure 5-3 where sessions are established from either the primary or the live-standby system database.

The tremendous advanced features, configurable components flexibility, communication robustness and extendibility of Oracle GoldenGate have technically established a distinctive architecture for designing an enterprise heterogeneous platform suitable for protection and integration the enterprise data. Having a live-standby database continuously synchronized in real-time maximizes the usability of the database for other applications. Another distinctive advantage of Oracle GoldenGate is heterogeneity.

The capability of the live-standby system to run on a dissimilar platform from the primary system attracts decision makers and technical team members to build a sophisticated, one-of-a-kind disaster recovery site. Mostly designed to run on a mid-range Intel-based system powered by Oracle Linux. It delivers high-quality service and low-cost per transaction with best of class stability and reliability.

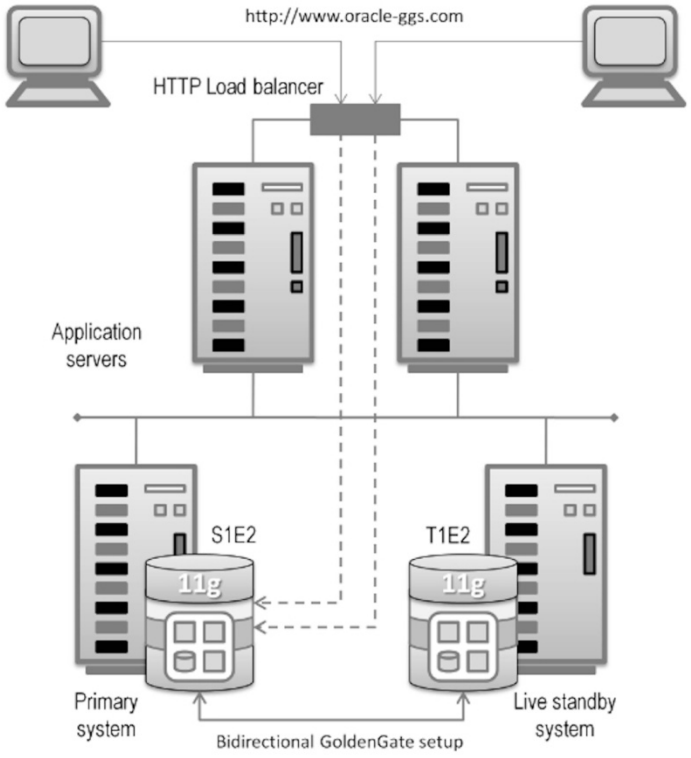


Figure 5-3: Disaster Recovery Connections

Oracle GoldenGate Value Proposition

A value proposition of Oracle GoldenGate is the certification with Oracle Database Standard Edition (SE). Small to mid-range server sites running Oracle Database SE can develop their disaster recovery solution using Oracle GoldenGate only. The demonstration of this chapter was developed using Oracle Database 11g SE running on Oracle Linux. This value proposition should not be underestimated!

```
SQL> SELECT * FROM v$version;

BANNER
----------------------------------------------------------------
Oracle Database 11g Release 11.2.0.3.0 - 64bit Production
```

```
PL/SQL Release 11.2.0.3.0 - Production
CORE    11.2.0.3.0    Production
TNS for Linux: Version 11.2.0.3.0 - Production
NLSRTL Version 11.2.0.3.0 - Production

SQL>
```

Disaster Recovery Considerations

The concept of disaster recovery (DR) imposes the standby database to operate on read-only mode at all times. However, because Oracle GoldenGate requires that the database operates on read-write, it is only the synchronized tables that are restricted from DML operation outside of the Oracle GoldenGate apply session (Replicat). Also, even though the bi-directional setup is already in place, it is active from only one direction at any time. Other guidelines to consider when using Oracle GoldenGate for disaster recovery are listed below:

Live-Standby System

At all times, the data flow is active from only one direction. The live-standby database (target) remains on read-write mode but fundamentally inactive – used only by the Replicat sessions. Under normal situations, OLTP users are connected to the primary system database (source) only. Utilizing the live-standby system for other applications should avoid interaction with mapped tables used by Oracle GoldenGate delivery processes (Replicats).

Operating Mode and Failover

The live-standby system database should be operated on ARCHIVELOG mode. This prepares the live-standby system for a planned switchover and unplanned failover. Oracle GoldenGate switchover and failover is an interactive operation and handled manually. Using tools such as Oracle GoldenGate Veridata and the Oracle GoldenGate Monitor are highly recommended to proactively avoid data divergence and minimizing latency.

Query Offloading

OLAP users are normally directed to the live-standby system database (target). It is the ideal platform for running long query reports, but the database should not be modified by creating additional objects such as indexes and materialized views. Otherwise, these created objects must be dropped as part of a failover/switchover operation.

Unsupported Data Types

The unsupported data types by Oracle GoldenGate is minimum. The most common applications data types not supported by Oracle GoldenGate is BFILE, which handled using the PL/SQL supplied API DBMS_LOB and DBMS_FILE_TRANSFER. Refer to Chapter 11 to learn more about using these packages with Oracle GoldenGate. Evaluating the extent of unsupported data types is necessary to qualify using Oracle GoldenGate for disaster recovery solution. Refer to Chapter 1 to learn more about unsupported data types.

Disaster Recovery Implementation

A disaster recovery implementation using Oracle GoldenGate consists of the tasks listed in Table 5-1. Testing switchover and failover for moving users from the primary system to the live-standby system and then moving users back to the primary system is the last step and executed for planned & unplanned scenarios.

Task	
1.	Install software, configure manager process, and start Oracle GoldenGate instance.
2.	Enable database supplemental logging and archive log mode.
3.	Create Oracle GoldenGate database user and grant privileges.
4.	Login to the database from GGSCI and Add table-level logging.
5.	Prepare GLOBALS parameters and the checkpoint table.
6.	Perform initial load from primary (source) to live standby (target) systems.
7.	Configure, create, and deploy Oracle GoldenGate groups for bi-directional topology.
8.	Enable data capture and delivery from primary to live standby system.
9.	Perform moving users for planned & unplanned switchover and failover.

Table 5-1: Bi-Directional Configuration for High Availability and Disaster Recovery Solutions

The destination for each of the above tasks is shown in Figure 5-4. Since this is a bi-directional topology architecture, the tasks are performed on the primary system/source database and on the live-standby system/target database.

The live-standby system should have adequate resources to support the post-failover/ switchover operation timeframe. Having matched resources is recommended. Even though Oracle GoldenGate supports a heterogeneous database environment, it is absolutely not recommended when implementing high availability and disaster recovery solutions using Oracle GoldenGate. The primary and live-standby platform should have identical architecture. The planned switchover and the unplanned failover is a manual process. Automation is developed using Linux shell script and Oracle GoldenGate obey files. The use of Oracle GoldenGate Veridata for data consistency and the Oracle GoldenGate Monitor are highly recommended for such an environment.

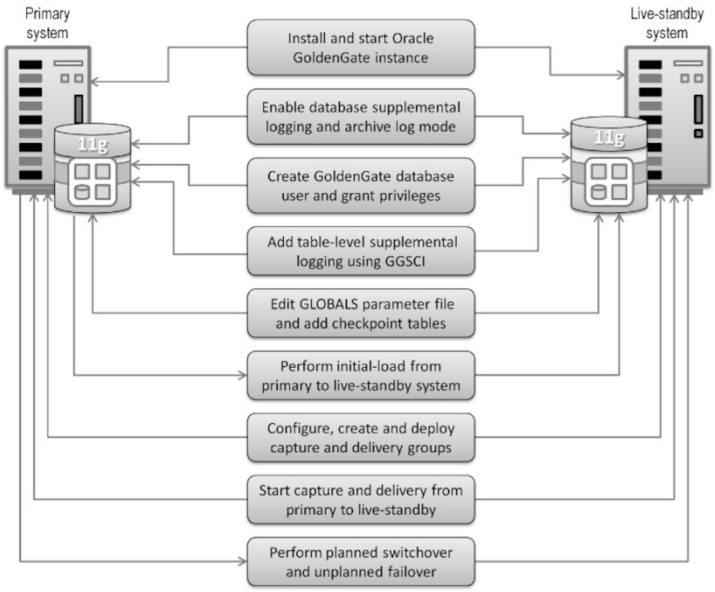


Figure 5-4: Bi-Directional Topology Architecture Tasks vs. Destinations

Software Installation

Successful installation of Oracle GoldenGate on the primary and live-standby systems must be successfully completed to proceed further. Consider the platform architecture. The chapter uses Oracle GoldenGate for Oracle Database 11g 64-bit. The same bundle of Oracle GoldenGate 12c provides support for Oracle database enterprise and standard editions. Refer to Chapter 2 to learn about installing Oracle GoldenGate.

Supplemental Logging

Enable supplemental logging on the primary and live-standby databases. Also, ensure the databases are operating on ARCHIVELOG mode.

```
SQL> CONN / AS SYSDBA
Connected.
SQL> SELECT log_mode FROM v$database;

LOG_MODE
------------
ARCHIVELOG

SQL> ALTER DATABASE ADD SUPPLEMENTAL LOG DATA;
```

```
Database altered.

SQL>
```

Refer to Chapter 2 to learn more about database supplemental logging settings. Even though the data flow is in one direction at any time, ensuring the database supplemental logging and ARCHIVELOG mode are enabled on the primary and Live-standby systems minimizes the meantime to migrate users and reverse the processing direction.

Database Users

Create Oracle GoldenGate database user on the primary and live-standby databases. The use of dedicated Oracle GoldenGate is required for active-active bi-directional topology. It is used for an endless loop detection purpose as explained in this chapter.

```
SQL> CREATE USER ggs_admin identified by oracle;

User created.

SQL> GRANT DBA TO ggs_admin;

Grant succeeded.

SQL> CONN ggs_admin/oracle
Connected.
SQL>
```

Refer to Chapter 2 to learn more about the precise database privileges granted to Oracle GoldenGate users on the primary (source) and live-standby (target) database. The above command grants DBA which is considered an over grant and not recommended or even possible for some sites.

Start Oracle GoldenGate Instances

The primary system and live-standby system are running on Oracle Database 11g Standard Edition 64-bit. The primary system Oracle GoldenGate instance manager process is listening on port 7805 and the live-standby Oracle GoldenGate manager process is listening on port 7806. Using the GoldenGate Software Command Interface (GGSCI), configure and start the manager process on the primary and live-standby systems.

```
GGSCI (apps-srv1) 6> START MGR
Manager started.

GGSCI (apps-srv1) 7> INFO MGR
```

```
Manager is running (IP port apps-srv1.7805, Process ID 13632).

GGSCI (apps-srv1) 8>
```

Repeat the above GGSCI commands for the live-standby system.

```
GGSCI (apps-srv2) 2> START MGR
Manager started.

GGSCI (apps-srv2) 3> INFO MGR

Manager is running (IP port apps-srv2.7806, Process ID 30133).

GGSCI (apps-srv2) 4>
```

Refer to Chapter 2 to learn more about the manager process common parameters settings for managing trails and unattended restart.

Checkpoint Table

The database checkpoint table is the default for the Replicat process checkpoint. Prepare the GLOBALS parameter file, then use GGSCI to create the checkpoint table. The GLOBALS parameter file contains the following one line only. Where GGS-ADMIN is the Oracle GoldenGate database schema.

```
CHECKPOINTTABLE ggs-admin.chkpttab

GGSCI (apps-srv1) 1> DBLOGIN USERID ggs_admin, PASSWORD oracle
Successfully logged into database.

GGSCI (apps-srv1) 2> ADD CHECKPOINTTABLE

No checkpoint table specified. Using GLOBALS specification
(ggs_admin.chkpttab)...

Successfully created checkpoint table ggs_admin.chkpttab.

GGSCI (apps-srv1) 3>
```

Create the checkpoint table on primary system and live-standby systems. Remember to exit and restart GGSCI to activate parameters added to GLOBALS parameter file.

Performing Initial-Data Load

This is a major task and must be delivered 100% successfully. The live-standby system database must be established prior to beginning the configuration tasks. Even though this demonstration is from Oracle-to-Oracle database (homogenous), file to Replicat Oracle GoldenGate initial-data load method is used to establish the live-standby (target)

database. Since we consider the initial load from an inactive database, the initial load is performed using the following steps:

- Configure and start the initial load Extract (ELOAD03)
- Configure the initial load Replicat (RLOAD03)
- Start the initial load Replicat (RLOAD03)

Initial Load Extract

Before starting the initial load extract, some analysis must be conducted to determine any unsupported column data types. Since this configuration is for developing a disaster recovery (DR), all tables must be included in the capture groups. For a very large list of tables, use the following PL/SQL code or the Extract and Replicat wild-card syntax format with exclude option to prepare the ELOAD03.PRM file.

```
SQL> CREATE DIRECTORY ggs_dir as '/d01/oracle/ggs/dirprm';

Directory created.

SQL> GRANT ALL ON DIRECTORY ggs_dir to public;

Grant succeeded.

SQL> DECLARE
  2      filehandler UTL_FILE.FILE_TYPE;
  3      CURSOR c1 is SELECT table_name
  4                   FROM   dba_tables
  5                   WHERE  owner='OSM$REPAPI';
  6      vtable_name VARCHAR2(30);
  7  BEGIN
  8      fileHandler := UTL_FILE.FOPEN('GGS_DIR', 'eload03.prm', 'W');
  9      UTL_FILE.PUTF(fileHandler, 'SOURCEISTABLE\n');
 10      UTL_FILE.PUTF(fileHandler, 'USERID ggs_admin, PASSWORD oracle\n');
 11      UTL_FILE.PUTF(fileHandler, 'RMTHOST apps-srv2, MGRPORT 7806\n');
 12      UTL_FILE.PUTF(fileHandler, 'RMTFILE ./dirdat/initload01.dat, PURGE\n');
 13      OPEN c1;
 14      LOOP
 15         FETCH c1 into vtable_name;
 16         EXIT WHEN c1%NOTFOUND;
 17         UTL_FILE.PUTF(fileHandler, 'TABLE osm$repapi.'||vtable_name||';\n');
 18      END LOOP;
 19      CLOSE c1;
 20      UTL_FILE.FCLOSE(fileHandler);
 21  EXCEPTION
 22      WHEN utl_file.invalid_path THEN
 23        raise_application_error(-20000, 'ERROR: Invalid GoldenGate path.');
 24  END;
 25  /

PL/SQL procedure successfully completed.

SQL>
```

```
GGSCI (apps-srv1) 1> EDIT PARAMS eload03
```

```
SOURCEISTABLE
USERID ggs_admin, PASSWORD oracle
RMTHOST apps-srv2, MGRPORT 7806
RMTFILE ./dirdat/initload03.dat, PURGE
TABLE osm$repapi.POLICY_TYPES;
TABLE osm$repapi.DISCOUNT_TYPES;
TABLE osm$repapi.CUSTOMERS;
TABLE osm$repapi.POLICIES;
```

```
GGSCI (apps-srv1) 1> SHELL ./extract pf ./dirprm/eload03.prm rf
./dirrpt/eload03.rpt
```

The GGSCI SHELL command allows you to invoke programs and operating system shell commands from within Oracle GoldenGate. The parameter SOURCEISTABLE directly reads records from the database tables, and must be the first parameter in the configuration file.

Initial Load Replicat

The initial load data file has been created on the live-standby system. Verify the existence of ./dirdat/initload03.dat on the live standby. Before starting the Replicat program, create a special run Replicat process (RLOAD03) as shown below.

```
GGSCI (apps-srv2) 1> ADD REPLICAT rload03, SPECIALRUN
REPLICAT added.

GGSCI (apps-srv2) 2>
```

```
GGSCI (apps-srv2) 2> EDIT PARAMS rload03
```

```
REPLICAT rload03
SPECIALRUN
ASSUMETARGETDEFS
HANDLECOLLISIONS
USERID ggs_admin, PASSWORD oracle
EXTFILE ./dirdat/initload03.dat
DISCARDFILE ./dirrpt/rload03.dsc, PURGE
MAP osm$repapi.*, TARGET osm$repapi.*;
END RUNTIME
```

```
GGSCI (apps-srv2) 1> SHELL ./replicat pf ./dirprm/rload03.prm rf
./dirrpt/rload03.rpt
```

Now, the live-standby database is established and identical to the primary database. Before proceeding further, verify ascertain the data using Oracle GoldenGate Veridata.

Also, because the initial-load is done from an inactive database, review Oracle GoldenGate report generated from the initial-load Replicat process.

```
GGSCI (apps-srv2) 4> VIEW REPORT ./dirrpt/rload03.rpt

Reading ./dirdat/initloa3.dat, current RBA 37987353, 200007 records

Report at 2014-01-04 17:27:04 (activity since 2014-01-04 17:26:08)

From Table OSM$REPAPI.POLICY_TYPES to OSM$REPAPI.POLICY_TYPES:
       #                   inserts:         4
       #                   updates:         0
       #                   deletes:         0
       #                  discards:         0
From Table OSM$REPAPI.DISCOUNT_TYPES to OSM$REPAPI.DISCOUNT_TYPES:
       #                   inserts:         3
       #                   updates:         0
       #                   deletes:         0
       #                  discards:         0
From Table OSM$REPAPI.CUSTOMERS to OSM$REPAPI.CUSTOMERS:
       #                   inserts:    100000
       #                   updates:         0
       #                   deletes:         0
       #                  discards:         0
From Table OSM$REPAPI.POLICIES to OSM$REPAPI.POLICIES:
       #                   inserts:    100000
       #                   updates:         0
       #                   deletes:         0
       #                  discards:         0

Last log location read:
     FILE:      ./dirdat/initload03.dat
     RBA:       37987353
     TIMESTAMP: 2014-01-04 14:24:58.801283
     EOF:       NO
     READERR:   400
```

As per the report details, the initial load has been completed successfully. Refer to Chapter 3 for more details regarding initial data load techniques and options.

Bi-Directional Topology Configuration

The basic task of configuring Oracle GoldenGate to support a disaster recovery site is to have a working bi-directional data flow, from the primary system to the live-standby system and vice versa. The purpose of the basic configuration is to test the capture, route, and delivery groups on a bi-directional environment using automated database transactions developed in PL/SQL and run from the command line. Figure 5-5 depicts Oracle GoldenGate groups that are configured next.

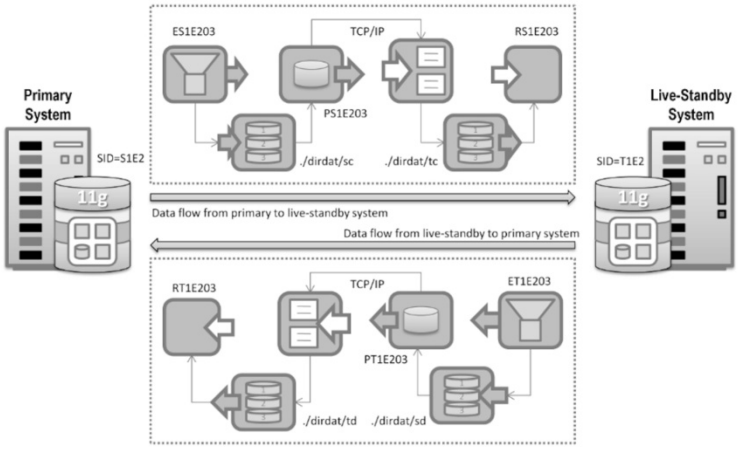


Figure 5-5: Bi-Directional Capture and Delivery Groups

After completing the bi-directional configuration, run and verify the transactional testing from both sides. The following procedure TRANS_LOAD is run as an anonymous PL/SQL block. To download the TRANS_LOAD source code, visit http://www.oracle-ggs.com.

```
SQL> BEGIN
  2     FOR i IN 100901..101000 LOOP
  3         trans_load(1, i);
  4     END LOOP;
  5  END;
  6  /

PL/SQL procedure successfully completed.

SQL>
```

Since the procedure TRANS_LOAD does not handle primary key collision, ensure the lower and upper range of the loop statement is correctly bounded to avoid primary key violation error.

Intermediate Systems

The use of an intermediate system between the primary and live-standby systems is an option. This configuration is only recommended when the current active system needs to ship trail files to a third system, not really part of this high availability bi-directional topology. Other reasons for using an intermediate system is to offload the current active system (primary system) from routing trail records to the remote system, or network security prevents either system from sending trail records, and the

need of a dedicated system for sending trail to the third system regardless of which system (primary or live-standby) is active. The disadvantage is increasing the latency, which increases the likelihood of data conflicts. Refer to Chapter 12 for details regarding intermediate systems.

Primary to Standby Data Flow

The configuration of the primary (source) to the live-standby (target) systems data flow is composed of deploying the primary & secondary extract groups and a delivery group. Figure 5-6 shows the components configured to move data from the primary to the live-standby system.

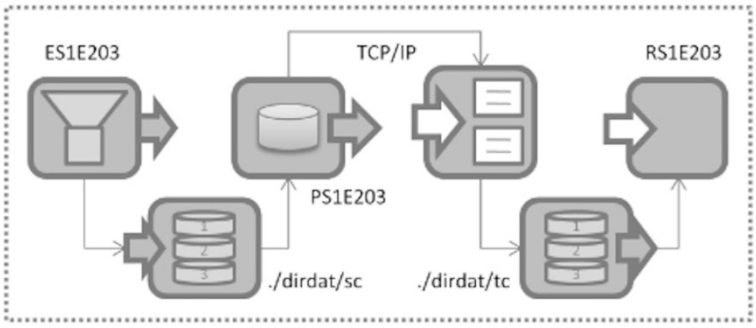


Figure 5-6: Primary to Standby Data Flow

Primary Extract Group

The primary group is composed of the parameter file ES1E203.PRM, the primary extract process ES1E203, the local trail ./dirdat/sc, and the checkpoint file ./dirchk/ES1E203.CPE. The primary extract ES1E203 is configured to capture from the transaction logs and writes to the local trail. The next two commands create the primary extract and its local trail. The local trail files is sized 128MB.

```
GGSCI (apps-srv1) 2> EDIT PARAMS es1e203
```

```
EXTRACT es1e203
SETENV  (ORACLE_SID=S1E2)
SETENV  (ORACLE_HOME="/d01/app/oracleSE")
SETENV  (NLS_LANG=AMERICAN_AMERICA.AR8ISO8859P6)
USERID  ggs_admin, password oracle
EXTTRAIL ./dirdat/sc
TABLE osm$repapi.CUSTOMERS;
TABLE osm$repapi.POLICIES;
```

```
GGSCI (apps-srv1) 3> ADD EXTRACT es1e203, TRANLOG, BEGIN now, THREADS 1
```

```
EXTRACT added.
```

```
GGSCI (apps-srv1) 5> ADD EXTTRAIL ./dirdat/sc, EXTRACT es1e203, MEGABYTES 128
EXTTRAIL added.

GGSCI (apps-srv1) 6>
```

Secondary Extract Group

The secondary group is composed of the parameter file PS1E203.PRM, the data pump extract process PS1E203, the remote trail ./dirdat/tc, and the checkpoint file ./dirchk/PS1E203.CPE. The data pump extract PS1E203 sends data over TCP/IP to the live-standby system. The next commands create the secondary group (data pump) and its remote trail. The remote trail files is sized 128MB.

```
GGSCI (apps-srv1) 6> EDIT PARAMS ps1e203
```

```
GGSCI (apps-srv1) 6> EDIT PARAMS ps1e203 EXTRACT ps1e203
USERID ggs_admin, PASSWORD oracle
PASSTHRU
RMTHOST apps-srv2, MGRPORT 7806
RMTTRAIL ./dirdat/tc
TABLE osm$repapi.CUSTOMERS;
TABLE osm$repapi.POLICIES;
```

```
GGSCI (apps-srv1) 7> ADD EXTRACT ps1e203, EXTTRAILSOURCE ./dirdat/sc
EXTRACT added.
```

```
GGSCI (apps-srv1) 11> ADD RMTTRAIL ./dirdat/tc, EXTRACT ps1e203, MEGABYTES 128
RMTTRAIL added.

GGSCI (apps-srv1) 12>
```

Delivery Group

The delivery group is composed of the parameter file RS1E203.PRM, the Replicat process PS1E203, the remote trail ./dirdat/tc, and the checkpoint file ./dirchk/ES1E203.CPR. The delivery process uses the live-standby database SQL Native Call to apply transactions to the database.

```
GGSCI (apps-srv2) 2> ADD REPLICAT rs1e203, EXTTRAIL ./dirdat/tc
REPLICAT added.

GGSCI (apps-srv2) 3>
```




Standby to Primary Data Flow

To support switchover and failover operations for planned and unplanned modes, the data flow from the standby to the primary systems must be in place. Figure 5-7 shows the data flow from the live-standby to the primary system.

The live-standby to primary data flow configuration remains inactive. It is activated only when users are moved from the primary system to the live-standby system. The application servers, where the applications are running, are configured to connect to one and only one specific system – either the primary system or the live-standby system, but not both.

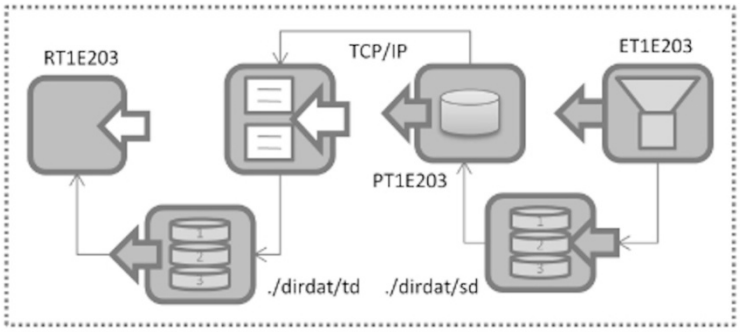


Figure 5-7: Standby to Primary Data Flow

Primary Extract Group

The primary group is composed of the parameter file ET1E203.PRM, the primary extract process ET1E203, the local trail ./dirdat/sd and the checkpoint file ./dirchk/ET1E203.CPE. The next commands create the primary extract and its local trail. The local trail files sized 128MB.

```
GGSCI (apps-srv2) 1> EDIT PARAMS et1e203
```

```
EXTRACT et1e203
SETENV (ORACLE_SID="T1E2")
SETENV (ORACLE_HOME="/d01/app/oracleSE/product/11.2.0/dbhome_1")
SETENV (NLS_LANG="AMERICAN_AMERICA.AL32UTF8")
USERID ggs_admin, PASSWORD oracle
EXTTRAIL ./dirdat/sd
TABLE osm$repapi.customers;
TABLE osm$repapi.policies;
```

```
GGSCI (apps-srv2) 5> ADD EXTRACT et1e203, TRANLOG, BEGIN now, THREADS 1
EXTRACT added.
```

```
GGSCI (apps-srv2) 6> ADD EXTTRAIL ./dirdat/sd, EXTRACT et1e203, MEGABYTES 128
EXTTRAIL added.

GGSCI (apps-srv2) 7>
```

Secondary Extract Group

The secondary group is composed of the parameter file PT1E203.PRM, the data pump extract process PT1E203, the remote trail ./dirdat/td and the checkpoint file ./dirchk/PT1E203.CPE. The next commands create the secondary group (data pump) and its remote trail. The remote trail files sized 128MB.

```
GGSCI (apps-srv2) 7> EDIT PARAMS pt1e203
```

```
EXTRACT pt1e203
USERID ggs_admin, PASSWORD oracle
PASSTHRU
RMTHOST apps-srv1, MGRPORT 7805
RMTTRAIL ./dirdat/td
TABLE osm$repapi.customers;
TABLE osm$repapi.policies;
```

```
GGSCI (apps-srv2) 9> ADD EXTRACT pt1e203, EXTTRAILSOURCE ./dirdat/sd
EXTRACT added.
```

```
GGSCI (apps-srv2) 11> ADD RMTTRAIL ./dirdat/td, EXTRACT pt1e203, MEGABYTES 128
RMTTRAIL added.

GGSCI (apps-srv2) 12>
```

Delivery Group

The delivery group is composed of the parameter file RT1E203.PRM, the Replicat process PT1E203, the local trail ./dirdat/sd and the checkpoint file ./dirchk/ET1E203.CPR for reporting. The Replicat process ES1E203 uses a checkpoint table, which is the default location for checkpoint. The next command creates the delivery group (Replicat).

```
GGSCI (apps-srv1) 3> EDIT PARAMS rt1e203
```

```
REPLICAT rt1e203
SETENV (ORACLE_SID=S1E2)
SETENV (NSL_LANG=AMERICAN_AMERICA.AL32UTF8)
USERID ggs_admin, PASSWORD oracle
ASSUMETARGETDEFS
DISCARDFILE ./dirrpt/rt1e203.dsc, PURGE
MAP osm$repapi.customers, TARGET osm$repapi.customers;
```

```
MAP osm$repapi.policies, TARGET osm$repapi.policies;
```

```
GGSCI (apps-srv1) 4> ADD REPLICAT rt1e203, EXTTRAIL ./dirdat/td
REPLICAT added.

GGSCI (apps-srv1) 5>
```

This concludes Oracle GoldenGate bi-directional configuration. Under normal business operation, the data flow is always from the primary system (source) to the live-standby system (target). Users are temporary migrated to the live-standby system either due to switchover or failover operation.

Switchover and Failover

Moving users from the primary to the standby site is performed as a planned (switchover) or unplanned (failover) situation. A switchover is for well-planned specific system maintenance tasks such as primary site database upgrade, standby operating system upgrade, etc. The following sections describe the process workflow for a primary site protected by one live standby site, which may need to perform either a switchover or failover.

A switchover consists of the following tasks:

- Switchover for planned primary to standby site
- Moving users back to the primary site

A failover consists of the following tasks:

- Failover for unplanned primary to standby site
- Re-establish the primary site
- Moving users back to the primary site

During the switchover and failover operations, the primary system is referred to by (P) and the live standby system is referred to by (S). This distinction avoids executing commands on the wrong Oracle GoldenGate instance or database.

Planned Primary to Live-Standby Switchover Workflow

Database and system administrators plan and perform system maintenance tasks. For systems operating 24/7, temporarily moving users from the primary system to the live-standby system enables business continuity by sustaining the minimum downtime while ensuring data integrity. Figure 5-8 depicts the steps to move users from the primary to the live-standby system.

The two common maintenance task benefits from a planned switchover are:

- Database Upgrades
- Operating System Upgrades

Due to less resource availability on the live-standby system, the switchover is temporary. Users are moved back to the primary system after successfully performing the maintenance tasks.

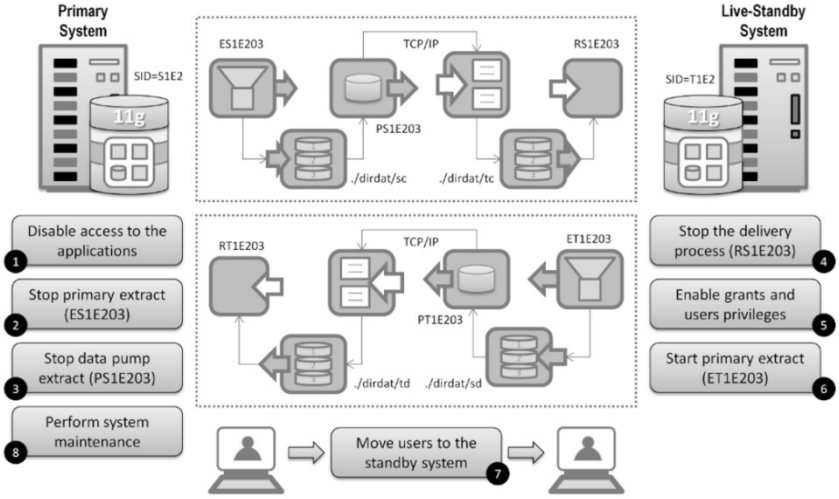


Figure 5-8: Planned Switchover from Primary to Standby System

Tasks performed on the primary system are referred to as (P), whereas tasks performed on the live-standby system referred to as (S).

Disable Access to the Applications (P)

For web-based applications accessing the database, shutdown of the application servers disables access to the database. To ascertain that no transactions were applied to the database during the switchover, after shutdown of the application servers, place the primary database on restricted sessions as shown below.

```
SQL> CONN / AS SYSDBA
Connected.
SQL> ALTER SYSTEM ENABLE RESTRICTED SESSION;

System altered.

SQL> GRANT RESTRICTED SESSION to ggs_admin;
```

```
Grant succeeded.

SQL> SELECT logins
  2  FROM   v$instance;

LOGINS
----------
RESTRICTED
```

The primary and secondary extracts are still running to complete capture and routing of the transaction after the shutdown of the application servers. After finishing Step 8 of Figure 5-8, revoke the restricted session privilege from the Oracle GoldenGate database user and disable the database restricted session.

```
SQL> REVOKE RESTRICTED SESSION FROM ggs_admin;

Revoke succeeded.

SQL> ALTER SYSTEM DISABLE RESTRICTED SESSION;

System altered.

SQL>
```

Stop the Primary Extract (P)

Before stopping the primary extract (ES1E203), issue the LAG EXTRACT command until it returns the message "At EOF, no more records to process." This indicates the primary extract has processed the transactions logs and is ready to issue the STOP EXTRACT command.

```
GGSCI (apps-srv1) 2> LAG EXTRACT es1e203

Sending GETLAG request to EXTRACT ES1E203 ...
Last record lag 1 seconds.
At EOF, no more records to process.

GGSCI (apps-srv1) 3> STOP EXTRACT es1e203

Sending STOP request to EXTRACT ES1E203 ...
Request processed.

GGSCI (apps-srv1) 4>
```

Stop the Data Pump Extract (P)

Run the LAG EXTRACT command until it returns "At EOF, no more records to process." This indicates the data pump has sent all transactions to the live-standby system. Now, issue the STOP EXTRACT command.

```
GGSCI (apps-srv1) 5> LAG EXTRACT ps1e203

Sending GETLAG request to EXTRACT PS1E203 ...
Last record lag 4 seconds.
At EOF, no more records to process.

GGSCI (apps-srv1) 6> STOP EXTRACT ps1e203

Sending STOP request to EXTRACT PS1E203 ...
Request processed.

GGSCI (apps-srv1) 7>
```

Stop the Delivery Process (S)

Run the command LAG REPLICAT until it returns the message "At EOF, no more records to process." This indicates that the delivery process has applied all of the transaction to the live-standby database. Now, issue the STOP REPLICAT command.

```
GGSCI (apps-srv2) 2> LAG REPLICAT rs1e203

Sending GETLAG request to REPLICAT RS1E203 ...
Last record lag 7 seconds.
At EOF, no more records to process.

GGSCI (apps-srv2) 3> STOP REPLICAT rs1e203

Sending STOP request to REPLICAT RS1E203 ...
Request processed.

GGSCI (apps-srv2) 4>
```

Enable Grants and Constraints (S)

This step is site specific. To minimize the downtime, the database DDL commands to grant and enable constraints that should be developed prior to initiating the switchover.

Start the Primary Extract (S)

Prior to moving users to the standby system, run the ALTER EXTRACT command to change the BEGIN timestamp. Then, start the primary extract only. Do not start the data pump extract. Ensure the storage level for the local trail is preserved to handle the primary system maintenance timeframe.

```
GGSCI (apps-srv2) 6> ALTER EXTRACT et1e203, BEGIN now
EXTRACT altered.

GGSCI (apps-srv2) 7> START EXTRACT et1e203

Sending START request to MANAGER ...
```

```
EXTRACT ET1E203 starting

GGSCI (apps-srv2) 8>
```

Since the manager process is configured to handle local trail space management when the trail files are shipped, they are automatically purged and keep disk space level optimized.

Switch User to the Live-Standby System

Run the necessary commands (scripts) to switchover the application servers to connect to the standby system. This enables the user to connect and perform transactions on the standby system.

Perform System Maintenance (P)

The time incurred to complete the system maintenance tasks impacts the storage level on the live-standby system. Performing the system maintenance on a separate test system minimizes the downtime of the primary system and reveals site specific issues.

The downtime for performing a planned switchover from the primary system to the live-standby system depends on Steps 5 & 7. These are the steps to run the database DDL commands to grant privileges, enable constraints, and start the application servers respectively. To successfully complete Steps 1 –7, it will take an estimated 10-30 minutes, when the commands are well-tested in the form script files.

Planned Standby to Primary Switchover Workflow

After completing the maintenance tasks on the primary system, a planned switchover from the live-standby to the primary system is performed. Figure 5-9 depicts the workflow steps to move users back to the primary system. Ascertain the current active database instance for a connected session using the following SQL statement:

```
SQL> CONN / AS SYSDBA
Connected.
SQL> COLUMN INSTANCE FORMAT A10
SQL> COLUMN DATABASE FORMAT A10
SQL> SELECT SYS_CONTEXT('USERENV', 'INSTANCE_NAME') INSTANCE,
  2         SYS_CONTEXT('USERENV', 'DB_NAME') DATABASE
  3  FROM   dual;

INSTANCE   DATABASE
---------- ----------
S1E2       S1E2

SQL>
```

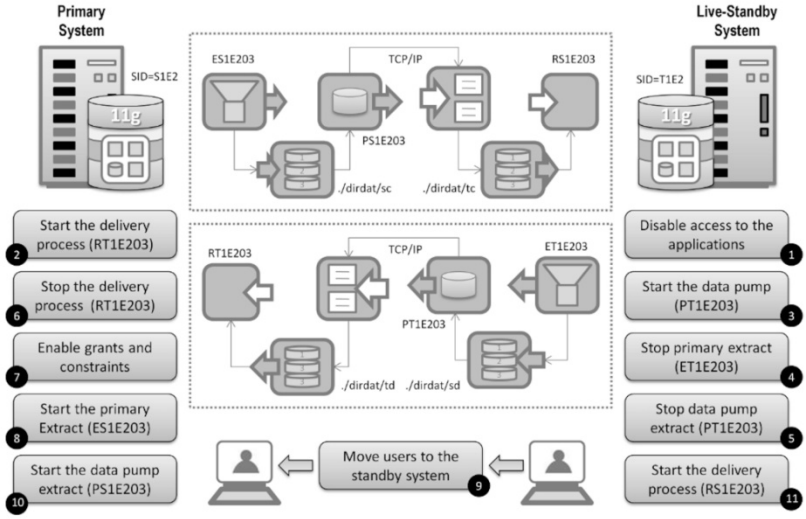


Figure 5-9: Planned Switchover from Live Standby to Primary System

Disable Application Access (S)

Start disabling users access to the applications by a shutdown of the application servers. The primary extract on the standby system remains running to capture the end of the transactions logs.

Start the Delivery Process (P)

To apply the transaction that occurred during the maintenance timeframe, start the delivery process (RT1E203).

```
GGSCI (apps-srv1) 9> START REPLICAT rt1e203

Sending START request to MANAGER ...
REPLICAT RT1E203 starting

GGSCI (apps-srv1) 10>
```

Start the Data Pump (S)

Since the data pump process stopped for the duration of the maintenance on the primary system, it needs to be started to send the transactions to the primary system.

```
GGSCI (apps-srv2) 9> START EXTRACT pt1e203
```

```
Sending START request to MANAGER ...
EXTRACT PT1E203 starting
```

Stop the Primary Extract (S)

While the applications are not running, keep issuing the command LAG EXTRACT until the message "At EOF, no more records to process" returns. This indicates all transaction logs are captured.

```
GGSCI (apps-srv2) 10> LAG EXTRACT et1e203

Sending GETLAG request to EXTRACT ET1E203 ...
No records yet processed.
At EOF, no more records to process.

GGSCI (apps-srv2) 11> STOP EXTRACT et1e203

Sending STOP request to EXTRACT ET1E203 ...
Request processed.

GGSCI (apps-srv2) 12>
```

Stop the Data Pump Extract (S)

Repeatedly issue the command LAG EXTRACT until it returns the message "At EOF, no more records to process." This indicates all transaction have been sent to the primary system. Then, issue the command STOP EXTRACT.

```
GGSCI (apps-srv2) 12> LAG EXTRACT pt1e203

Sending GETLAG request to EXTRACT PT1E203 ...
No records yet processed.
At EOF, no more records to process.


GGSCI (apps-srv2) 13> STOP EXTRACT pt1e203

Sending STOP request to EXTRACT PT1E203 ...
Request processed.

GGSCI (apps-srv2) 14>
```

Stop the Delivery Process (P)

The next step is to make sure the data processed during the maintenance is applied to the primary database. Keep issuing the command LAG REPLICAT until it returns "At EOF, no more records to process." This indicates all transactions have been applied to the primary database.

```
GGSCI (apps-srv1) 10> LAG REPLICAT rt1e203

Sending GETLAG request to REPLICAT RT1E203 ...
No records yet processed.
At EOF, no more records to process.

GGSCI (apps-srv1) 11> STOP REPLICAT rt1e203

Sending STOP request to REPLICAT RT1E203 ...
Request processed.

GGSCI (apps-srv1) 12>
```

Enable Grants and Constraints (P)

The primary system is now ready for switchover. Before doing so, run the database DDL command to grant privileges and enable constraints.

Start the Primary Extract (P)

Start the primary extract to capture from the transaction logs.

```
GGSCI (apps-srv1) 12> START EXTRACT es1e203

Sending START request to MANAGER ...
EXTRACT ES1E203 starting
```

Switch User Activity

Starting the application servers allows users to connect to the primary system. Now users are connected back to the primary system. Trails created need to be shipped to the live standby system.

Start the Data Pump (P)

Start the data pump to begin sending data to the live-standby system. In the case of data pump process failure due to connection rejection by the target system, verify the hostname or IP address is reachable by performing PING from within GGSCI command-line and the manager listening port on the live-standby system is correct.

```
GGSCI (apps-srv1) 13> START EXTRACT ps1e203

Sending START request to MANAGER ...
EXTRACT PS1E203 starting

GGSCI (apps-srv1) 14>
```

Start the Delivery Process (S)

The delivery process (RS1E203) is started to keep applying transactions to the live standby system.

```
GGSCI (apps-srv2) 14> START REPLICAT rs1e203

Sending START request to MANAGER ...
REPLICAT RS1E203 starting

GGSCI (apps-srv2) 15>
```

On the primary system, verify the status of the primary and secondary extracts. This is where users are currently executing transactions. On the live standby system, verify the status of the delivery process. Using tools such Oracle GoldenGate Veridata to compare the primary and live-standby databases allows database administrators to discover and fix missing rows, proactively. Another effective administration tool is the Oracle GoldenGate 11g Monitor, which is used for monitor health checks and diagnostics.

Unplanned Primary to Standby Failover Workflow

Situations such as a primary system disk failure are unplanned situations where users are moved to the live standby system until the primary system is recovered. Figure 5-10 depicts the steps to move users from the primary to the live-standby system. The unplanned failover from the primary to live-standby system presumes the live-standby Oracle GoldenGate configuration is in-place, and the database scripts to enable constraints are available.

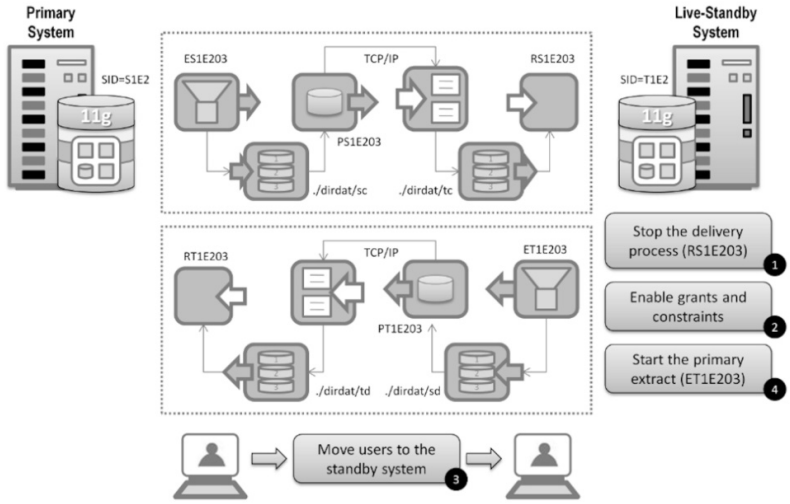


Figure 5-10: Primary to Standby Failover

Stop the Delivery Process (S)

Ensure transactions have been applied to the live standby system by continuing to issue the command LAG REPLICAT, until it returns "At EOF, no more records to process." This indicates all transactions have been applied to the live standby database.

```
GGSCI (apps-srv2) 1> LAG REPLICAT rs1e203

Sending GETLAG request to REPLICAT RS1E203 ...
Last record lag 8 seconds.
At EOF, no more records to process.

GGSCI (apps-srv2) 2> STOP REPLICAT rs1e203

Sending STOP request to REPLICAT RS1E203 ...
Request processed.

GGSCI (apps-srv2) 3>
```

Enable Grants and Constraints (S)

Since grants and constraints remain disabled during normal operation, before moving users, run the database DDL commands to grant database privileges and enable constraints.

Start the Primary Extract (S)

By starting the primary extract only, it continuously captures the transactions to the local trail, so that it is sent by the data pump when moving users back to the primary system.

```
GGSCI (apps-srv2) 4> START EXTRACT et1e203

Sending START request to MANAGER ...
EXTRACT ET1E203 starting

GGSCI (apps-srv2) 5>
```

Switch Users to Primary System

Redirect the application servers to the standby system. Since the primary system is not available until it is fixed, ensure the disk space is monitored to preserve the size of the local trail because the data pump is not available.

Unplanned Standby to Primary Failover Workflow

In the event of an unplanned failover, users are quickly moved to the standby system where business continues to operate. The downtime associated with an unplanned failover is minimal and mostly proportional to the unapplied amount of the remote trail files currently being applied to the live-standby system.

The next step is to determine the cause of the primary system failure and re-instantiate it from the live-standby system in preparation to move users back to the primary system. Figure 5-11 depicts the workflow for an unplanned live-standby to primary system failover.

The amount of downtime is greatly minimized by performing an initial load from the active live-standby system, then applying the transactions that occurred during the initial load timeframe to the primary system database. When Step 4 is completed, users are unable to execute transactions until Step 12 is completed (which enables the primary system database grants and constraints). Step 16 includes starting the application servers and moving users back to the primary system. The amount of downtime varies and depends on the size of the live-standby system local trail.

Task preformed on the primary system referred to by (P), whereas task performed on the live-standby system referred to by (S).

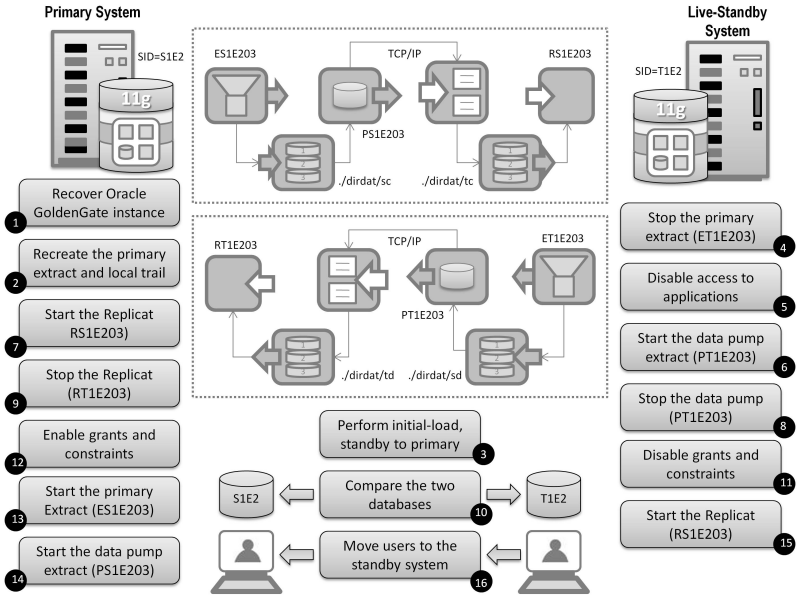


Figure 5-11: Unplanned Standby to Primary System Failover

Recover Oracle GoldenGate Instance (P)

Recovering Oracle GoldenGate instance may require installing the software, preferably to the same location to avoid modifying the instance parameter files. Then, restore the parameter files from the backup and start the manager process, using the same TCP/IP port to avoid modifying the manager and data pump parameter files. Verify the manager is running on the same port.

```
GGSCI (apps-srv1) 19> START mgr
Manager started.

GGSCI (apps-srv1) 20> INFO mgr

Manager is running (IP port apps-srv1.7805, Process ID 11252).

GGSCI (apps-srv1) 21>
```

Recreate the Primary Extract (P)

Regardless of the type of system failure, deleting and creating the primary extract is necessary. This avoids any type of reference to the old primary group.

```
GGSCI (apps-srv1) 13> DELETE EXTTRAIL ./dirdat/sc
Deleting extract trail ./dirdat/sc for extract ES1E203

GGSCI (apps-srv1) 14> DELETE EXTRACT es1e203
Deleted EXTRACT ES1E203.


GGSCI (apps-srv1) 15>
```

```
GGSCI (apps-srv1) 15> ADD EXTRACT es1e203, TRANLOG, BEGIN now, THREADS 1
EXTRACT added.

GGSCI (apps-srv1) 16> ADD EXTTRAIL ./dirdat/sc, EXTRACT es1e203, MEGABYTES 128
EXTTRAIL added.

GGSCI (apps-srv1) 17>
```

Perform Initial Load

There are several techniques to establish the primary database from the live-standby database system. Refer to Chapter 3 for details regarding the available techniques. Oracle GoldenGate direct load technique is illustrated below.

Before performing initial load to establish the primary system database using the direct load method, the database table structure must be prepared. Several techniques are available including the Oracle Database Data Pump Utility.

For direct initial load method, start by preparing the initial load delivery process – as it will be referenced by the initial load extract. The objective is to perform initial load from an active live-standby system database to minimize the downtime.

Create the initial load Replicat using the SPECIALRUN option.

```
GGSCI (apps-srv1) 21> EDIT PARAMS rload03
```

```
REPLICAT rload03
SPECIALRUN
ASSUMETARGETDEFS
HANDLECOLLISIONS
USERID ggs_admin, PASSWORD oracle
EXTFILE ./dirdat/initload03.dat
DISCARDFILE ./dirrpt/rload03.dsc, PURGE
MAP osm$repapi.*, TARGET osm$repapi.*;
END RUNTIME
```

```
GGSCI (apps-srv1) 22> ADD REPLICAT rload03, SPECIALRUN
REPLICAT added.

GGSCI (apps-srv1) 23>
```

Configure and create the initial load extract, ELOAD03. This extract reads from the active standby database and performs direct inserts using an initial load delivery process.

```
GGSCI (apps-srv2) 6> EDIT PARAMS eload03
```

```
EXTRACT eload03
USERID ggs_admin, PASSWORD oracle
RMTHOST apps-srv1, MGRPORT 7805
RMTTASK REPLICAT, GROUP rload03
TABLE osm$repapi.policy_types;
TABLE osm$repapi.discount_types;
TABLE osm$repapi.policies;
TABLE osm$repapi.customers;
```

```
GGSCI (apps-srv2) 7> ADD EXTRACT eload03, SOURCEISTABLE
EXTRACT added.

GGSCI (apps-srv2) 8>
```

The primary system database is established by starting the initial load extract of the live standby system.

```
GGSCI (apps-srv2) 16> START EXTRACT eload03

Sending START request to MANAGER ...
EXTRACT ELOAD03 starting
```




```
GGSCI (apps-srv2) 17>
```

The following shows part of the report to verify the successful completion of the initial load.

```
GGSCI (apps-srv1) 5> VIEW REPORT rload03

***********************************************************************
*                   ** Run Time Statistics **                         *
***********************************************************************


Report at 2014-01-24 17:28:40 (activity since 2014-01-24 17:27:48)

From Table OSM$REPAPI.POLICY_TYPES to OSM$REPAPI.POLICY_TYPES:
       #                   inserts:         4
       #                   updates:         0
       #                   deletes:         0
       #                  discards:         0
From Table OSM$REPAPI.DISCOUNT_TYPES to OSM$REPAPI.DISCOUNT_TYPES:
       #                   inserts:         3
       #                   updates:         0
       #                   deletes:         0
       #                  discards:         0
From Table OSM$REPAPI.POLICIES to OSM$REPAPI.POLICIES:
       #                   inserts:    102100
       #                   updates:         0
       #                   deletes:         0
       #                  discards:         0
From Table OSM$REPAPI.CUSTOMERS to OSM$REPAPI.CUSTOMERS:
       #                   inserts:    102100
       #                   updates:         0
       #                   deletes:         0
       #                  discards:         0
```

Disable Application Access (S)

Stop the applications. This is "application server dependent."

Stop the Primary Extract (S)

Stop the primary extract in preparation to applying the changes that occurred during the initial load timeframe. Issue the LAG EXTRACT before stopping the extract, this ensures all transactions have been processed by the extract.

```
GGSCI (apps-srv2) 19> LAG EXTRACT et1e203

Sending GETLAG request to EXTRACT ET1E203 ...
Last record lag 2 seconds.
At EOF, no more records to process.
```




```
GGSCI (apps-srv2) 20> STOP EXTRACT et1e203

Sending STOP request to EXTRACT ET1E203 ...
Request processed.

GGSCI (apps-srv2) 21>
```

Start the Data Pump (S)

The data pump is started to send data to the primary system. These are the changes that occurred during the initial load.

```
GGSCI (apps-srv2) 21> START EXTRACT pt1e203

Sending START request to MANAGER ...
EXTRACT PT1E203 starting

GGSCI (apps-srv2) 22>
```

Start the Delivery Process (P)

Apply the changes to the primary database. Ensure handling collision to avoid a unique constraint error.

```
GGSCI (apps-srv1) 7> START REPLICAT rt1e203

Sending START request to MANAGER ...
REPLICAT RT1E203 starting

GGSCI (apps-srv1) 8>
```

Stop the Data Pump (S)

Stop the data pump on the standby system. Keep issuing the command LAG EXTRACT until it returns "At EOF, no more records to process." This indicates all transactions have been processed.

```
GGSCI (apps-srv2) 23> LAG EXTRACT pt1e203

Sending GETLAG request to EXTRACT PT1E203 ...
Last record lag 621,259 seconds.
At EOF, no more records to process.

GGSCI (apps-srv2) 24> STOP EXTRACT pt1e203

Sending STOP request to EXTRACT PT1E203 ...
Request processed.
```

```
GGSCI (apps-srv2) 25>
```

Stop the Delivery Process (P)

Stop the delivery process on the primary system. Keep issuing the command LAG REPLICAT until it returns "At EOF, no more records to process." This indicates all transactions have been applied to the primary database.

```
GGSCI (apps-srv1) 9> LAG REPLICAT rt1e203

Sending GETLAG request to REPLICAT RT1E203 ...
Last record lag 621,594 seconds.
At EOF, no more records to process.

GGSCI (apps-srv1) 10> STOP REPLICAT rt1e203

Sending STOP request to REPLICAT RT1E203 ...
Request processed.

GGSCI (apps-srv1) 11>
```

Compare Primary and Standby Databases

For data consistency, prior to proceeding, compare the primary and standby databases. Use tools such as Oracle GoldenGate Veridata to discover the missing rows or SQL*Plus to compare table rows count. For now, compare the two databases using rows count for tables CUSTOMERS and POLICIES using SQL*Plus for the tables.

```
SQL> conn osm$repapi/oracle
Connected.
SQL> select count(*) from customers;

  COUNT(*)
----------
    102100

SQL>
```

Execute the statement for the databases S1E2 and T1E2. Refer to Chapter 6 to learn more about Oracle GoldenGate Veridata.

Disable Grants and Constraints (S)

The receiving system must have grants and constraints disabled. This avoids referential integrity violations.

Enable Grants and Constraints (P)

Enable the database grants and constraints on the primary system.

Start the Primary Extract (P)

Start the primary extract. Since the primary was recreated, it will start transaction logs capture from the creation timestamp.

```
GGSCI (apps-srv1) 13> START EXTRACT es1e203

Sending START request to MANAGER ...
EXTRACT ES1E203 starting

GGSCI (apps-srv1) 14>
```

Start the Data Pump (P)

Start the data pump. This starts sending data to the standby system.

```
GGSCI (apps-srv1) 15> START EXTRACT ps1e203

Sending START request to MANAGER ...
EXTRACT PS1E203 starting

GGSCI (apps-srv1) 16>
```

Start the Delivery Process (S)

Start the delivery process to apply transactions sent to the standby system.

```
GGSCI (apps-srv2) 25> START REPLICAT rs1e203

Sending START request to MANAGER ...
REPLICAT RS1E203 starting

GGSCI (apps-srv2) 26>
```

Move Users to the Primary System

Now, confirm the Replicat has applied all the transactions to the standby system database, start the application servers to move users back to the primary system. This step is application server/site specific. For example, if it's PL/SQL portal application such as Oracle APEX, it's changing the database connect string only from the live-standby system to the primary system database.

High Availability

Oracle High Availability has been always associated with Oracle Real Application Clusters (RAC), where within the same data center, multiple tightly coupled servers (nodes) access the same database. Another advanced architecture of RAC is Extended RAC, where nodes are loosely coupled and spread across two or more data centers. Figure 5-12 compares the high availability of Oracle GoldenGate and Extended RAC.

The upper part of Figure 5-12 is Oracle GoldenGate configured using an active-active bi-directional topology model. Over the TCP/IP network, data flows from the site A (node-1) to site B (node-2) and vice versa. The self-evident advantage of this solution is the unlimited distance between node-1 and node-2. The drawback is the chance of scalability and data conflict handling. However, because the data is delivered in canonical format for a well-developed application, node-1 and node-2 may use different platforms. For example node-1 is Oracle Database 11g on Oracle Solaris while node-2 is Oracle Database 11g on Oracle Linux. Avoid heterogeneous database architecture for a high availability solution, unless the applications developed to support all database types transparently.

The lower part of Figure 5-12 shows Oracle Extended RAC. The databases are mirrored using dense wavelength division multiplexing (DWDM) over a dark fiber link. This enables always in-sync copies of the database. The self-evident advantage is out-of-box scalability and high availability. The drawback is the distance limitation between site A and site B for up to 100 kilometers and cost-factor.

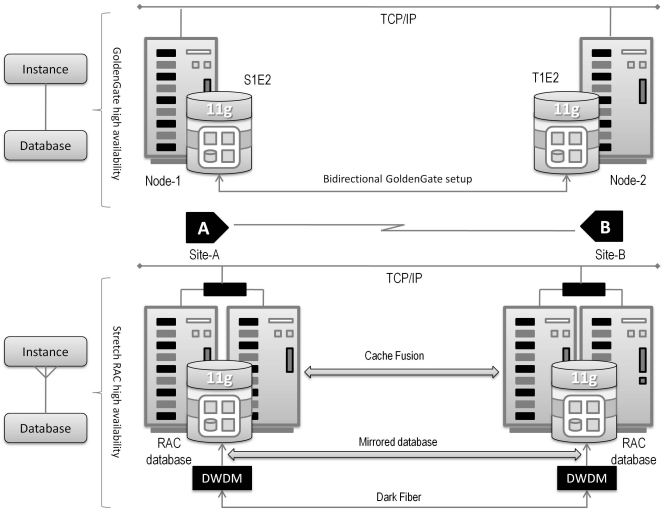


Figure 5-12: Comparison of 2-Node Oracle GoldenGate and Extended RAC

When using Oracle GoldenGate to implement a high availability platform, it is configured as an active-active bi-directional topology model. The source system (node-1) and target system (node-2) are available for user connections. Users may explicitly connect to either system based on predetermined attributes such as location, range of TCP/IP address, built-in application context, etc.

Implicit connections will be based on database and/or system workload. Users requests are routed to the system with minimal workload. This provides connection time workload evaluation to distribute database sessions across both systems. Figure 5-12 shows an implicit form of handling users connections. The database load balancer evaluates the preferred database, which is the database with lesser workload. Then, send the request to the application server which has its own HTTP load balancer. For more details regarding a high availability load balancer, refer to Figure 5-2.

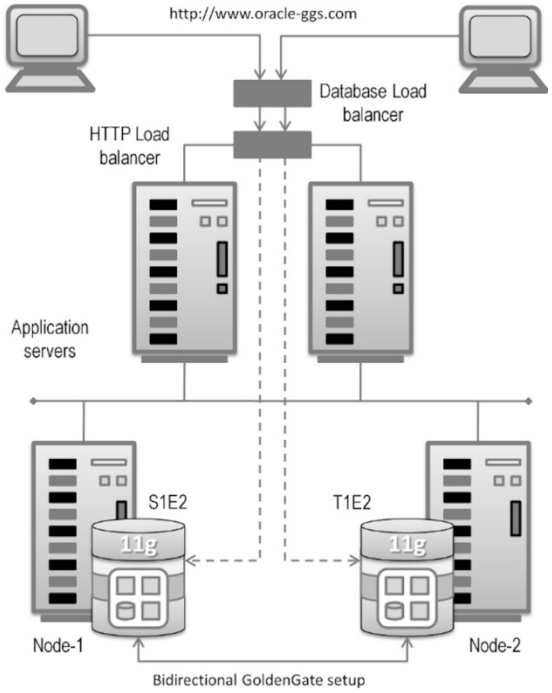


Figure 5-13: Load Balancing on Bi-Directional Topology

Applications running on Windows such as Oracle for .NET employ Oracle SQLNET to connect to the database. Whereas Java desktop applications employs JDBC drivers to connect to the database. For this type of applications, HTTP Servers are eliminated, the client applications establish stateful types of database connections.

Bi-Directional Replication Considerations

Since data manipulation language (DML) statements may occurs at 'almost' the same time from either direction, the below must be considered and configured on a bi-directional active-active Oracle GoldenGate configuration.

Endless Loop Detection

In an active-active bi-directional topology, because the Extract and the Replicat are operating on the same table, the Replicat must be prevented from sending the same applied record back to the source to avoid an endless loop situation.

There are two techniques for loop detection to avoid re-extracting and re-replicating the same record:

- Replicat-Extract trace table
- Replicat-Extract parameter

Replicat-Extract Trace Table

When the Replicat applies transactions to the target database, it writes to a trace table which the Extract on the same system uses for loop detection by eliminating the Replicat transactions. When the Replicat starts, if the parameter TRACETABLE is configured then it uses the specified trace table. If the parameter TRACETABLE is not configured, it tries the default trace table name GGS_TABLE owned by the Oracle GoldenGate user of the DBLOGIN parameter. The following commands illustrate a non-default trace table GGS_T1E2_TRC is created and configured as follows:

- Create the trace table GGS_T1E2_TRC. This requires connecting to the database from within Oracle GoldenGate and issues the two GGSCI commands below:

```
GGSCI (apps-srv2) 1> DBLOGIN USERID ggs_admin, PASSWORD oracle
Successfully logged into database.

GGSCI (apps-srv2) 2> ADD TRACETABLE ggs_admin.ggs_t1e2_trc

Successfully created trace table GGS_ADMIN.GGS_T1E2_TRC.

GGSCI (apps-srv2) 3>
```

- Add the parameter TRACETABLE to the Extract and the Replicat parameter file. The GGS_ADMIN is the Oracle GoldenGate database schema.

```
TRACETABLE ggs_admin.GGS_T1E2_TRC
```

Replicat Extract Parameter

The second method uses the Replicat TRANLOGOPTIONS parameter only. This methods eliminates the processing overhead associated with using the database trace table explained above. The option EXCLUDEUSER or EXCLUDEUSERID of the parameter TRANLOGOPTIONS explicitly avoid the re-capture of the transactions applied by specified user. This requires using a dedicated Oracle GoldenGate user so that transactions applied by this user are eliminated – avoiding re-sending the transaction back to the originated system. The TRANLOGOPTIONS is shown next.

```
TRANLOGOPTIONS EXCLUDEUSER ggs_admin
```

Query DBA_USERS to replace the username with the user ID.

```
SQL> SELECT username, user_id
  2  FROM   dba_users
  3  WHERE  username='GGS_ADMIN';

USERNAME                          USER_ID
------------------------------ ----------
GGS_ADMIN                            1101

SQL>
```

Data Conflicts

This issue is related to data integrity that must be addressed comprehensively in an active-active bi-directional configuration. There are several types of conflict, avoidance, and detection techniques. Because applications are mostly not replication-ware, resolving data conflict is required and should be governed by the business rules. Oracle GoldenGate best practices recommend resolving data conflicts in the following order:

- Apply avoidance techniques of data segmentation and application partitioning
- Adapt Oracle GoldenGate automatic conflict detection and resolution (CDR)
- Develop custom conflict detection and resolution routines

Details regarding data conflicts and detection techniques are described in this chapter.

Truncate Table

Since the TRUNCATE TABLE command is not supported in an active-active bi-directional configuration, at any point in time, Extracts and Replicats must be configured for one-and-only-one dedicated system to issue the TRUNCATE TABLE database command. This is achieved by adding the parameter GETTRUNCATES

on the allowed system and the parameter IGNORETRUNCATES on the disallowed system. To ensure the TRUNCATE TABLE command is issued from the allowed database only, revoke the TRUNCATE TABLE privilege from the application roles.

Database Sequences

When using database sequence numbers for tables primary and unique keys values on an active-active bi-directional configuration, avoid primary and unique key uniqueness conflicts. There are several techniques for implementing database sequences on an active-active bi-directional configuration. Common techniques for using database sequences are:

- Node-1 database takes odd numbers, while node-2 database takes even numbers
- Node-1 database uses a different range of numbers than node-2 database
- Implement database sequence values based on built-in application contexts

Details regarding database sequence implementations are described in this chapter.

Real-Time Disconnected Transactions

When the applications retrieve and manipulate data synchronously from either of the two databases, database links are created which enable the application to fetch and to manipulate data from either databases. There is no data conflict. This is known as distributed database applications and adhere to database dependency, locks, and resource management as illustrated below.

Create two database links from S1E2 to T1E2 and from T1E2 to S1E2. Before creating the database link, ensure the database connect string is able to establish remote connection.

Create the database link from S1E2 to T1E2.

```
SQL> CREATE PUBLIC DATABASE LINK t1e2.precisetrace.com
  2  CONNECT TO osm$repapi IDENTIFIED BY oracle
  3  USING 'T1E2';

Database link created.

SQL>
```

Create the database link from T1E2 to S1E2.

```
SQL> CREATE PUBLIC DATABASE LINK s1e2.precisetrace.com
  2  CONNECT TO osm$repapi IDENTIFIED BY oracle
  3  USING 'S1E2';
```

```
Database link created.

SQL>
```

The following update statement executed remotely on S1E2 over the database link. Do not issue COMMIT statement to continue locking resources.

```
SQL> host hostname
apps-srv2

SQL> CONN osm$repapi/oracle
Connected.
SQL> UPDATE customers@s1e2 SET
  2     cust_email='falswaimil@gmai.com'
  3  WHERE cust_no=100100;

1 row updated.

SQL>
```

The following update statement waits for the above remote transaction to release resources, issuing COMMIT or ROLLBACK, then execute the update.

```
SQL> host hostname
apps-srv1

SQL> CONN osm$repapi/oracle
Connected.
SQL> UPDATE customers SET
  2     cust_email='falswaimil@yahoo.com'
  3  WHERE cust_no=100100;

SQL>
```

This illustrates how synchronous transactions used by distributed applications do not trigger data conflict. They adhere to the target database resource management, locks, and queue handling.

However, for near real-time 'disconnected' transactions that are running asynchronously, the need to consider and handle data conflict caused by executing the applications is very obvious. Conflicts are either avoided using data and applications segmentation or handled using Oracle GoldenGate built-in and customs conflict detection and resolution. The section 'conflicts resolution' describes the built-in and advance custom techniques for handling data conflict.

Applications and Database Design for High Availability

The key to avoid data conflict in an asynchronous environment is to partition data and application dynamically, which is referred to as segmentation. This guarantees

an application's group interacts and manipulates a logically partitioned subset of the database that is isolated from others application's groups. This is the fundamental concept of virtual private database (VPD). Figure 5-14 shows two application's groups in an active-active bi-directional topology model that are designed to be virtually isolated.

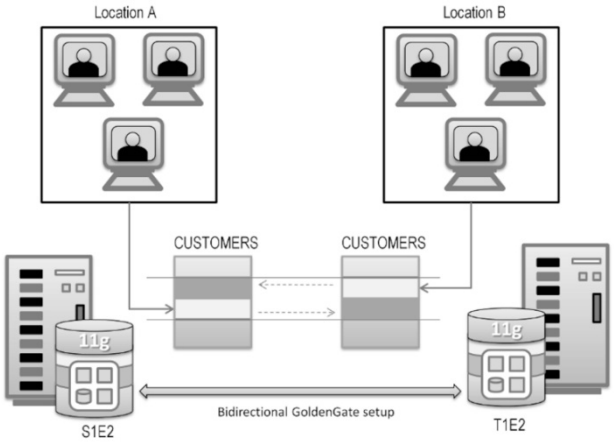


Figure 5-14: Conflict Avoidance Using Data and Applications Segmentation

The above data conflict avoidance technique is either implemented at the applications level or the database. Best practices recommend using the Oracle Database Enterprise Edition Virtual Private Database (VPD) feature. VPD uses policy functions for dynamically modifying SQL Statements to enforce data segmentation through data access control based on application contexts. Virtual Private Database implementation is explained in more detail in the section 'Virtual Private Database.'

An valuable feature for implementing segmentation by location is performance. Users are located in close proximity to access data. For example, a call center with two locations, where each location handles sales for specific products list avoids data conflict and has local data access which eliminates the overhead associated with network failure. Users continue to enter transactions locally. When the network becomes available, Oracle GoldenGate starts routing data to a remote system.

Bi-Directional Extract and Delivery Groups

The following Oracle GoldenGate configuration is built on the components used by the disaster recovery described above. Since Oracle GoldenGate high availability is deployed as an active-active bi-directional topology architecture, it is recommended to divide it into two phases:

Phase I: Working Bi-Directional Topology Architecture – An end-to-end optimized and well-tested bi-directional replication capable of handling the transactions volume for near real-time data movement. Data conflict must be addressed as it's an unavoidable.

Phase II: Comprehensive Conflicts Handling – Implementing avoidance and detection techniques for handling data conflicts. Since Oracle GoldenGate operates asynchronously, near real-time data movement minimizes conflict occurrences.

Figure 5-15 outlines the groups and trails. The upper part is for node-1 to node-2 data flow and the lower part is the opposite configuration, from node-2 to node-1.

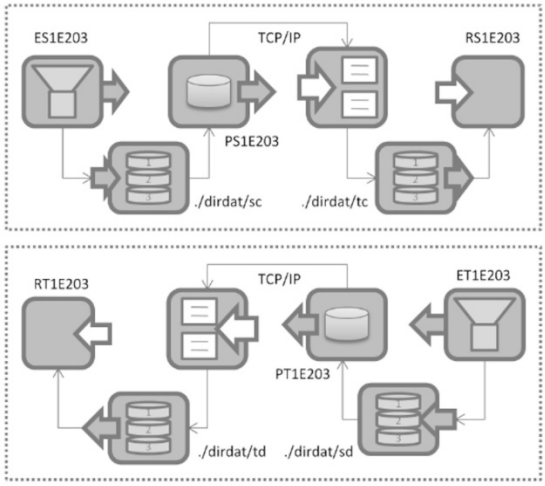


Figure 5-15: Bi-Directional High Availability Topology

Node-1 Parameter Files, Extracts, and Replicats Processes

Node-1 Oracle GoldenGate instance operates the processes listed in Table 5-2. Since an end-to-end data integrity among referenced tables must be enforced, designing multiple capture-and-delivery groups requires understanding the data model, our illustration is based on a single process architecture.

Component	Description
ES1E203	Primary capture process from the primary system transactions logs
./dirdat/sc	Local trail
PS1E203	Secondary extract data pump process send data to node-2 system
./dirdat/tc	Remote trail
RT1E203	Replicat process for node-2 database

Table 5-2: Node-1 Components

The parameter files must be modified to support bi-directional replication for conflict handling. Re-start the processes to activate the newly added parameters.

Node-1 primary extract process ES1E203 is shown below. Similarly modify Node-2 primary extract.

```
EXTRACT es1e203
SETENV  (ORACLE_SID=S1E2)
SETENV  (ORACLE_HOME="/d01/app/oracleSE")
SETENV  (NLS_LANG=AMERICAN_AMERICA.AR8ISO8859P6)
USERID  ggs_admin, password oracle
EXTTRAIL ./dirdat/sc
TABLE osm$repapi.CUSTOMERS;
TABLE osm$repapi.POLICIES
```

Node-2 delivery process RS1E203 is shown below. Similarly modify node-1 delivery process.

```
REPLICAT rt1e203
SETENV (ORACLE_SID=S1E2)
SETENV (NSL_LANG=AMERICAN_AMERICA.AL32UTF8)
USERID ggs_admin, PASSWORD oracle
ASSUMETARGETDEFS
DISCARDFILE ./dirrpt/rt1e203.dsc, PURGE
MAP osm$repapi.customers, TARGET osm$repapi.customers;
MAP osm$repapi.policies, TARGET osm$repapi.policies;
```

Verify the status of the processes on node-1 Oracle GoldenGate instance. The status of RUNNING indicates the Extract, the data pump, and the Replicat processes are configured correctly, at least.

The current status of the primary extract, ES1E203.

```
GGSCI (apps-srv1) 4> INFO EXTRACT es1e203

EXTRACT    ES1E203   Last Started 2014-01-28 22:23   Status RUNNING
Checkpoint Lag       03:42:54 (updated 00:00:05 ago)
Process ID           6378
Log Read Checkpoint  Oracle Redo Logs
                     2014-01-28 18:41:00  Thread 1, Seqno 30, RBA 39536
                     SCN 0.1997389 (1997389)
```

The local trail details, ./dirdat/sc.

```
GGSCI (apps-srv1) 5> INFO EXTTRAIL ./dirdat/sc

       Extract Trail: ./dirdat/sc
             Extract: ES1E203
               Seqno: 10
                 RBA: 1601
```




```
           File Size: 128M
```

The current status of the secondary (data pump) extract, PES1E203.

```
GGSCI (apps-srv1) 8> INFO EXTRACT ps1e203

EXTRACT    PS1E203   Last Started 2014-01-28 22:25   Status RUNNING
Checkpoint Lag       100:45:36 (updated 00:00:01 ago)
Process ID           6410
Log Read Checkpoint  File ./dirdat/sc000009
                     2014-01-24 17:40:22.000000  RBA 44546492
```

The current status delivery process from standby to primary system, RT1E203.

```
GGSCI (apps-srv1) 10> INFO REPLICAT rt1e203

REPLICAT   RT1E203   Last Started 2014-01-28 22:27   Status RUNNING
Checkpoint Lag       00:00:00 (updated 00:00:08 ago)
Process ID           6428
Log Read Checkpoint  File ./dirdat/td000008
                     First Record  RBA 45282

GGSCI (apps-srv1) 11>
```

Node-2 Parameter Files, Extracts, and Replicats Processes

Node-2 Oracle GoldenGate instance operates the processes listed in Table 5-3. When Oracle GoldenGate is configured for high availability, node-1 and node-2 systems should have similar resources in terms of CPU speed and RAM size. This allows similar workload handling across the two nodes.

Component	Description
ET1E203	Primary Extract process from node-2 transactions logs
./dirdat/sd	Local trail
PT1E203	Secondary extract data pump process from node-2 to node-1
./dirdat/td	Remote trail
RS1E203	Replicat process on node-1

Table 5-3: Node-2 Components

Verify the status of the processes on node-2 Oracle GoldenGate instance.

```
GGSCI (apps-srv2) 4> INFO EXTRACT et1e203

EXTRACT    ET1E203   Last Started 2014-01-28 22:31   Status RUNNING
Checkpoint Lag       100:58:00 (updated 00:00:07 ago)
Process ID           17047
Log Read Checkpoint  Oracle Redo Logs
                     2014-01-24 17:33:26  Thread 1, Seqno 25, RBA 42078224
                     SCN 0.1889870 (1889870)
```

```
GGSCI (apps-srv2) 6> INFO EXTRACT pt1e203

EXTRACT    PT1E203   Last Started 2014-01-28 22:32   Status RUNNING
Checkpoint Lag       273:31:53 (updated 00:00:04 ago)
Process ID           17083
Log Read Checkpoint  File ./dirdat/sd000005
                     2014-01-17 13:00:31.000000  RBA 45235

GGSCI (apps-srv2) 8> INFO REPLICAT rs1e203

REPLICAT   RS1E203   Last Started 2014-01-28 22:32   Status RUNNING
Checkpoint Lag       00:00:29 (updated 00:00:00 ago)
Process ID           17097
Log Read Checkpoint  File ./dirdat/tc000016
                     2014-01-28 22:32:32.880946  RBA 1686454

GGSCI (apps-srv2) 9>
```

Oracle Sequence for Artificial Primary Key Values

Oracle database sequence is a memory-based object. Hence, it is an efficient technique for generating primary key values. For a non-replicated database, sequences are suitable for generating primary key values without major issues. However, when it is used in a database synchronized by Oracle GoldenGate, it should be created to avoid primary key conflict for SQL insert statements. The following illustrates techniques for using Oracle sequences in databases synchronized in real-time by Oracle GoldenGate.

Odd-Even Numbers Distribution

The node-1 database fetches the odd number values only, while node-2 database fetches the even number values. Figure 5-16 depicts the sequence design created below:

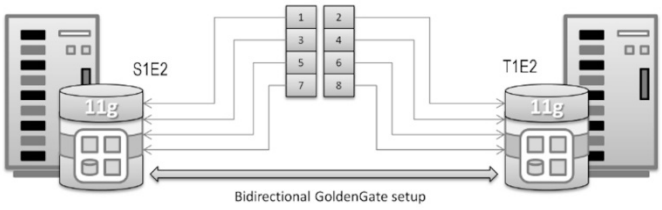


Figure 5-16: Odd-Even Numbers Sequencing

Node-1 database sequence is created to start with 1 and fetches odd numbers.

```
SQL> CREATE SEQUENCE cust_no_seq
  2  START WITH 1
  3  INCREMENT BY 2;

Sequence created.
```

```
SQL> SELECT cust_no_seq.nextval FROM dual;

   NEXTVAL
----------
         1

SQL> SELECT cust_no_seq.nextval FROM dual;

   NEXTVAL
----------
         3

SQL>
```

Node-2 database sequence is created to start with 2 and fetches even numbers.

```
SQL> CREATE SEQUENCE cust_no_seq
  2  START WITH 2
  3  INCREMENT BY 2;

Sequence created.

SQL> SELECT cust_no_seq.nextval FROM dual;

   NEXTVAL
----------
         2

SQL> SELECT cust_no_seq.nextval FROM dual;

   NEXTVAL
----------
         4

SQL>
```

Range Numbers Distribution

Node-1 database fetches from one range while node-2 database fetches from a second range. Figure 5-17 depicts the sequence design created below:

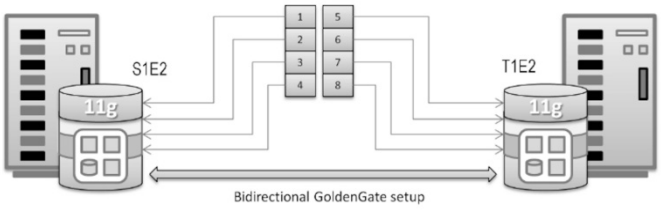


Figure 5-17: Range Numbers Sequencing

Node-1 database begins from 1 to n-1, where n is node-2 and begins sequence number.

```
SQL> CREATE SEQUENCE cust_no_seq
  2  START WITH 1;

Sequence created.
```

Node-2 begins at n, where n=5.

```
SQL> CREATE SEQUENCE cust_no_seq
  2  START WITH 5;

Sequence created.

SQL>
```

Application Context of Numbers Distribution

The following illustrates the use of sequence to generate unique values based on the application context. Figure 5-18 depicts the application context value generator for a two-node architecture.

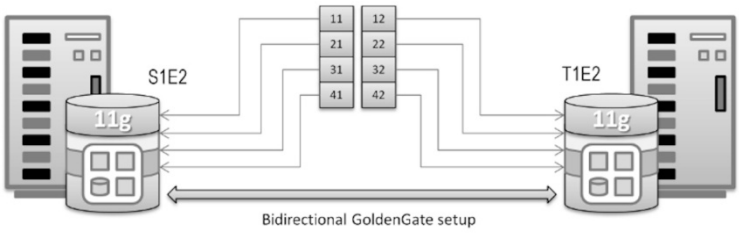


Figure 5-18: Numbers Sequencing based on Application Context

```
SQL> CREATE SEQUENCE cust_no_seq
  2  START WITH 1;

Sequence created.

SQL>
```

Use the database function SYS_CONTEXT to concatenate the sequence number with the instance number (INSTANCE) for the session. This technique is similar to a sequence created with INCREMENT BY 10.

```
SQL> CREATE SEQUENCE cust_no_seq;

Sequence created.

SQL> SELECT TO_NUMBER(cust_no_seq.nextval||SYS_CONTEXT('USERENV', 'INSTANCE'))
  2  FROM   dual;
```

```
TO_NUMBER(CUST_NO_SEQ.NEXTVAL||SYS_CONTEXT('USERENV','INSTANCE'))
-----------------------------------------------------------------
                                                               11
SQL>
```

Instead of using the generated number by the sequence alone for the primary key values, when combined with SYS_CONTEXT('USERENV', 'DB_NAME), it generates a unique alphanumeric value of a character data type primary key.

Multi-Master Numbers Distribution

When Oracle GoldenGate is configured to support 3 systems for multi-master data replication, the first system sequence number begins at 1, the second system sequence begins at 2, and the third system sequence number begins at 3. All sequences increment by 3 as shown in Figure 5-19.

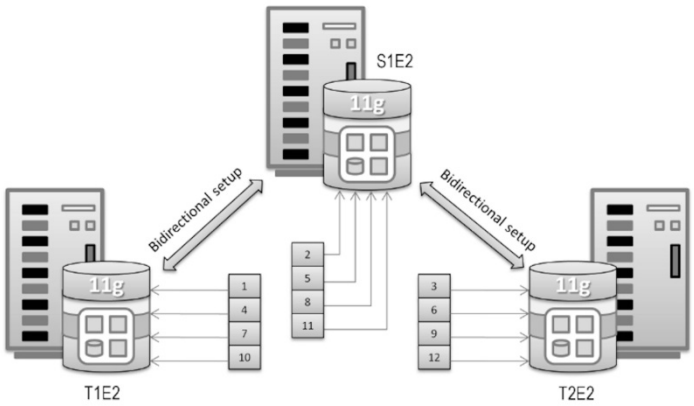


Figure 5-19: Multi-Master Numbers Sequencing

On node-1 (T1E2), create the sequence to begin at 1.

```
SQL> CREATE SEQUENCE cust_no_seq
  2  START WITH 1
  3  INCREMENT BY 3;

Sequence created.

SQL> SELECT cust_no_seq.nextval FROM dual;

   NEXTVAL
----------
         1

SQL> SELECT cust_no_seq.nextval FROM dual;

   NEXTVAL
----------
```

```
         4

SQL>
```

On node-2 (S1E2), create the sequence to begin at 2.

```
SQL> CREATE SEQUENCE cust_no_seq
  2  START WITH 2
  3  INCREMENT BY 3;

Sequence created.

SQL> SELECT cust_no_seq.nextval FROM dual;

   NEXTVAL
----------
         2

SQL> SELECT cust_no_seq.nextval FROM dual;

   NEXTVAL
----------
         5

SQL>
```

On node-3 (T2E2), create the sequence to begin at 3.

```
SQL> CREATE SEQUENCE cust_no_seq
  2  START WITH 3
  3  INCREMENT BY 3;

Sequence created.

SQL> SELECT cust_no_seq.nextval FROM dual;

   NEXTVAL
----------
         3

SQL> SELECT cust_no_seq.nextval FROM dual;

   NEXTVAL
----------
         6

SQL>
```

Database sequences created to avoid SQL insert conflicts are not to be included on the TABLE and MAP statements. To avoid encountering gaps due to database restart, use the NOCACHE attribute when creating sequences. Since the sequence START WITH and INCREMENT BY attributes are different on both systems, ascertain that the Oracle GoldenGate DDL replication is disabled when creating sequences and manually create them on both systems.

Data Conflicts

Without explicit conflict detection, applications will continue processing transactions of inconsistent data. Logical data conflict and inconsistencies should be handled using either Oracle GoldenGate, a built-in, or custom developed conflict detection and resolution techniques. In addition, virtual private database implementation minimizes data conflict as explained in this chapter. The demonstrated conflict resolution techniques are developed for an online application system for a two-node configuration architecture, synchronized using Oracle GoldenGate. The primary database (node-1) is located at Boston, MA and the target database (node-2) is located at Albany, NY as shown in Figure 5-20.

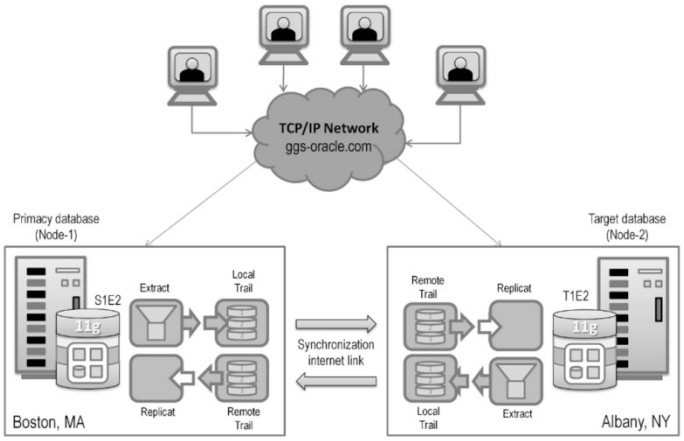


Figure 5-20: Bi-Directional Replication for Online Applications

Let us start by explaining the usages and applications of SQLEXEC parameter. This parameter is used in conjunction with custom data conflict handling technique developed in PL/SQL.

SQLEXEC Parameter

The use of the SQLEXC parameter extends Oracle GoldenGate capabilities, it enables invoking PL/SQL procedures and executes queries from within Oracle GoldenGate Extract or Replicat to support complex data conflict resolutions. The vast majority of data conflicts are handled using queries only, while procedures are mostly applied to more complex data conflicts and transformations.

Data manipulation and data conflict resolution are developed by utilizing Oracle GoldenGate built-in functions. The use of the functions @IF, @STREQ, @COMPUTE, @GETVAL and @GETENV is common. The following illustration

of using SQLEXEC is developed to support a de-normalized POLICIES table which includes a PT_DESC column.

```
SQL> DESC policies
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------
 POL_NO                                    NOT NULL NUMBER
 POL_FROM                                  NOT NULL DATE
 POL_TO                                    NOT NULL DATE
 POL_VALUE                                 NOT NULL NUMBER(8,2)
 POL_SUB_TOTAL                             NOT NULL NUMBER(8,2)
 PT_CODE                                   NOT NULL VARCHAR2(8)
 DT_CODE                                            VARCHAR2(8)
 POL_TOTAL                                 NOT NULL NUMBER(8,2)
 CUST_NO                                   NOT NULL NUMBER
 PT_DESC                                            VARCHAR2(128)

SQL>
```

The PT_DESC (Policy Type Description) column exists on the target database only. Hence, a lookup operation within the Replicat is necessary to complete the transaction successfully.

SQLEXEC with Queries

The components of SQLEXEC with a query specifies the query ID, which is used by the built-in function @GETVAL to assign the value to the target column. The PARAMS clause lists the pair of parameter name and value used in the query. The last clause is the COLMAP to perform column mapping which references the query columns list using the query ID.

The Replicat MAP statement uses SQLEXEC to fetch the value for the column PT_DESC using PT_CODE as a parameter.

```
MAP osm$repapi.policies, TARGET osm$repapi.policies, &
    SQLEXEC (ID lookup_pt, &
             QUERY " SELECT pt_desc
                     FROM   osm$repapi.policy_types
                     WHERE  pt_code = :pt_code_param ", &
             PARAMS (pt_code_param = pt_code)), &
             COLMAP (USEDEFAULTS, pt_desc = @GETVAL(lookup_pt.pt_desc));
```

The Replicat (RS1E203) parameter file using SQLEXEC with query clause.

```
REPLICAT rs1e203
SETENV (ORACLE_SID=T1E2)
SETENV (NSL_LANG=AMERICAN_AMERICA.AL32UTF8)
USERID ggs_admin, PASSWORD oracle
ASSUMETARGETDEFS
DISCARDFILE ./dirrpt/rs1e203.dsc, PURGE
MAP osm$repapi.customers, TARGET osm$repapi.customers;
```

```
MAP osm$repapi.policies, TARGET osm$repapi.policies, &
    SQLEXEC (ID lookup_pt, &
             QUERY " SELECT pt_desc
                     FROM   osm$repapi.policy_types
                     WHERE  pt_code = :pt_code_param ", &
             PARAMS (pt_code_param = pt_code)), &
             COLMAP (USEDEFAULTS, pt_desc = @GETVAL(lookup_pt.pt_desc));
```

SQLEXEC with Stored Procedures

SQLEXEC invokes stored procedure to perform advanced data transformations and conflicts resolutions. The simplest form of using SQLEXEC with stored procedure is performing a table look operation. The below PL/SQL database procedure performs a table look up operation instead of using query.

```
SQL> CREATE OR REPLACE PROCEDURE lookup_pt(pt_code_param IN VARCHAR2,
  2                                        pt_desc_param OUT VARCHAR2) IS
  3  BEGIN
  4     SELECT pt_desc into pt_desc_param
  5     FROM   osm$repapi.policy_types
  6     WHERE  pt_code = pt_code_param;
  7  END;
  8  /

Procedure created.

SQL> GRANT EXECUTE on lookup_pt to ggs_admin;

Grant succeeded.

SQL>
```

The Replicat parameter file is modified to reference the stored procedure using the SQLEXEC parameter.

```
REPLICAT rs1e203
SETENV (ORACLE_SID=T1E2)
SETENV (NSL_LANG=AMERICAN_AMERICA.AL32UTF8)
USERID ggs_admin, PASSWORD oracle
ASSUMETARGETDEFS
DISCARDFILE ./dirrpt/rs1e203.dsc, PURGE
MAP osm$repapi.customers, TARGET osm$repapi.customers;
MAP osm$repapi.policies, TARGET osm$repapi.policies, &
    SQLEXEC (SPNAME osm$repapi.lookup_pt, &
             ID     lookup_pt, &
             PARAMS (pt_code_param = pt_code)), &
             COLMAP (USEDEFAULTS, &
                     pt_desc = @GETVAL(lookup_pt.pt_desc_param));
```

The above SQLEXEC invokes a stored procedure by specifying the following clauses:

SPNAME is the name of the stored procedure. Ascertain granting Oracle GoldenGate database user to EXECUTE privilege on the called procedure.

ID is the logical name for the procedure and is needed by the @GETVAL function to retrieve the look up value.

PARAMS is the list of input parameters and their initial values.

COLMAP performs column mapping which references the output parameters using the logical name.

SQLEXEC as Stand-Alone Parameter

The third form of SQLEXEC is a stand-alone. This is a one-time call-out to execute query, DDL command, or stored procedure. This type of SQLEXEC is valid for Extract and Replicat parameter files.

The next parameter files ES1E203.PRM use SQLEXEC parameters. The first parameter executes a query to avoid IDLE_TIME enforced by the user profile.
The second parameter executes the DDL command TRUNCATE. The third parameter executes a stored procedure. Notice the procedure must not include input or output formal parameters.

```
EXTRACT es1e203
SETENV (ORACLE_SID=S1E2)
SETENV (ORACLE_HOME=/d01/app/oracleSE/product/11.2.0/dbhome_1)
SETENV (NLS_LANG=AMERICAN_AMERICA.AL32UTF8)
USERID ggs_admin, PASSWORD oracle
EXTTRAIL ./dirdat/sc
TRANLOGOPTIONS EXCLUDEUSER ggs_admin
SQLEXEC "SELECT * FROM DUAL" EVERY 55 MINUTES
SQLEXEC "TRUNCATE TABLE osm$repapi.logging" EVERY 1 DAYS
SQLEXEC "CALL send_notify()" EVERY 10 MINUTES
TABLE osm$repapi.customers;
TABLE osm$repapi.policies;
```

Custom Data Conflicts Handling

Consider an online application for managing a single inventory site. The inventory database is replicated across two cities – Boston, MA and Albany, NY as depicted in Figure 5-20. Customers' requests are directed to either database based on workload determined by the load balancer. Hence, data conflict is unavoidable. A classical data conflict may occur due to updates occurring at the time. When a conflict occurs, several resolutions are considered and should be governed by business rules.

- Use the net difference instead of using the after image value
- Perform comparison on other columns to accept, reject, or resolve transaction modifications
- Use the timestamp (date and time) then decide to accept or to reject the transaction modifications
- Manually handle the conflict by mapping data to an exception table

Conflicts Resolution Using SQLEXEC

When simultaneous bi-directional transactions update the same non-key row value on node-1 and on node-2, data divergence occurs slightly. This is a form of data conflict and must be detected and resolved.

Data Divergence Scenario

The customer has sent a change of email address request (non-key column). The request is received and routed to an agent at Boston. Shortly thereafter, the same customer sends another change of email address request with a different email from the previous request. The request is received and routed to an agent at Albany. Here is an illustration of processing this request without data conflict handling in place.

Node-1 Database: Boston	Node-2 Database: Albany

SQL> SELECT cust_email
  2 FROM   customers
  3 WHERE cust_no=100200;
 	
CUST_EMAIL
--------------- 	
fawzi@yahoo.com
 	
SQL>
	SQL> SELECT cust_email
	  2 FROM customers
	  3 WHERE cust_no=100200;
	
	CUST_EMAIL
	--------------- 	
	fawzi@yahoo.com
	
	SQL>

Data pump PS1E203 is down due to network failure	Data pump PT1E203 is down due to network failure

SQL> UPDATE customers SET
  2 cust_email='fawzi@gmail.com'
  3 WHERE cust_no=100200;
 	
1 row updated.
 	
SQL> COMMIT;
 	
Commit complete.
 	
SQL>
	SQL> UPDATE customers SET
	  2 cust_email='fawzi@mail.com'
	  3 WHERE cust_no=100200;
	
	1 row updated.
	
	SQL> COMMIT;
	
	Commit complete.
	
	SQL>

SQL> SELECT cust_email
  2 FROM   customers
  3 WHERE cust_no=100200;
 	
CUST_EMAIL
	SQL> SELECT cust_email
	  2 FROM customers
	  3 WHERE cust_no=100200;
	
	CUST_EMAIL

---------- fawzi@gmail.com SQL>	---------- fawzi@mail.com SQL>
Network restored. The data pump PS1E203 started automatically.	Network restored. The data pump PT1E203 started automatically.
SQL> SELECT cust_email 2 FROM customers 3 WHERE cust_no=100200; CUST_EMAIL ---------- fawzi@mail.com SQL>	SQL> SELECT cust_email 2 FROM customers 3 WHERE cust_no=100200; CUST_EMAIL ---------- fawzi@gmail.com SQL>

There is no error, but notice that the change of address resulted in mismatched CUST_EMAIL values between the two databases. This is a simple data conflict, yet critical to be addressed to avoid a data integrity problem.

Data Divergence Handling

Oracle GoldenGate captures before and after image of updates to support custom and built-in conflict resolutions by performing a compare operation before applying the transactions on the target database. Data divergence conflicts are prevented. Figure 5-21 shows the workflow of updating the email for CUST_NO=100200. Since there is no data segmentation at the application level, a custom data conflict is developed.

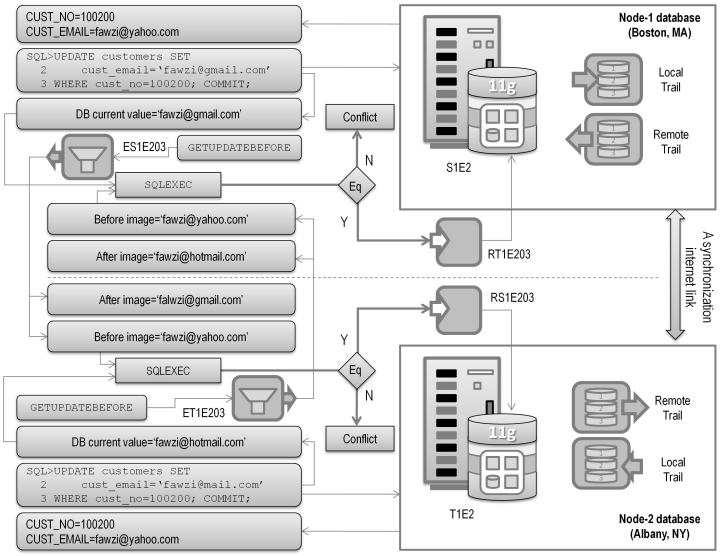


Figure 5-21: Resolving Conflict Using SQLEXEC

The Replicat applies the transaction to the database and the conflict occurs silently. The conflict is resolved either using the built-in conflict detection and resolution (CDR) or customer resolution using the SQLEXEC parameter. Using either technique, the database current value is compared with before image value from the trail record. If the two values are equal, then the transaction is applied. Otherwise, a data conflict is detected and needs to be handled accordingly.

Using the parameter SQLEXEC to handle the above data conflict requires using a query or a procedure to fetch the current database value and compare with the before image from the trail record. If the two values are identical, then apply the transaction. Otherwise report or handle data conflict. The stored procedure GET_EMAIL for handling this type of update conflict follows.

```
SQL> CREATE OR REPLACE PROCEDURE get_email(pcust_no    IN NUMBER,
  2                                        pcust_email OUT VARCHAR2) IS
  3  BEGIN
  4     SELECT cust_email into pcust_email
  5     FROM   osm$repapi.customers
  6     WHERE  cust_no = pcust_no;
  7  END;
  8  /

Procedure created.

SQL>
```

The Extract ES1E203 uses the parameter GETUPDATEBEFORES to force capture the transaction post values and before image values for use by the Replicat to detect conflict. Also, to support operation granularity, the parameter GETBEFORECOLS is used in conjunction with GETUPDATEBEFORES to associate the before image with the operation type of UPDATE, DELETE, or both. The argument ALL specifies the before image value is for all columns. The example below illustrates using GETBEFORECOLS to capture the before image values for update and delete operations on the CUST_EMAIL column only. The argument KEYINCLUDING includes capture of the primary key before image value.

```
TABLE osm$repapi.customers &
      GETBEFORECOLS(ON UPDATE KEYINCLUDING (cust_email), &
                    ON DELETE KEYINCLUDING (cust_email));
```

The parameter GETUPDATEBEFORES remains in effect for all subsequent listed tables until the parameter IGNOREUPDATEBEFORES is encountered – enabling control of the before image capture for a specific list of tables.

```
EXTRACT es1e203
SETENV (ORACLE_SID=S1E2)
SETENV (ORACLE_HOME=/d01/app/oracleSE/product/11.2.0/dbhome_1)
SETENV (NLS_LANG=AMERICAN_AMERICA.AL32UTF8)
```




```
USERID ggs_admin, PASSWORD oracle
EXTTRAIL ./dirdat/sc
TRANLOGOPTIONS EXCLUDEUSER ggs_admin
GETUPDATEBEFORES
TABLE osm$repapi.customers GETBEFORECOLS(ON UPDATE ALL);
IGNOREUPDATEBEFORES
TABLE osm$repapi.policies;
```

The Replicat RS1E203 parameter file includes the SOURCEDEFS parameter for mapping non-identical tables structure. Using SQLEXEC to invoke the procedure GET_EMAIL returns the current database value for compression with the before image value. The COLMAP clause includes reference to the Oracle GoldenGate built-in function to detect conflict. If the built-in function @STREQ returns 1 (true) then no conflict was detected. Otherwise, a conflict has been detected and the column CUST_EMAIL assigned the value 'conflict@pt.com' as a means of notification and intervention.

```
cust_email     = @IF(@STREQ(BEFORE.cust_email,
                     @GETVAL(get_email.pcust_email)),
                     cust_email, 'conflict@pt.com'),
```

```
REPLICAT rs1e203
SETENV (ORACLE_SID=T1E2)
SETENV (NSL_LANG=AMERICAN_AMERICA.AL32UTF8)
USERID ggs_admin, PASSWORD oracle
SOURCEDEFS /d01/oracle/ggs/dirdef/defgen.def
--ASSUMETARGETDEFS
DISCARDFILE ./dirrpt/rs1e203.dsc, PURGE
GETUPDATES
GETINSERTS
GETDELETES
MAP osm$repapi.customers, TARGET osm$repapi.customers, &
    SQLEXEC (ID get_email, &
        SPNAME osm$repapi.get_email, &
        PARAMS (pcust_no = cust_no)), &
        COLMAP (cust_no        = cust_no,
                cust_name      = cust_name,
                cust_email     = @IF(@STREQ(BEFORE.cust_email,
                                     @GETVAL(get_email.pcust_email)),
                                     cust_email, 'conflict@pt.com'),
                cust_mobile    = cust_mobile,
                cust_address   = cust_address,
                cust_commit_ts = @GETENV('GGHEADER', 'COMMITTIMESTAMP'));
--
MAP osm$repapi.policies, TARGET osm$repapi.policies, &
    SQLEXEC (SPNAME osm$repapi.lookup_pt, &
        ID     lookup_pt, &
        PARAMS (pt_code_param = pt_code)), &
        COLMAP (USEDEFAULTS, &
                pt_desc = @GETVAL(lookup_pt.pt_desc_param));
```

The following scenario illustrates conflict detection for an update operation. The Replicat applying the transaction detects the conflict and assigns the



CUST_EMAIL column with temporary email 'conflict@pt.com' for further action.

Node-1 Database: Boston	Node-2 Database: Albany

SQL> SELECT cust_email
  2 FROM   customers
  3 WHERE cust_no=4846;
 	
CUST_EMAIL
-------------------- 	
deema@yahoo.com
 	
SQL>
	SQL> SELECT cust_email
	  2 FROM customers
	  3 WHERE cust_no=4846;
	
	CUST_EMAIL
	-------------------- 	
	deema@yahoo.com
	
	SQL>

Data pump PS1E203 is down due to network failure	Data pump PT1E203 is down due to network failure

SQL> UPDATE customers SET
  2 cust_email='deema@gmail.com'
  3 WHERE cust_no=4846;
 	
1 row updated.
 	
SQL> COMMIT;
 	
Commit complete.
 	
SQL>
	SQL> UPDATE customers SET
	  2 cust_email='deema@hotmail'
	  3 WHERE cust_no=4846;
	
	1 row updated.
	
	SQL> COMMIT;
	
	Commit complete.
	
	SQL>

SQL> SELECT cust_email
  2 FROM   customers
  3 WHERE cust_no=4846;
 	
CUST_EMAIL
--------------- 	
deema@gmail.com
 	
SQL>
	SQL> SELECT cust_email
	  2 FROM customers
	  3 WHERE cust_no=4846;
	
	CUST_EMAIL
	-------------- 	
	deema@hotmail.com
	
	SQL>

Network restored. The data pump PS1E203 started automatically.	Network restored. The data pump PT1E203 started automatically.

SQL> SELECT cust_email
  2 FROM   customers
  3 WHERE cust_no=4846;
 	
CUST_EMAIL
------------- 	
deema@hotmail
 	
SQL>
	SQL> SELECT cust_email
	  2 FROM customers
	  3 WHERE cust_no=4846;
	
	CUST_EMAIL
	--------------- 	
	conflict@pt.com
	
	SQL>

Refer to Chapter 11 to learn more regarding generating the definition file and advanced transformation.

Applying Net Difference Resolution Using SQLEXEC

When the application manipulates a single entity such as inventory level, bank account, order entry, etc., in an active-active bi-directional Oracle GoldenGate topology, conflicts may occur and must be handled. Figure 5-22 illustrates an ATM workflow for simultaneous transactions executed at the same time.

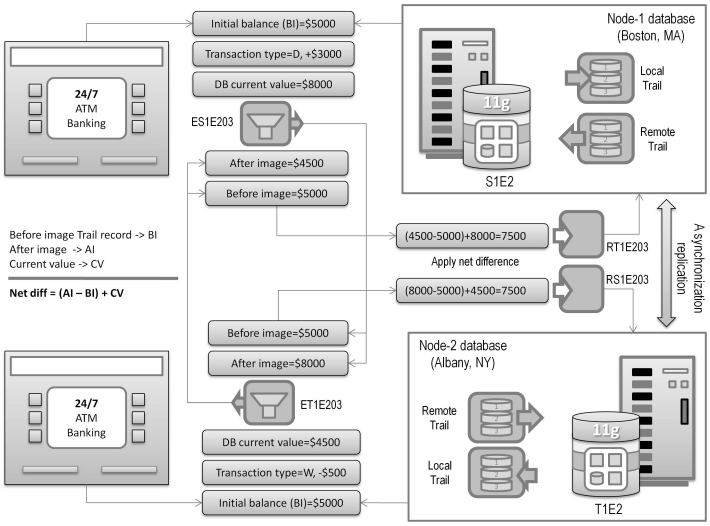


Figure 5-22: Applying Net Difference Resolution Using SQLEXEC

The initial account balance is $5,000. A deposit transaction (D) for the amount of $3,000 is entered using the primary ATM card. This transaction is located at Boston, MA increasing the account balance to $8,000.

Almost simultaneously, while the initial balance is still $5000, a withdrawal transaction (W) for the amount of $500 is entered using the secondary ATM card. This transaction is located at Albany, NY decreasing the account balance to $4,500.

When the Replicat applies the transaction on either database, conflict occurs. The database located at Boston, MA has a current value of $8,000 and the database located at Albany, NY has a current value of $4,500. By comparing the trail record and the current database values, the data conflict is detected and needs to be resolved — using either SQLEXEC for applying the net difference or Oracle GoldenGate built-in conflict detection and resolution (CDR) to calculate and apply the transaction correctly.

The resolution below is developed using SQLEXEC. The Replicat COLMAP clause determines and applies the net difference of: $5000+$3000-$500=$7500.

The net balance applied at Boston, MA is calculated as:

Net balance = ((Trail after image - Trail before image) + current value)
= ((4500 – 5000) + 8000) = $7,500

The net balance applied at Albany, NY is calculated as:

Net balance = ((Trail after image - Trail before image) + current value)
= ((8000 – 5000) + 4500) = $7,500

To demonstrate data conflict by applying the net difference value, the table CURRENT_ACCOUNTS exists on node-1 database (S1E2) and node-2 database (T1E2). It is configured on bi-directional configuration with transactions entered from both nodes.

```
SQL> DESC current_accounts
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------
 ACCT_NO                                   NOT NULL NUMBER
 ACCT_NAME                                 NOT NULL VARCHAR2(64)
 ACCT_AMOUNT                                        NUMBER
 ACCT_TS                                            TIMESTAMP(6)

SQL>
SQL> SELECT ACCT_AMOUNT
  2  FROM   current_accounts
  3  WHERE  acct_no=49664996;

ACCT_AMOUNT
-----------
       5000

SQL>
```

The Extract parameter file lists the tables. The parameters GETUPDATEBEFORES and GETBEFORECOLS enable capture of the before image values.

```
EXTRACT es1e203
SETENV (ORACLE_SID=S1E2)
SETENV (ORACLE_HOME=/d01/app/oracleSE/product/11.2.0/dbhome_1)
SETENV (NLS_LANG=AMERICAN_AMERICA.AL32UTF8)
USERID ggs_admin, PASSWORD oracle
EXTTRAIL ./dirdat/sc
TRANLOGOPTIONS EXCLUDEUSER ggs_admin
GETUPDATEBEFORES
TABLE osm$repapi.customers &
      GETBEFORECOLS(ON UPDATE KEYINCLUDING (cust_email), &
                    ON DELETE KEYINCLUDING (cust_email));
IGNOREUPDATEBEFORES
TABLE osm$repapi.policies;
```

```
GETUPDATEBEFORES
TABLE osm$repapi.current_accounts &
      GETBEFORECOLS(ON UPDATE ALL);
IGNOREUPDATEBEFORES
```

The Replicat parameter file uses SQLEXEC to execute a query then compares the before image with the query return value. The COLMAP clause uses Oracle GoldenGate built-in @COMPUTE function to determine the net difference.

```
REPLICAT rs1e203
SETENV (ORACLE_SID=T1E2)
SETENV (NSL_LANG=AMERICAN_AMERICA.AL32UTF8)
USERID ggs_admin, PASSWORD oracle
SOURCEDEFS /d01/oracle/ggs/dirdef/defgen.def
DISCARDFILE ./dirrpt/rs1e203.dsc, PURGE
GETUPDATES
GETINSERTS
GETDELETES
MAP osm$repapi.customers, TARGET osm$repapi.customers, &
    SQLEXEC (ID     get_email, &
             SPNAME osm$repapi.get_email, &
             PARAMS (pcust_no = cust_no)), &
    COLMAP (cust_no        = cust_no,
            cust_name      = cust_name,
            cust_email     = @IF(@STREQ(BEFORE.cust_email,
                               @GETVAL(get_email.pcust_email)), cust_email,
                               'conflict@pt.com'),
            cust_mobile    = cust_mobile,
            cust_address   = cust_address,
            cust_commit_ts = @GETENV('GGHEADER', 'COMMITTIMESTAMP'));
--
MAP osm$repapi.policies, TARGET osm$repapi.policies, &
    SQLEXEC (SPNAME osm$repapi.lookup_pt, &
             ID     lookup_pt, &
             PARAMS (pt_code_param = pt_code)), &
             COLMAP (USEDEFAULTS, &
                     pt_desc = @GETVAL(lookup_pt.pt_desc_param));
--
MAP osm$repapi.current_accounts, TARGET osm$repapi.current_accounts,
    SQLEXEC (ID lookup_acct_amount, &
        QUERY " SELECT acct_amount FROM osm$repapi.current_accounts
                WHERE  acct_no = :vacct_no ", &
        PARAMS (vacct_no = acct_no)), &
        COLMAP (acct_no     = acct_no,
                acct_name   = acct_name,
                acct_amount = @COMPUTE(((acct_amount - BEFORE.acct_amount) +
                               @GETVAL(lookup_acct_amount.acct_amount))),
                acct_ts     = @GETENV('GGHEADER', 'COMMITTIMESTAMP'));
```

The below sequence of query and update statements demonstrates the same ACCT_NO is updated when the data pump is ABENDED due to network failure, which should lead to data conflict when the data pump is re-started. However, by applying the net difference value, the ACCT_AMOUNT value column is correctly calculated and converged.



Node-1 database, Boston	Node-2 database, Albany

SQL> SELECT ACCT_AMOUNT
  2 FROM   current_accounts
  3 WHERE acct_no=49664996;
 	
ACCT_AMOUNT
----------- 	
       5000
 	
SQL>
	SQL> SELECT ACCT_AMOUNT
	  2 FROM current_accounts
	  3 WHERE acct_no=49664996;
	
	ACCT_AMOUNT
	----------- 	
	       5000
	
	SQL>

Data pump PS1E203 is down due to network failure	Data pump PT1E203 is down due to network failure

SQL> UPDATE current_accounts SET
  2   acct_amount=acct_amount+3000
  3 WHERE acct_no=49664996;
 	
1 row updated.
 	
SQL> COMMIT;
	SQL> UPDATE current_accounts SET
	  2 acct_amount=acct_amount-500
	  3 WHERE acct_no=49664996;
	
	1 row updated.
	
	SQL> COMMIT;

SQL> SELECT acct_amount
  2 FROM   current_accounts
  3 WHERE acct_no=49664996;
 	
ACCT_AMOUNT
----------- 	
       8000
	SQL> SELECT acct_amount
	  2 FROM current_accounts
	  3 WHERE acct_no=49664996;
	
	ACCT_AMOUNT
	----------- 	
	       4500

Network restored. The data pump PS1E203 started automatically.	Network restored. The data pump PT1E203 started automatically.

SQL> SELECT acct_amount
  2 FROM   current_accounts
  3 WHERE acct_no=49664996;
 	
ACCT_AMOUNT
----------- 	
       7500
 	
SQL>
	SQL> SELECT acct_amount
	  2 FROM current_accounts
	  3 WHERE acct_no=49664996;
	
	ACCT_AMOUNT
	----------- 	
	       7500
	
	SQL>

DML Conflict Types

Fundamentally, conflicts occur and must be handled. Why? It is because Oracle GoldenGate Replicat preserves data integrity when applying transactions. It does so by maintaining the transactions commit order and uses the target database native calls when applying transactions, which have database constraints enabled.

An active-active bi-directional configuration must implement, detect, and resolve data conflict techniques. These conflicts arising from DML transactions manipulate the same row stored on the source and target databases. The resolution is detected and resolved using either built-in or custom developed conflict detection and resolution techniques.

The resolution is developed as part of the transaction integrity or by applying basic resolution rules such as timestamps. There are (3) types of conflicts that occur when executing SQL statements on asynchronous mode:

Insert Conflicts

Due to enabled and valid primary key or unique key constraints on tables, insert conflicts occur when duplicate values violate the constraint. Without conflict handling the Replicat terminates with an error.

Update Conflicts

Modifying the same row at the same time results in update conflicts. However, because access to the updated row's before and after image is available, conflicts are resolved by applying suitable resolution methods.

Delete Conflicts

This type of conflict commonly occurs when the Replicat attempts to apply a delete transaction, but the row is not found on the target database. This is because the row is already deleted or updated by another transaction.

Data Conflict and Resolution Roadmap

The fundamental rule when developing conflict resolution is to consider the technique consistency. This allows applying the same technique across the vast majority of the database tables. There are several approaches to data conflict avoidance, detection, and resolution. The combination of avoidance and resolution promotes data convergence and data consistency. Oracle GoldenGate provides its own built-in function and conflict detection and resolution (CDR). It is the distinctive feature of Oracle GoldenGate 11g Release 2, explained and demonstrated shortly.

The techniques and order for handling data conflicts in an active-active bi-directional topology is shown in Figure 5-23. Since Oracle GoldenGate is mostly deployed after the application, it is less likely the application is conflict-aware. To minimize the application changes and make Oracle GoldenGate data replication transparent on bi-directional topology, the roadmap begins by implementing data segmentation as an avoidance technique. When data segmentation may not totally avoid data conflict, partitioning the applications provides the second line of defense.

In addition to data segmentation and application partitioning, Oracle GoldenGate

built-in conflict detection and resolution applies consistent methods to resolve conflicts using declarative parameter settings for all types of DML conflict types.

For complex data conflict detection and resolution, using the SQLEXEC parameter requires developing PL/SQL sub-programs and SQL queries that are invoked from within Oracle GoldenGate.

When SQLEXEC is not meeting the requirements, user exits is another option to consider for handling sophisticated conflicts beyond those provided by Oracle GoldenGate built-in functions and techniques. User exits are developed in C and C++ programming language and invoked from the Extract and Replicat, but mostly adapted for advanced integration and data transformation.

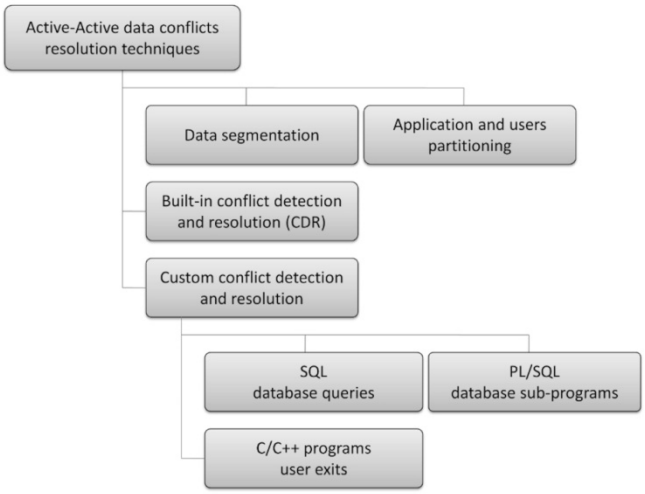


Figure 5-23: Conflicts Resolutions Roadmap

Data Segmentation

When the application is not replication-aware, then implementing data segmentation is the first approach of the data conflict avoidance technique. It is developed and deployed within the database only. An example is implementing Oracle Virtual Private Database (VPD) to dynamically apply a table-level security policy function to avoid the likelihood of having the same row concurrently accessed by multiple users.

Application Segmentation

Application segmentation is implemented from within the virtual private database – making the application replication-aware. Then by segregating applications so that

groups of users are associated with an application, accessing a subset of tables reduces the possibility of data conflicts. Application segmentation is developed using application contexts such as locations, transaction type, etc. This is an avoidance technique to minimize and not to entirely eliminate data conflicts.

Built-In Conflict Detection and Resolution

Oracle GoldenGate 11g Release 2 is shipped with built-in conflict detection and resolution (CDR). This is a core feature of any data replication software. Using simple declarative conflicts and methods, Oracle GoldenGate handles the majority of conflicts for database applications accessing active-active bi-directional topology.

Custom Conflict Resolution

Using queries, PL/SQL sub-programs, and user exits enable developing conflict resolution beyond the capabilities of Oracle GoldenGate built-in conflict resolution methods. The technique extends the capabilities of Oracle GoldenGate to include using the database built-in features for events handling.

Built-In Conflict Detection and Resolution

While complex business logic custom data conflicts resolutions are developed using PL/SQL, SQL queries, and C/C++ user exits, the majority of data conflicts can be handled by adapting Oracle GoldenGate automatic conflicts detection and resolution (CDR) techniques.

The advantages of employing the built-in CDR are performance, consistency, and lower maintenance cost. Resolving conflicts using CDR delivers better performance when compared to custom resolution using SQLEXEC to invoke database PL/SQL procedures or SQL executing queries. The built-in resolution and detection avoid runtime context switching. Since the vast majority of tables are handled using similar conflict resolution methods, CDR delivers consistency using declarative conflict types and methods. Lower maintenance cost is also an advantage of using CDR. For example, when modifying CDR methods, deploying CDR new features, adding new table columns and database tables, maintenance is executed within GoldenGate only. Oracle GoldenGate built-in conflict detection and resolution uses the four parameters listed in Table 5-4, promoting ease of configuration and deployment.

Parameter	Description



GETUPDATEBEFORES	An Extract parameter. It forces capture of the transaction post values (after image) and before image values for use by the Replicat to detect conflicts.
GETBEFORECOLS	An Extract parameter. It associates the before image with the operation type of UPDATE, DELETE, or both and the column list.
COMPARECOLS	A Replicat parameter. It lists the columns to check for conflicts with the target database current value. A mismatch is a conflict and handled by the RESOLVECONFLICT settings.
RESOLVECONFLICT	A Replicat parameter. It specifies the operation type, the resolution type, and the method applied to resolve the conflict. The parameter is repeated for each resolution to handle the different operations.

Table 5-4: Built-In Conflict Detection and Resolution Parameters

For tables referenced by transactions on an active-active bi-directional topology, the parameter RESOLVECONFLICT is configured to handle conflicts. Within the same MAP statement, it resolves all types of data conflict transactions for UPDATE, DELETE, and INSERT SQL statements. The below illustrates resolving update conflicts. The named resolution DELTA_RESOLUTION_METHOD uses the method USEDELTA for the column ACCT_AMOUNT. The resolution MAX_RESOLUTION_METHOD uses the method USEMAX for the columns ACCT_NAME and ACCT_TS.

```
RESOLVECONFLICT (UPDATEROWEXISTS,
                    (DELTA_RESOLUTION_METHOD, USEDELTA, COLS (acct_amount)),
                    (MAX_RESOLUTION_METHOD, USEMAX(acct_ts),
                     COLS (acct_name, acct_ts))
```

Figure 5-24 depicts the hierarchy of using Oracle GoldenGate built-in conflict detection and resolution (CDR).

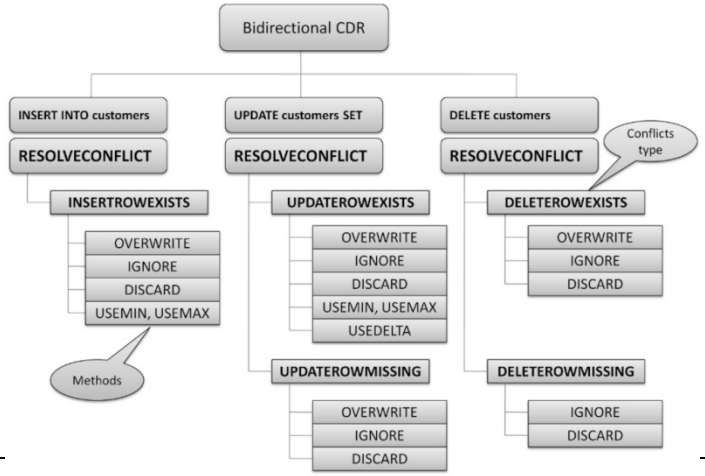


Figure 5-24: Built-In Conflicts Detection and Resolution

The parameter RESOLVECONFLICT takes any of the arguments listed in Table 5-5. Depending on the business rules for data conflict resolution handling, associate the argument with a resolution method.

Conflict type	Description
INSERTROWEXISTS	Handles unique constraint violation. The target database has primary key or unique key constraint enabled.
UPDATEROWEXISTS	Handles update data conflict when the before image value for one or more columns differ from the current database value.
UPDATEROWMISSING	Handles data conflict when the row does not exist on the target database.
DELETEROWEXISTS	Handles delete data conflict when before image value for one or more columns differ from the current database value.
DELETEROWMISSING	Handles data conflict when the row deleted on source does not exist on the target database.

Table 5-5: RESOVLECONFLICT Parameter Conflict Types

Each conflict type is associated with one resolution method. The following is the list of methods applied to a DEFAULT or named resolution:

- OVERWRITE
- IGNORE
- DISCARD
- USEMIN
- USEMAX
- USEDELTA

Figure 5-25 outlines the syntax of the Replicat built-in conflict detection and resolution. The combination of COMPARECOLS and RESOLVECONFLICT parameters handles the operation types and the conflict types. For each conflict type, a resolution method is selected and may be applied to one or more columns.

```
MAP  <source table>, TARGET <target table>,
COMPARECOLS (
  { ON UPDATE {ALL | KEY | KEYINCLUDING(col1, col2,..., coln) |
                           KEYEXCULDING(col1, col2,..., coln)}|
   ,ON DELETE {ALL | KEY | KEYINCLUDING(col1, col2,..., coln) |
                           KEYEXCULDING(col1, col2,..., coln)} )
RESOLVECONFLICT (
    { INSERTROWEXISTS  |
      UPDATEROWEXISTS  | UPDATEROWMISSING |
      DELETEROWEXISTS  | DELETEROWMISSING }
                    (
                      { DEFAULT | <resolution name>},
                      { USEMAX (column) |
                        USEMIN (column) |
                        DELTA           |
                        DISCARD         |
                        OVERWRITE }
                    ),
                       COLS (col1, col2,..., coln)
                  )
```

Figure 5-25: Built-In Conflict Detection and Resolution Syntax

Configuring Built-In Conflict Detection and Resolution

This section implements active-active configuration using Oracle GoldenGate built-in conflict detection and resolution (CDR). Consider the CURRENT_ACCOUNTS table, the Extract and Replicat are configured for CDR. The ACCT_AMOUNT column is handled using the DELTA method. The ACCT_NAME and ACCT_TS columns are handled using the USEMAX method.

```
SQL> DESC current_accounts
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------
 ACCT_NO                                   NOT NULL NUMBER
 ACCT_NAME                                 NOT NULL VARCHAR2(64)
 ACCT_AMOUNT                                        NUMBER
 ACCT_TS                                            TIMESTAMP(6)

SQL>
```

As part of the change data capture (CDC), ensure to run the ADD TRANDATA command on node-1 and node-2 databases using the syntax below.

```
GGSCI (apps-srv1) 4> ADD TRANDATA osm$repapi.current_accounts ALLCOLS
```

Node-1 Extract and Replicat Parameter Files

The parameter files for node-1 configured to support the Extract ES1E203, the data pump PS1E203 and the Replicat RT1E203.

The use of the parameters GETUPDATEBEFORES, GETBEFORECOLS and IGNOREUPDATEBEFORES is for supporting capture of the before image on the trail records.

```
EXTRACT es1e203
SETENV (ORACLE_SID=S1E2)
SETENV (ORACLE_HOME=/d01/app/oracleSE/product/11.2.0/dbhome_1)
SETENV (NLS_LANG=AMERICAN_AMERICA.AL32UTF8)
USERID ggs_admin, PASSWORD oracle
EXTTRAIL ./dirdat/sc
TRANLOGOPTIONS EXCLUDEUSER ggs_admin
GETUPDATEBEFORES
TABLE osm$repapi.customers &
      GETBEFORECOLS(ON UPDATE KEYINCLUDING (cust_email), &
                    ON DELETE KEYINCLUDING (cust_email));
IGNOREUPDATEBEFORES
TABLE osm$repapi.policies;
GETUPDATEBEFORES
TABLE osm$repapi.current_accounts &
      GETBEFORECOLS(ON UPDATE ALL, ON DELETE ALL);
IGNOREUPDATEBEFORES
```

The data pump PS1E203 is configured on PASSTHRU mode, sending data to node-2 without transformation.

```
EXTRACT ps1e203
USERID ggs_admin, PASSWORD oracle
PASSTHRU
RMTHOST apps-srv2, MGRPORT 7806
RMTTRAIL ./dirdat/tc
TABLE osm$repapi.customers;
TABLE osm$repapi.policies;
TABLE osm$repapi.current_accounts;
```

The Replicat RT1E203 applies transactions to the node-1 database.

```
REPLICAT rt1e203
SETENV (ORACLE_SID=S1E2)
SETENV (NSL_LANG=AMERICAN_AMERICA.AL32UTF8)
USERID ggs_admin, PASSWORD oracle
ASSUMETARGETDEFS
DISCARDFILE ./dirrpt/rt1e203.dsc, PURGE
MAP osm$repapi.customers, TARGET osm$repapi.customers;
MAP osm$repapi.policies, TARGET osm$repapi.policies;
MAP osm$repapi.current_accounts, TARGET osm$repapi.current_accounts,
    COMPARECOLS (ON UPDATE ALL, ON DELETE ALL),
    RESOLVECONFLICT (UPDATEROWEXISTS,
```

```
                    (DELTA_RESOLUTION_METHOD, USEDELTA, COLS (acct_amount)),
                    (MAX_RESOLUTION_METHOD, USEMAX(acct_ts),
                     COLS (acct_name, acct_ts)),
                                          (DEFAULT, USEMAX(acct_ts))),
    RESOLVECONFLICT (UPDATEROWMISSING,  (DEFAULT, OVERWRITE)),
    RESOLVECONFLICT (DELETEROWEXISTS,   (DEFAULT, OVERWRITE)),
    RESOLVECONFLICT (DELETEROWMISSING,  (DEFAULT, DISCARD)),
    RESOLVECONFLICT (INSERTROWEXISTS,   (DEFAULT, USEMAX(acct_ts)));
```

Node-2 Extract and Replicat Parameter Files

The set of node-2 parameter files are similar to node-1. The parameter files support the Extract ET1E203, the data pump PT1E203 and the Replicat RS1E203.

The Extract ET1E203 parameter file.

```
EXTRACT et1e203
SETENV (ORACLE_SID="T1E2")
SETENV (ORACLE_HOME="/d01/app/oracleSE/product/11.2.0/dbhome_1")
SETENV (NLS_LANG="AMERICAN_AMERICA.AL32UTF8")
USERID ggs_admin, PASSWORD oracle
TRANLOGOPTIONS EXCLUDEUSER ggs_admin
EXTTRAIL ./dirdat/sd
TABLE osm$repapi.customers;
TABLE osm$repapi.policies;
GETUPDATEBEFORES
TABLE osm$repapi.current_accounts &
      GETBEFORECOLS(ON UPDATE ALL, ON DELETE ALL);
IGNOREUPDATEBEFORES
```

The data pump PT1E203 parameter file.

```
EXTRACT pt1e203
USERID ggs_admin, PASSWORD oracle
PASSTHRU
RMTHOST apps-srv1, MGRPORT 7805
RMTTRAIL ./dirdat/td
TABLE osm$repapi.customers;
TABLE osm$repapi.policies;
TABLE osm$repapi.current_accounts;
```

The Replicat RS1E203 parameter file.

```
REPLICAT rs1e203
SETENV (ORACLE_SID=T1E2)
SETENV (NSL_LANG=AMERICAN_AMERICA.AL32UTF8)
USERID ggs_admin, PASSWORD oracle
ASSUMETARGETDEFS
DISCARDFILE ./dirrpt/rs1e203.dsc, PURGE
GETUPDATES
GETINSERTS
GETDELETES
```

```
MAP osm$repapi.customers, TARGET osm$repapi.customers, &
    SQLEXEC (ID get_email, &
             SPNAME osm$repapi.get_email, &
             PARAMS (pcust_no = cust_no)), &
             COLMAP (cust_no        = cust_no,
                     cust_name      = cust_name,
                     cust_email     = @IF(@STREQ(BEFORE.cust_email,
                                      @GETVAL(get_email.pcust_email)),
                                      cust_email, 'conflict@pt.com'),
                     cust_mobile    = cust_mobile,
                     cust_address   = cust_address,
                     cust_commit_ts = @GETENV('GGHEADER',
                                              'COMMITTIMESTAMP'));
--
MAP osm$repapi.policies, TARGET osm$repapi.policies, &
    SQLEXEC (SPNAME osm$repapi.lookup_pt, &
             ID     lookup_pt, &
             PARAMS (pt_code_param = pt_code)), &
             COLMAP (USEDEFAULTS, &
                     pt_desc = @GETVAL(lookup_pt.pt_desc_param));
--
MAP osm$repapi.current_accounts, TARGET osm$repapi.current_accounts,
    COMPARECOLS (ON UPDATE ALL, ON DELETE ALL),
    RESOLVECONFLICT (UPDATEROWEXISTS,
                     (DELTA_RESOLUTION_METHOD, USEDELTA, COLS (acct_amount)),
                     (MAX_RESOLUTION_METHOD, USEMAX(acct_ts),
                                          COLS (acct_name, acct_ts)),
                                    (DEFAULT, USEMAX(acct_ts))),
    RESOLVECONFLICT (UPDATEROWMISSING, (DEFAULT, OVERWRITE)),
    RESOLVECONFLICT (DELETEROWEXISTS,  (DEFAULT, OVERWRITE)),
    RESOLVECONFLICT (DELETEROWMISSING, (DEFAULT, DISCARD)),
    RESOLVECONFLICT (INSERTROWEXISTS,  (DEFAULT, USEMAX(acct_ts)));
```

Highly Available Applications

Most call center's applications are the front line for customer service companies that provide 24/7 services. Requests and inquiry calls coming from various input devices such as mobile phones, smart phone applications, and browser-based applications are system received and routed to the next available agent. Figure 5-26 shows the workflow for incoming customer calls. This is a high level description to demonstrate using Oracle Virtual Private Database (VPD) in supporting the data conflict avoidance technique for this kind of business environment.

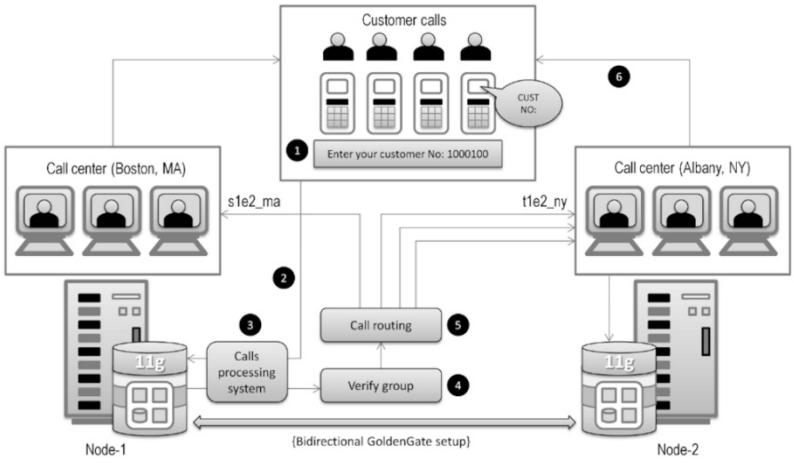


Figure 5-26: Call Center Applications Workflow

The customer calls in for inquires, such as requesting a policy renewal quotation. An automated system welcomes the customer. Then, the customer call is processed by assigning an agent to fulfill the request as follows:

1. The caller enters their customer number, which is the unique identifier.
2. The number is sent to the current database dedicated for receiving calls.
3. The call processing system validates the call and updates the calls log.
4. Based on the customer number, the owner group is determined.
5. Call routing is made based on the group name of the customer number.
6. An assigned agent receives the routed call and is locked with the customer.

The next section is developed on the basis of the above applications workflow. Data separation is achieved by routing requests and inquires to either Boston or Albany call centers. The objective is to maximize the return of investment (ROI) from licensing Oracle Database Enterprise Edition and Oracle GoldenGate software bundles for achieving a two-node architecture as an active-active topology applied for high availability and disaster recovery business requirements.

Transparent Avoidance Using Virtual Private Database

Virtual Private Database (VPD) is an advanced data conflict avoidance technique for transparent data segmentation. After creating database's table polices and implementing fine-grain access control and application contexts, SQL statements are dynamically modified to control rows (and columns) level access by database users. Figure 5-27 shows how users on node-1 and on node-2 access different subset of data, even when

the session is connected to the same database. Fine-grain access control limits rows access by implementing a database function that returns a predicate clause (WHERE) to virtually limit row access by the database users.

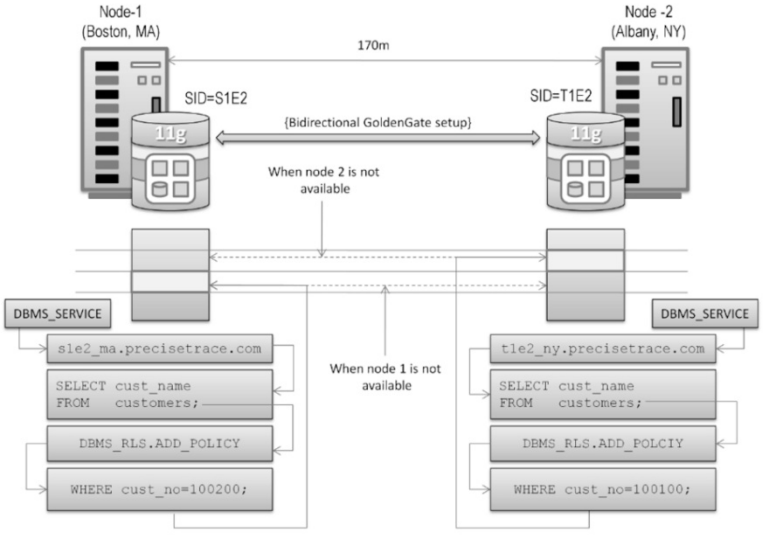


Figure 5-27: Data Segmentation Using Virtual Private Database

Our objective for using Virtual Private Database is to enforce data access based on ownership, so that a group of users within a database sees only their own accounts. The implementation is transparent – defined using policies that are independent of the applications. These active policies are cached in the SGA shared pool allowing in-memory access for faster evaluation and processing of fine-grain access control using SYS_CONTEXT function.

Begin by using DBMS_SERVICE PL/SQL supplied package to create non-default service names which will be used to associate a database user to one-and-only-one group at any time. However, in the event of a node failure, the user connects to the surviving node using the same service name but referencing the surviving node database service. Figure 5-27 shows the service name s1e2_ma for users connecting to node-1 database and t1e2_ny for users connecting to node-2 database.

The calls to DBMS_SERVICE.CREATE_SERVICE create the service names s1e2_ma and t1e2_ny, then start these service names for user connections. The procedure DBMS_SERVICE.CREATE_SERVICE requires the service name and service network name, use the service name to manage the service name.

```
SQL> BEGIN
  2   DBMS_SERVICE.CREATE_SERVICE(service_name=>'s1e2_ma',
  3                               network_name=>'s1e2_am.precisetrace.com');
  4  END;
  5  /

PL/SQL procedure successfully completed.

SQL> EXEC DBMS_SERVICE.START_SERVICE('s1e2_ma');

PL/SQL procedure successfully completed.

SQL> BEGIN
  2   DBMS_SERVICE.CREATE_SERVICE(service_name=>'t1e2_ny',
  3                               network_name=>'t1e2_ny.precisetrace.com');
  4  END;
  5  /

PL/SQL procedure successfully completed.

SQL> EXEC DBMS_SERVICE.START_SERVICE('t1e2_ny');

PL/SQL procedure successfully completed.
```

Use 'netca utility' to create the database connect string for the new service names and perform quick tests. By creating the two service names on node-1 and node-2 instances, they are readily available in the event of either node failure. The connect string for s1e2_ma is configured (t1e2_ny is not shown), followed by a successful test using the connect string s1e2_ma and t1e2_ny.

```
S1E2_MA =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = ggs_source)(PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = s1e2_ma.precisetrace.com)
    )
  )

SQL> CONN fawzi/oracle@s1e2_ma
Connected.
SQL> CONN deema/oracle@t1e2_ny
Connected.
SQL>
```

How Does Virtual Private Database Work?

The major advantage for using Virtual Private Database is ease of deployment and flexibility. After the policy is created, developed, and enabled, it becomes part of the database and access control applies across all applications of the table or view.

The workflow of executing an SQL application enforced by the Virtual Private Database security policy on the CUSTOMERS table is as follows:

1. The user accesses the CUSTOMERS table using SQL applications.
2. The application invokes the package for preparing and setting the context attribute values.
3. The policy function returns the predicate. It uses the session application context attributes stored in memory.
4. The database server dynamically modifies the statement by appending the returned predicate to SQL applications of step 1.
5. The session executes the dynamically generated SQL applications.

Implementing Virtual Private Database

The implementation of Virtual Private Database may require designing the database to provide the application context attribute values at runtime. Figure 5-28 shows a modified data model where customers are created by a group, meaning the customers and policies are owned and modified by a specific location (group). This model provides data conflict avoidance as the data is modified by transactions within the same database.

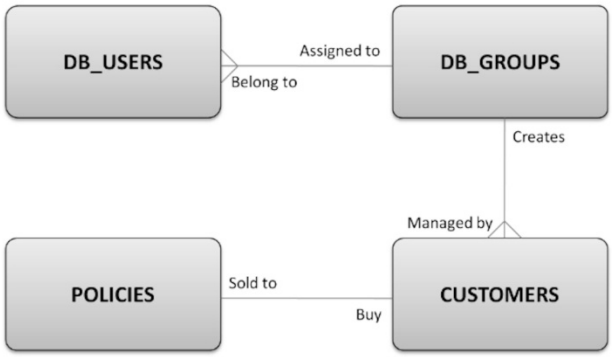


Figure 5-28: Data Modeling for VPD

The table structure of Figure 5-28 is listed below. The column GROUP_ID for each customer supports developing a conflict avoidance by group (location) and not by users. Transactions are processed as one unit of work and sent to the remote database after being processed by the local database.

```
SQL> DESC db_groups
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------
 GROUP_ID                                  NOT NULL VARCHAR2(30)
 GROUP_DESC                                NOT NULL VARCHAR2(128)
```

```
SQL> DESC db_users
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------
 USER_ID                                   NOT NULL VARCHAR2(30)
 USER_NAME                                 NOT NULL VARCHAR2(128)
 GROUP_ID                                  NOT NULL VARCHAR2(30)

SQL> DESC customers
 Name                                      Null? Type
 ----------------------------------------- --------  ----------------------
 CUST_NO                                   NOT NULL NUMBER
 CUST_NAME                                 NOT NULL VARCHAR2(128)
 CUST_EMAIL                                NOT NULL VARCHAR2(128)
 CUST_MOBILE                                         VARCHAR2(16)
 CUST_ADDRESS                                        VARCHAR2(128)
 GROUP_ID                                  NOT NULL VARCHAR2(30)

SQL> DESC policies
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------
 POL_NO                                    NOT NULL NUMBER
 POL_FROM                                  NOT NULL DATE
 POL_TO                                    NOT NULL DATE
 POL_VALUE                                 NOT NULL NUMBER(8,2)
 POL_SUB_TOTAL                             NOT NULL NUMBER(8,2)
 PT_CODE                                   NOT NULL VARCHAR2(8)
 DT_CODE                                            VARCHAR2(8)
 POL_TOTAL                                 NOT NULL NUMBER(8,2)
 CUST_NO                                   NOT NULL NUMBER

SQL>
```

Next, using the PL/SQL supplied package DBMS_RLS to create the CUSTOMERS table, access control policy named CUST_POLICY_VPD adapted for transactions of type SELECT, UPDATE, INSERT, and DELETE.

Creating the Application Groups and Users

The groups are created to identify users' location. Each group is associated with a specific instance service name. The below INSERT statements creates two groups. Each correspond to a service name.

```
SQL> INSERT INTO db_groups VALUES('s1e2_ma', 's1e2_ma.precisetrace.com');

1 row created.

SQL> INSERT INTO db_groups VALUES('t1e2_ny', 't1e2_ny.precisetrace.com');

1 row created.

SQL> COMMIT;

Commit complete.
```

```
SQL>
```

Then, the users are created using the format to associate them with the service network name for node 1 or node 2, but not both. For the database username 'fawzi', the application username is 'fawzi$s1e2_ma.precisetrace.com.'

```
SQL> INSERT INTO db_users VALUES
  2  ('fawzi$s1e2_ma.precisetrace.com', 'F. Alswaimil', 's1e2_ma');

1 row created.

SQL> INSERT INTO db_users VALUES
  2  ('deema$t1e2_ny.precisetrace.com', 'D. Alswaimil', 't1e2_ny');

1 row created.

SQL> COMMIT;

Commit complete.

SQL>
```

Virtual Private Database Prerequisite

There are several related virtual private policy privileges that need to be granted to certain users for managing and troubleshooting objects encompassed by virtual private database policies. The following DDL statements grant capabilities to the application owner, Oracle GoldenGate database users, and SYSTEM administrative when working with tables and views enforced by Virtual Private Database.

```
SQL> CONN / as sysdba
Connected.
SQL> GRANT EXEMPT ACCESS POLICY TO SYSTEM;

Grant succeeded.

SQL> GRANT EXEMPT ACCESS POLICY TO ggs_admin;

Grant succeeded.

SQL> GRANT EXEMPT ACCESS POLICY TO osm$repapi;

Grant succeeded.

SQL> GRANT EXECUTE ON DBMS_RLS to osm$repapi;

Grant succeeded.

SQL>
```




Create the Security Policy

Use the procedure DBMS_RLS.ADD_POLICY to create the policy CUST_POLICY_VPD. This implements fine grain row level access on the CUSTOMERS table. The STATEMENT_TYPES parameter lists the SQL application types enforced by this policy. The POLICY_FUNCTION parameter specifies the function name that returns the predicate.

```
SQL> BEGIN
  2     DBMS_RLS.ADD_POLICY(
  3        object_schema=>'system',
  4        object_name=>'customers',
  5        policy_name=>'cust_policy_vpd',
  6        function_schema=>'system',
  7        policy_function=>'ggs_security_p.cust_order',
  8        statement_types=>'select, insert, update, delete');
  9  END;
 10  /

PL/SQL procedure successfully completed.

SQL>
```

Create the Context Management Package

Develop the package to prepare the application context using the function SYS_CONTEXT and set the context attributes values using the procedure DBMS_SESSION.SET_CONTEXT. For all transaction types, the package set the value for context attribute GROUP_ID according to the USER_ID.

```
SQL> CREATE OR REPLACE PACKAGE ggs_security is
  2      PROCEDURE set_user_group;
  3  END;
  4  /

Package created.

SQL> CREATE OR REPLACE PACKAGE BODY ggs_security is
  2     PROCEDURE set_user_group(puser_id varchar2)  IS
  3        vgroup_id VARCHAR2(30);
  4     BEGIN
  5        SELECT group_id into vgroup_id
  6        FROM   db_users
  7        WHERE  UPPER(user_id) =
  8               UPPER(puser_id)||'$'||UPPER(SYS_CONTEXT('USERENV',
                  'SERVICE_NAME'));
  9        DBMS_SESSION.SET_CONTEXT('oeapp', 'group_id', vgroup_id);
 10     EXCEPTION
 11        WHEN OTHERS THEN
 12             DBMS_SESSION.SET_CONTEXT('oeapp', 'group_id', NULL);
 13     END;
 14  END;
```

```
 15  /

Package body created.

SQL>
```

Create the Security Function

Develop the function to return the predicate clause (WHERE). The return predicate is appended to the existing where clause is using AND as logical operator.

```
SQL> CREATE OR REPLACE PACKAGE ggs_security_p is
  2      FUNCTION cust_order(object_schema VARCHAR2, object_name VARCHAR2)
  3      RETURN VARCHAR2;
  4  END;
  5  /

Package created.

SQL> CREATE OR REPLACE PACKAGE BODY ggs_security_p is
  2      FUNCTION cust_order(object_schema VARCHAR2, object_name VARCHAR2)
  3      RETURN VARCHAR2 IS
  4      BEGIN
  5         RETURN 'group_id=SYS_CONTEXT(''oeapp'', ''group_id'')';
  6      END;
  7  END;
  8  /

Package body created.

SQL>
```

Create the Database Application Context

Connect as privileged user to create the context. The context references the package created above.

```
SQL> CREATE CONTEXT oeapp USING ggs_security;

Context created.

SQL>
```

Execute the Application

Now, it is time to run the application accessing tables enforced by Virtual Private Database. Depending on the database username, the context attributes assign runtime values allowing dynamic modified SQL application to be executed.



Figure 5-29 shows the original SQL statement tunnel through the security policy CUST_POLICY_VPD, the ultimate modified SQL statement restricts access only to rows that are within the context of the username session.



Figure 5-29: Dynamic SQL Modification for Data Access Control

The application starts by invoking the package that prepares and initializes the context attributes, which are used by the security function that return the predicate clause. The application heavily relies on the built-in database function SYS_CONTEXT which uses a pre-defined list of attributes such as DB_NAME and SERVICE_NAME for application runtime.

The following shows how data segmentation is achieved. It considers the combination of the database username and the service name to produce an active row set specific to the user's session. The following demonstrates the row set when connected to the Albany, NY system.

```
SQL> CONN deema/oracle@t1e2_ny
Connected.
SQL> DECLARE
  2     CURSOR c1 IS SELECT * FROM system.customers;
  3     vcust system.customers%ROWTYPE;
  4  BEGIN
  5     SYSTEM.GGS_SECURITY.SET_USER_GROUP(USER);
  6     OPEN c1;
  7     LOOP
  8        FETCH c1 into vcust;
  9        EXIT WHEN c1%NOTFOUND;
 10        DBMS_OUTPUT.PUT_LINE(vcust.cust_name);
 11     END LOOP;
 12     CLOSE c1;
 13  END;
 14  /
CAROL P. THOMPSON
MARK B. MILLER
RONALD H. ALLEN
NANCY U. WILSON
JESSICA Z. WILSON
DAVID O. JOHNSON
SANDRA J. ALSWAIMIL
SARAH N. KING
KENNETH O. ROBINSON
LINDA K. WILSON
```

```
PL/SQL procedure successfully completed.
```

The following demonstrates when the user connects to the Boston, MA system. It is the same application driven by dynamic context attributes and values to retrieve different active row sets.

```
SQL> CONN fawzi/oracle@s1e2_ma
Connected.
SQL> DECLARE
  2     CURSOR c1 IS SELECT * FROM system.customers;
  3     vcust system.customers%ROWTYPE;
  4  BEGIN
  5     SYSTEM.GGS_SECURITY.SET_USER_GROUP(USER);
  6     OPEN c1;
  7     LOOP
  8        FETCH c1 into vcust;
  9        EXIT WHEN c1%NOTFOUND;
 10        DBMS_OUTPUT.PUT_LINE(vcust.cust_name);
 11     END LOOP;
 12     CLOSE c1;
 13  END;
 14  /
JOHN V. KING
NOUR Z. LEWIS
KAREN Y. WALKER
KIMBERLY O. THOMPSON
KAREN Q. ALIBRAHIM
SARAH P. WHITE
JOHN C. DAVIS
KEVIN H. ALIBRAHIM
MARK L. MILLER
CHARLES N. WALKER

PL/SQL procedure successfully completed.

SQL>
```

Virtual Private Database Considerations

Be aware of Virtual Private Database (VPD) as an Oracle Database Enterprise feature. It is adapted mutually for data security by developing fine-grain access control at the row level and enhances application performance by reading metadata, context attribute values from Oracle shared pool memory of the System Global Area (SGA). The length of the returned predicate by the security function is limited to 4000 characters only, which may not be sufficient for some complex WHERE clauses. An alternative of DBMS_RLS is to use Oracle Enterprise Manager to create and manage policies from the graphical user interface (GUI).

Oracle GoldenGate High Availability Runtime

The following illustrates the runtime for high availability on an active-active bi-directional configuration. The building blocks for the demonstration are:

- Identical Tables Structure
- Oracle GoldenGate Active-Active Bi-Directional Topology Architecture
- Oracle GoldenGate Built-In Conflict Detection and Resolution
- Database Sequences, odd numbers for node-1 and even numbers for node-2
- Database Connections load-balancer for two-node Oracle GoldenGate high availability configuration

Figure 5-30 shows the load balancer used to handle high availability and workload balancing across the two databases, S1E2 and T1E2.

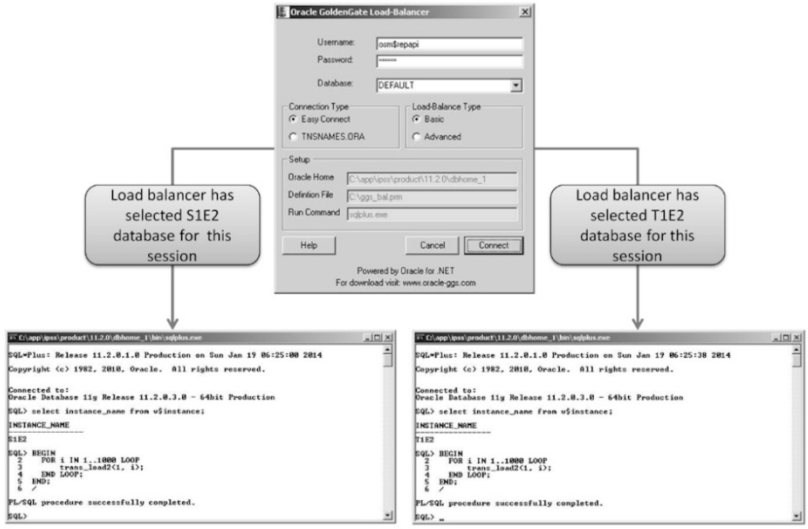


Figure 5-30: Runtime on High Availability Environment

Verify Transactions

The TRANS_LOAD2 is a workload program for generating database transactions. Oracle GoldenGate active-active bi-directional configuration synchronizes node-1 and node-2 databases, keeping the two databases identical using real-time data movement.

The test was completed successfully. The transaction size of 1000 performs concurrent 1000 inserts and replication. The logging table records conflict detection and error, if any.

Database (S1E2)	Database (T1E2)

SQL> BEGIN
  2 FOR i IN 1..1000 LOOP
  3 trans_load2(1, i);
  4 END LOOP;
  5 END
  6 /
	SQL> BEGIN
	  2 FOR i IN 1..1000 LOOP
	  3 trans_load2(1, i);
	  4 END LOOP;
	  5 END
	  6 /

SQL> SELECT cust_no
  2 FROM customers
  3 WHERE rownum < 11
  4 ORDER BY 1;
 	
  CUST_NO
---------- 	
  1006007
  1006009
  1006010
  1006011
  1006012
  1006013
  1006014
  1006015
  1006016
  1006017
 	
10 rows selected.
 	
SQL>
	SQL> SELECT cust_no
	  2 FROM customers
	  3 WHERE rownum < 11
	  4 ORDER BY 1;
	
	  CUST_NO
	---------- 	
	  1006007
	  1006009
	  1006010
	  1006011
	  1006012
	  1006013
	  1006014
	  1006015
	  1006016
	  1006017
	
	10 rows selected.
	
	SQL>

To effectively monitor data divergence and proactively detect data inconsistency, Oracle GoldenGate Veridata is a robust tool for detecting missing rows between source and target tables. Refer to Chapter 6 to learn about implementing and using Oracle GoldenGate Veridata.

Workload Simulator

Yes, Oracle GoldenGate is a non-destructive software. It is the software installable on a production environment without major negative performance impact. However, use of the workload simulator is not to be underestimated and can be used for:

- Data Replication Testing Environments
- Benchmarking the server performance for different levels of transaction volume
- Ejection of Data Conflict for detection and resolution

To download the TRANS_LOAD2 workload simulator, visit http://www.oracle-ggs.com.

Summary

The use of Oracle GoldenGate 12c for business continuity has advantages and disadvantages in using the same bi-directional topology architecture to build the foundation for a disaster recovery site and high availability on supported platforms.

When using Oracle GoldenGate for disaster recovery (DR), the major point of discussion is unsupported data types. For applications not using any of the unsupported data types, Oracle GoldenGate is an option for disaster recovery solutions. Switchover and failover from the primary system to the live standby system and vice versa is an interactive process and requires manual execution of commands or scripts.
The downtime associated with switchover and failover is very low and varies from site-to-site. The chapter has illustrated the following:

- Planned moving of users from primary to live standby system for switchover
- Planned moving of users back from live standby to primary system for switchover
- Unplanned moving of users from primary to live standby system for failover
- Unplanned moving of users back from live standby to primary system for failover

When using Oracle GoldenGate for high availability (HA), detailed analysis is carried out to evaluate Oracle GoldenGate capability of achieving the level of scalability for low-high transactions volume. When Oracle GoldenGate is deployed from the ground up, the applications become replication-aware (transparent replication) which promote Oracle GoldenGate for a high availability platform. However, deploying Oracle GoldenGate at a later stage results in the challenge of handling data conflict.
This chapter has discussed and illustrated techniques of handling data conflicts using:

- Data and Applications Segmentation
- Oracle GoldenGate Built-In Conflict Detection and Resolution (CDR)
- Develop PL/SQL Sub-Programs and implement callout using SQLEXEC
- Oracle Virtual Private Database (VPD) for total segmentation

Throughout the chapter, a two-node bi-directional topology architecture is demonstrated. A three-node topology architecture, multi-master brings challenges and should be carefully qualified for disaster recovery and high availability solutions. Oracle GoldenGate considerations, in respect of disaster recovery and high availability, are discussed to lead to a sound decision and minimize the impact of post-deployment changes.

The chapter also compared Oracle GoldenGate with Oracle Real Application Clusters (RAC) and Oracle Data Guard (DG). It was made clear, for an enterprise, Oracle GoldenGate complements RAC and DG rather than replaces them.

Oracle Corporation, World Headquarters
Oracle Parkway 500
Redwood Shores, CA 94065, USA

www.oracle.com

Worldwide Inquiries

Phone: +1.650.506.7000
Fax: +1.650.506.7200



Copyright © 2014, Oracle and/or its affiliates. All rights reserved. This document is provided for information purposes only, and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document, and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group. 0315

Connect with us

facebook.com/oracledataintegration
twitter.com/orclgoldengate
blogs.oracle.com/dataintegration
oracle.com/goto/oraclegoldengate