

ORACLE®

# Oracle Data Guard 12.2 - 18c

Uwe Kirchhoff  
Technical Architect  
ACS Global Delivery

**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 Data Guard Proposed Enhancements

- 1 Manageability
- 2 Protection
- 3 Performance
- 4 Planned Maintenance
- 5 Client Failover

# Manageability Enhancements

12c and Beyond

NEW IN  
12.2

# Oracle Data Guard - Manageability

## Standby Creation

- Create Data Guard Standbys with Database Creation Assistant (DBCA)
- RMAN DUPLICATE for Far Sync
- Enterprise Manager support for Far Sync creation
- EMCLI standby database creation

# DBCA Standby Creation

Simple and scriptable!

- At the Standby server install the Oracle Software and start a listener
- Execute a single line command

```
dbca -silent -createDuplicateDB -gdbName PROD.de.oracle.com -sid STBY  
-sysPassword oracle  
-primaryDBConnectionString linbar01:1521/PROD.de.oracle.com  
-createAsStandby -dbUniquename SPROD  
-customScripts $ORACLE_HOME/PostCR.sql
```

- Configure your network TNSNAMES for Data Guard
- Build your Broker configuration.
  - Or just add this new standby to your existing Broker configuration

## RMAN and Far Sync

- RMAN will be able to create a FAR SYNC Instance using DUPLICATE
  - duplicate target database for farsync from active database;
  - duplicate target database for farsync backup ...
- Same basic setup as the standby active duplicate process then...

```
connect target sys@PROD;
connect auxiliary sys@PROD1FS;
run {allocate channel prmy1 type disk;
      allocate auxiliary channel stby1 type disk;
      duplicate target database for farsync from active database
      spfile
      parameter_value_convert ('PROD','SPROD')
      set 'db_unique_name'='SPROD';}
```



# Enterprise Manager Cloud Control

## Far Sync Creation and Management

- You will also be able to create Far Sync Instances using a new, but familiar EMCC creation Wizard

**ORACLE** Enterprise Manager Cloud Control 13c

standby

Oracle Database Performance Availability Security Schema Administration

Data Guard  
Page Refreshed October 7, 2015 10:41:38 AM IST

View Data | Real Time: Manual Refresh

**Overview**

Data Guard Status: Normal  
Protection Mode: Maximum Performance  
Fast-Start Fallover: Enabled to dbprim2\_slc04idx.us.oracle.com  
Observer Location: slc04idx

**Primary Database**

Name: standby  
Host: slc04idx  
Data Guard Status: Normal  
Current Log: 169  
Properties: Edit

**Standby Database Progress Summary**

Transport lag is the time difference between the last update on the primary database and the last received redo on the standby database. Apply lag is the time difference between the last update on the primary database and the last applied redo on the standby database.

seconds

dbprim2\_slc04idx.us.oracle.com

Transport Lag  
Apply Lag

**Standby Databases**

[Add Far Sync](#) [Add Standby Database](#)

Select	Name	Host	Data Guard Status	Role	Redo Source	Real-time Query	Last Received Log	Last Applied Log	Estimated Fallover Time
<input checked="" type="radio"/>	dbprim2	slc04idx	Normal	Physical Standby	standby	Enabled	168	168	< 1 second

**Performance**  
Data Guard Performance  
Log File Details

**Additional Administration**  
Verify Configuration  
Remove Data Guard Configuration

# Enterprise Manager Cloud Control

## Far Sync Creation and Management

**ORACLE Enterprise Manager Cloud Control 13c**

standby

Oracle Database Performance Availability Security Schema Administration

**Data Guard**  
Page Refreshed October 7, 2015 10:41:38 AM IST

View Data | Real Time: Manual Refresh

**Overview**

- Data Guard Status: Normal
- Protection Mode: Maximum Performance
- Fast-Start Failover: Enabled to dbprim2\_slc04idx.us.oracle.com
- Observer Location: slc04idx

**Primary Database**

- Name: standby
- Host: slc04idx
- Data Guard Status: Normal
- Current Log: 169
- Properties: Edit

**Standby Database Progress Summary**

Transport lag is the time difference between the last update on the primary database and the last received redo on the standby database. Apply lag is the time difference between the last update on the primary database and the last applied redo on the standby database.

seconds

dbprim2\_slc04idx.us.oracle.com

Legend: Transport Lag, Apply Lag

**Standby Databases**

Select	Name	Host	Data Guard Status	Role	Redo Source	Real-time Query	Last Received Log	Last Applied Log	Estimated Fallover Time
<input checked="" type="radio"/>	dbprim2	slc04idx	Normal	Physical Standby	standby	Enabled	168	168	< 1 second

**Performance**  
Data Guard Performance  
Log File Details

**Additional Administration**  
Verify Configuration  
Remove Data Guard Configuration

**Buttons:** Add Far Sync, Add Standby Database



# Enterprise Manager Cloud Control

## Command line standby creation

- Enterprise Manager has had a Standby Creation Wizard for many versions
- It also has a command line interface called EMCLI
  - Today (<12.2) you cannot script standby creation through it.
  - But not for long!

```
emcli create_standby -source_db_target_name="PROD"  
-source_db_target_type="rac_database"  
-dest_oracle_sid="PROD1_1"  
-spname="PROD1"  
-use_broker
```

- Notice how the target can be a RAC or Single Instance database
- RMAN duplicate from active database is also the default

NEW IN  
12.2

# Oracle Data Guard - Manageability

## Ease of Use

- RESTful Services Interface
  - Creating, Monitoring and Managing a Data Guard Broker Configuration
- Oracle Data Guard for Data Warehouses - No Logging support
- Subset Standbys
- Automatically Synchronize Password Files
- TDE Tablespace Offline Conversion

## Data Guard and No Force Logging

- No-Logging operations only generate enough redo to tell the standby that some blocks are now unrecoverable.
  - Repairing them meant restoring the complete affected data files.
- No-Logging blocks will be tracked by the Primary database
- Standbys can then be easily repaired using RMAN
  - Validation and repair only of blocks on standby that were invalidated by non-logged operations (such as direct loads) on the primary

```
rman validate/recover .. nonlogged block
```

- This will query the Primary for block locations that contain non-logged blocks and repair the standby database.

NEW IN  
12.2

# Removing the headache of Password file management

## Password, Password, who's got the right password?

- Today(<12.2) when you change the password of any elevated privilege user
  - You must copy the password file to all other nodes of the Primary AND all standbys
  - Made a little simpler in 12.1.0.1 by a shared password file for RAC
- In Oracle Database 12c Release 2
  - Changing elevated privileged passwords on the primary automatically updates all the standbys' password files.
  - No need to manually copy password files!
- New DG redo authentication protocol that uses SSL certificate for 'redo\_transport\_user'

# TDE Tablespace Conversion

## < 12.2

- Requires an Export and Import to encrypt
- Cannot be done online so data is unavailable during the process
- Outage for the data can be mitigated by using a Transient Logical Standby
  - Convert a Physical Standby to a Transient Logical Standby
    - Export the data, Create the encrypted tablespace, Import the data at the transient logical standby
  - Switchover
  - At <http://www.oracle.com/goto/maa> see these two papers
    - Converting to Transparent Data Encryption Using Active Data Guard 12c (DBMS\_ROLLING)
    - Converting to Transparent Data Encryption Using Data Guard Transient Logical Standby with Oracle Database 11g

# TDE Tablespace Conversion

(>=12.2)

- Offline data file method still available
  - Complete with Physical Standby method

- A new SQL command will arrive on the scene!

```
alter tablespace users encryption encrypt
```

- Encrypts all data files in the tablespace online.
  - Primary only & Requires auxiliary disk space
- Can also be done OFFLINE for better performance
  - No Extra storage, but requires data outage on Primary.
  - Offline can be done on a Physical Standby first as with the previous example



# Oracle Data Guard - Manageability

## Broker General Enhancements

- DGMGRL Command Improvements
- Broker Configuration-Wide Service Name
- Simplified Observer Management for multiple Fast-Start Failover configurations

## New DGMGRL commands

- Enhanced Scripting and Execution
  - Script execution via '@'
  - Repeat previous DGMGRL command with '/'
  - Execute operating system command with 'HOST' or '!'
  - Store output in a file with 'SPOOL'
  - Display timestamp in prompt via 'SET TIME ON/OFF'

## Configuration-Wide Service Name

- Provides a single automatically started service to connect to any database, Primary or Standby, in a Data Guard Broker configuration.
  - Defaults to Primary DB Unique name with ‘\_CFG’ appended.
  - Required to implement the Simplified Observer management (Next slide)
- Can be set by the user with:

```
edit configuration set property  
ConfigurationWideServiceName = 'PROD_CFG';
```

- And seen with:

```
show configuration verbose ConfigurationWideServiceName;  
ConfigurationWideServiceName = 'PROD_CFG'
```

## Simplified Observer Management

- Today (<12.2) when you have multiple Broker configurations using Fast-Start Failover you must manage the Observers separately even if they all run on the same 'Observer' system.
- The Observer also runs in the foreground
- Let's fix that last one first

```
start observer PRODObserver logfile is '/tmp/PRODObserver'  
in background connect identifier is PROD;
```

- That settles that
- But what about multiple Fast-Start Failover configurations?

# Simplified Observer Management

## One Command to Run and Manage Multiple Observers

- Create an Observer Configuration file (here we have 4 Observers, 2 groups)

```
BROKER_CONFIGS = (  
  (CONFIG = (NAME=SALES) (CONNECT_ID=SALES_P) (FILES=/home))  
  (CONFIG = (NAME=HR) (CONNECT_ID=HR_P) (FILES=/home))  
  (CONFIG = (NAME=CUSTOMER) (CONNECT_ID=CUSTOMER_P) (FILES=/home))  
  (CONFIG = (NAME=ORDERS) (CONNECT_ID=ORDERS_P) (FILES=/home)))  
CONFIG_GROUPS = (  
  (GROUP = (NAME=GRP_A )  
    (CONFIG_LIST = (NAME=SALES) (NAME=ORDERS)))  
  (GROUP = (NAME=GRP_B)  
    (CONFIG_LIST = (NAME=HR) (NAME=CUSTOMER))))
```

- The CONNECT\_ID must point to the Primary and all potential FSFO Targets

# Simplified Observer Management

## One Command to Run and Manage Multiple Observers

- Set the Property to point to your file on the Observer system

```
Oracle12c > dgmgrl
DGMGRL for Linux: Release 12.2.0.0.0
Copyright (c) 1982, 2015, Oracle and/or its affiliates.
All rights reserved.
Welcome to DGMGRL, type "help" for information.

DGMGRL> set observerconfigfile='/home/myobservers.ora';
DGMGRL> show observerconfigfile;
ObserverConfigFile=/home/myobservers.ora
```

- Notice that there is no need to connect to any database to set this up

# Simplified Observer Management

## One Command to Run and Manage Multiple Observers

- Once the file is set up instruct the Broker to run the Observer groups

```
DGMGRL> start observing grp_a ;  
ObserverConfigFile=/home/myobservers.ora  
Observer configuration file parsing succeeded  
Submitted command "START OBSERVER" using connect identifier "SALES_P"  
Submitted command "START OBSERVER" using connect identifier "HR_P"
```

Check superobserver.log and individual observer logs

- There are no passwords in the configuration file
- You must use the Wallet for authentication on all databases involved.

# Protection Enhancements

**12c and Beyond**



NEW IN  
12.2

# Oracle Data Guard - Protection

## Enhanced Protection

- Enhanced Automatic Block Recovery
- Zero-data loss Fast-start Failover upon storage failure even in async mode
  - If instance fails, ships all redo data accumulated in log buffer to standby
- Oracle Data Guard Database Compare
- Enhanced Alternate Destinations
- Broker Protection Enhancements
- Disaster Recovery in the Cloud

# Enhanced Automatic Block Recovery

- Active Data Guard Automatic Block Repair introduced in 11g Release 2



- Blocks like file headers etc were not repaired.
- In **Oracle Database 12c Release 2** the types of corrupted blocks that can be repaired increases!

NEW IN  
12.2

## Zero-Data Loss (ZDL) Fast-Start Failover

### Enhanced for asynchronous configurations

- Today, ZDL Fast-Start Failovers are only possible in Synchronous configurations running in Maximum Availability
  - Many times though, failures are storage related and not complete system failures
- In **Oracle Database 12c Release 2**, a storage failure at the primary database will be handled by the Oracle Data Guard Asynchronous process.
  - If the instance fails, it will ship all redo data accumulated in log buffer to standby
  - This will allow Fast-Start Failover to perform a Zero Data Loss failover
    - Even though the configuration was in Maximum Performance
  - Similar to an Automatic In-Memory 'ALTER DATABASE FLUSH REDO;' at failure time

NEW IN  
12.2

## Oracle Data Guard Database Compare

- High speed database compare between primary and standby
  - Detects latent corruptions on blocks that have not been touched by Oracle
    - Useful in detecting latent corruptions, lost writes
    - Fast compare at physical level, accounts for blocks being modified
    - Network & compute efficiency
      - Send & compare blocks in “batches”
      - Optimizes for blocks where versions are the same
      - Ignores empty blocks

```
DGMGRL> validate database PROD1 datafile 10 output=checkPROD1;  
Operation requires a connection to database "PROD1"  
Connecting ...  
Output files are created in /.../PROD1/trace on host "GRONAU1"
```

# Oracle Data Guard Database Compare

```
Summary:
*****
ID: Block Type Id
TOTAL: Total number of blocks found
DIFFV: Number of block pairs with different version
LWLOC: Lost Writes at Local
LWRMT: Lost Writes at Remote
SAMEV: Number of block pairs with same version
SAMEV&C: Number of block pairs with same version and checksum
DIFFPAIR: Number of block pairs with same version but different contents
ENCERR: Undecided blocks related to encryption/decryption error.
        e.g. Wallet is not open.
SKIPPED: Skipped blocks due to data corruption, etc
```

ID	TOTAL	DIFFV	LWLOC	LWRMT	SAMEV	SAMEV&C	DIFFPAIR	ENCERR	SKIPPED
29	0000001	0000000	0000000	0000000	0000001	0000001	0000000	0000000	0000000
30	0000125	0000000	0000000	0000000	0000125	0000125	0000000	0000000	0000000
58	0000512	0000512	0000000	0000000	0000000	0000000	0000000	0000000	0000000

## Enhanced Alternate Destinations

- Alternate destinations got much better in 12.1
  - But failback to the first destination happened whether or not you wanted it
  - And you lost failback if you configured more than one alternate.
- So in 12.2 we threw away the ALTERNATE attribute of redo transport
  - Replacing it with 2 new attributes - GROUP and PRIORITY
    - GROUP says we are all potential alternates of each other in the same group
    - PRIORITY tells Data Guard when to failback to the original target between groups
- Did you get all that?
  - I didn't think so, this is my give you a headache feature.
  - Let's look at an example

## Improved Alternate Destinations

- Our configuration
  - We have a Primary with two Standbys (TS1 and TS2)
    - They could be one local and one remote or both remote or both local
      - But with Far Sync we ship once from the Primary and get ZDL at both if necessary
  - We also have 2 Far Sync instances that can service the two Standbys (FS1, FS2)
    - Far Sync 1 & 2 are of equal configuration and both local (within SYNC latency)
      - So we can lose one and the other will take over and maintain Maximum Availability
    - But if both Far Sync Instances are down we still want redo to be shipped to TS1 & TS2
    - So we set our 4 “LOG\_ARCHIVE\_DEST\_n” parameters as follows (all valid for the Primary role/logfile)

```
2:SERVICE=FS1 SYNC GROUP=1 PRIORITY=1  
3:SERVICE=FS2 SYNC GROUP=1 PRIORITY=1  
4:SERVICE=TS1 ASYNC GROUP=1 PRIORITY=8  
5:SERVICE=TS2 ASYNC GROUP=1 PRIORITY=8
```

- All clear?

## Improved Alternate Destinations

- OK, some English to go with the parameters, these are all the same group

```
2:SERVICE=FS1 SYNC GROUP=1 PRIORITY=1
3:SERVICE=FS2 SYNC GROUP=1 PRIORITY=1
4:SERVICE=TS1 ASYNC GROUP=1 PRIORITY=8
5:SERVICE=TS2 ASYNC GROUP=1 PRIORITY=8
```

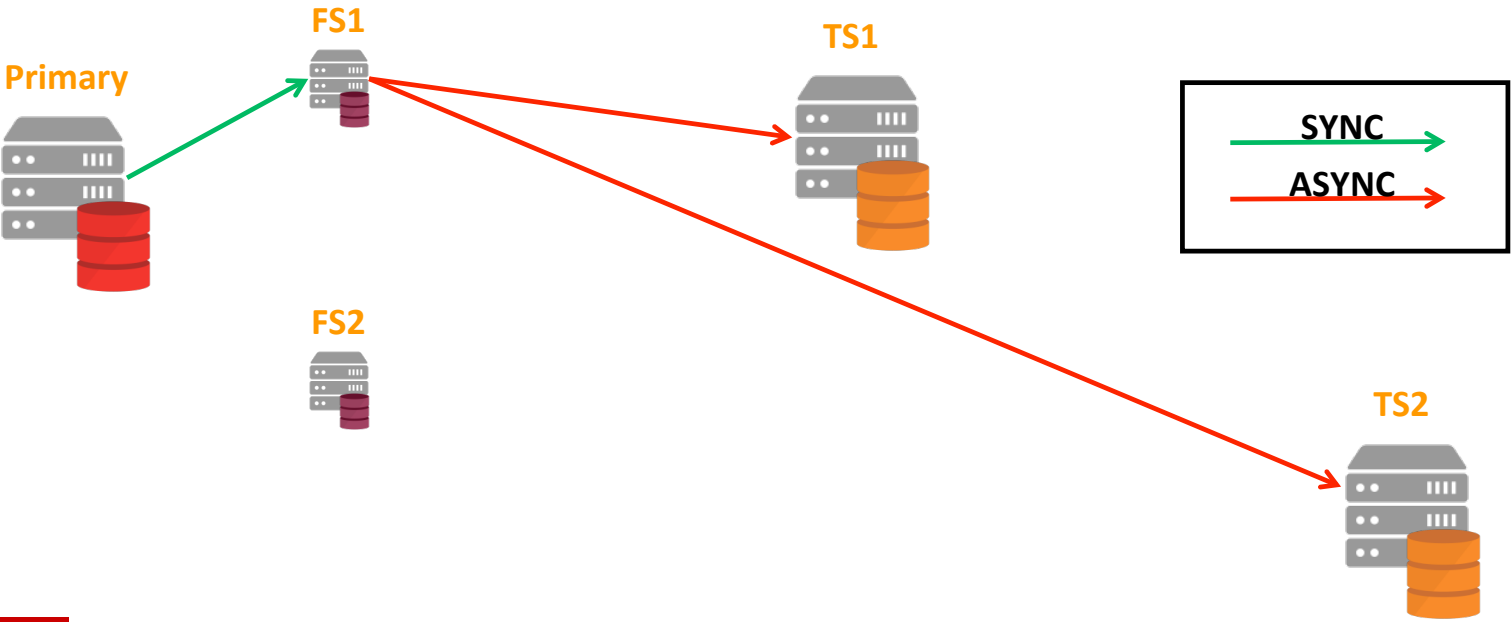
- But the Priority says
  - Send Redo to FS1 first, it forwards the redo to TS1 & TS2.
  - If FS1 fails start sending to FS2 who forwards the redo to TS1 & TS2
    - If FS1 comes back do nothing, stay with FS2
    - However if FS2 fails and FS1 is back, revert to FS1
  - If both FS1 and FS2 fail then send directly to TS1 and TS2.
    - Since they are configured as ASYNC the Protection Level drops to RESYNCHRONIZATION
    - After that if either FS1 or FS2 become available revert back to them
      - Protection Level goes back to Maximum Availability



# Improved Alternate Destinations

How about a picture?

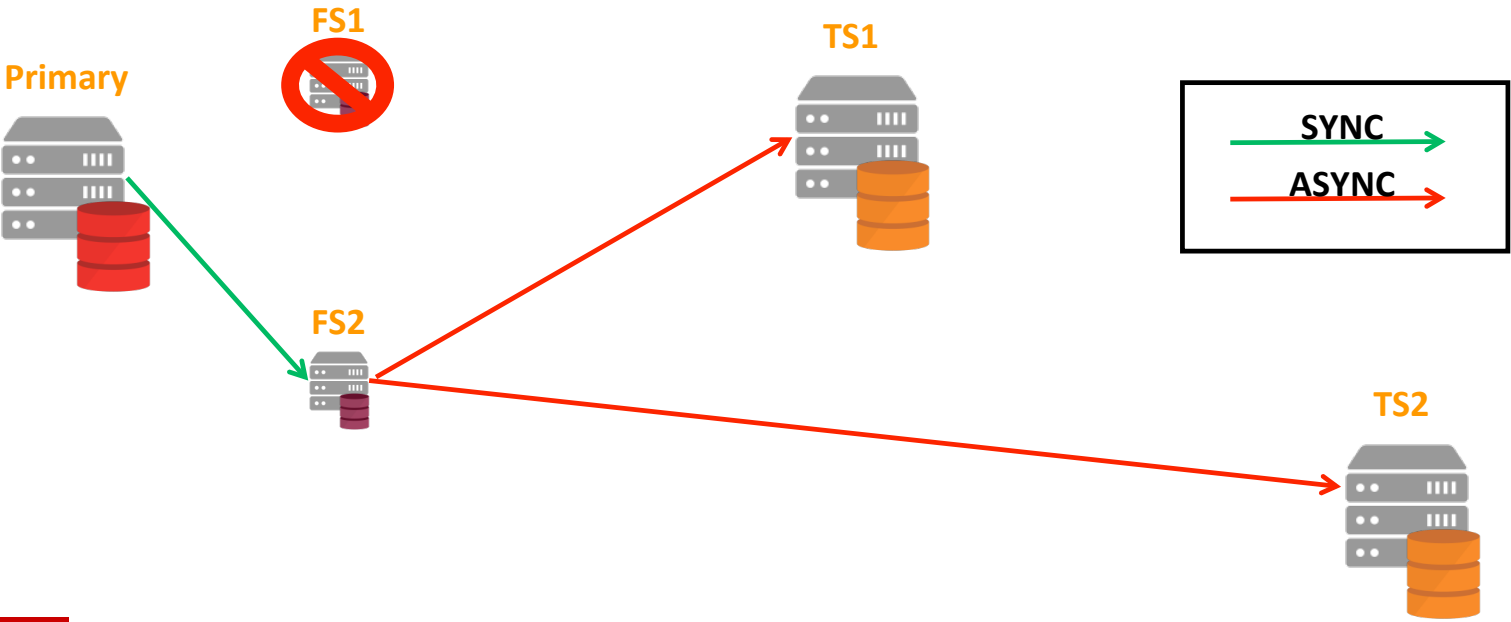
- Normal Steady State



# Improved Alternate Destinations

How about a picture?

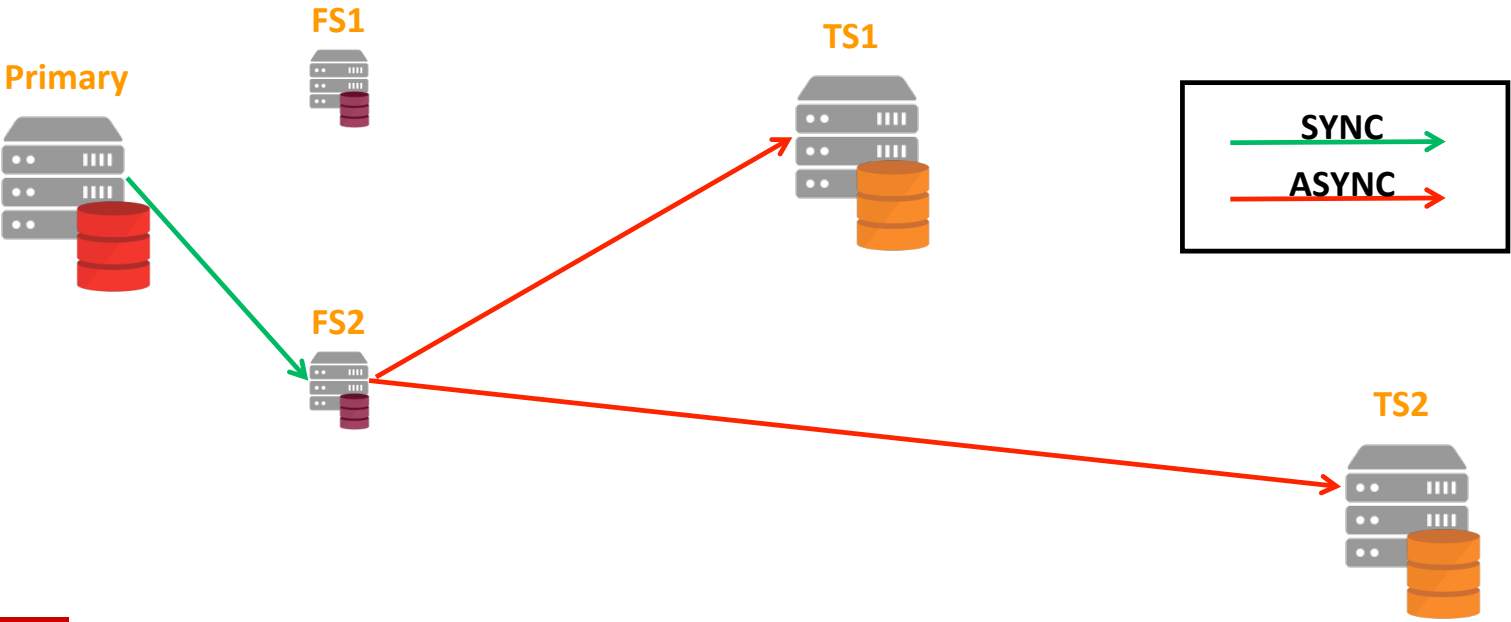
- Far Sync FS1 Fails (Stay in Priority 1)



# Improved Alternate Destinations

How about a picture?

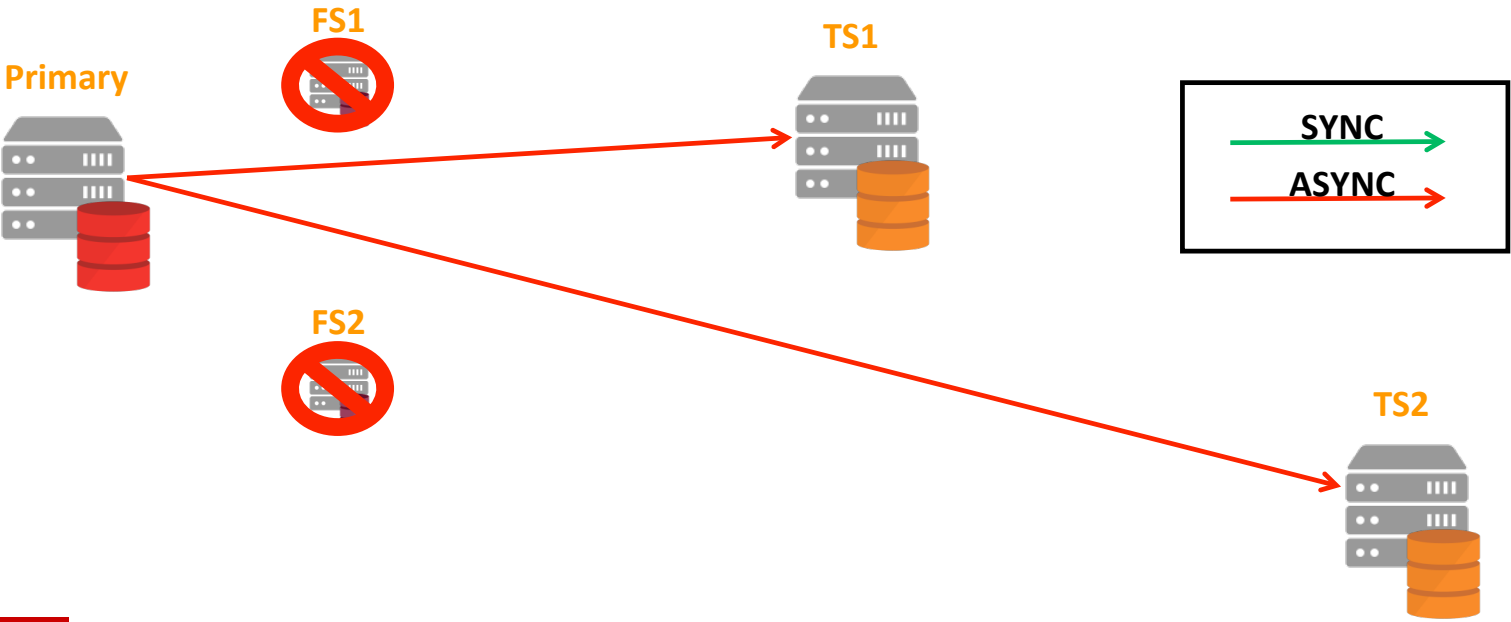
- Far Sync FS1 comes back (Stay in Priority 1, No failback)



# Improved Alternate Destinations

How about a picture?

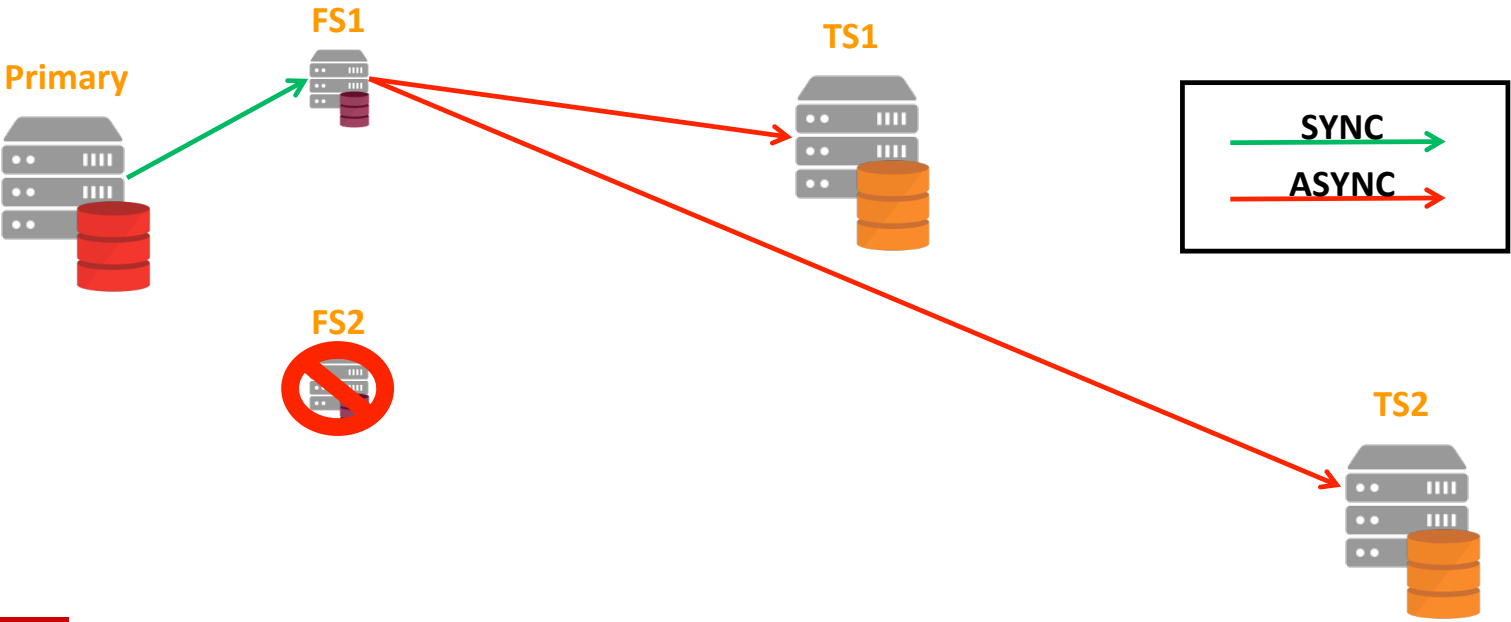
- Far Sync FS2 Fails and FS1 is still not available (Move to Priority 8)



# Improved Alternate Destinations

How about a picture?

- Far Sync FS1 comes back (Move back to Priority 1)



# Oracle Data Guard - Protection

## Broker Protection Improvements

- Block Comparison Tool Support in DGMGRL
  - Shown in the earlier example
- Multiple Automatic Failover Targets
- Multiple Observers
- Fast-Start Failover in Maximum Protection Mode
- Support for Enhanced Alternate Destination
- Support for Transport Destinations of Different Endianness
  - Cross-endian Data Guard: AIX to Recovery Appliance

# Performance Enhancements

**12c and Beyond**

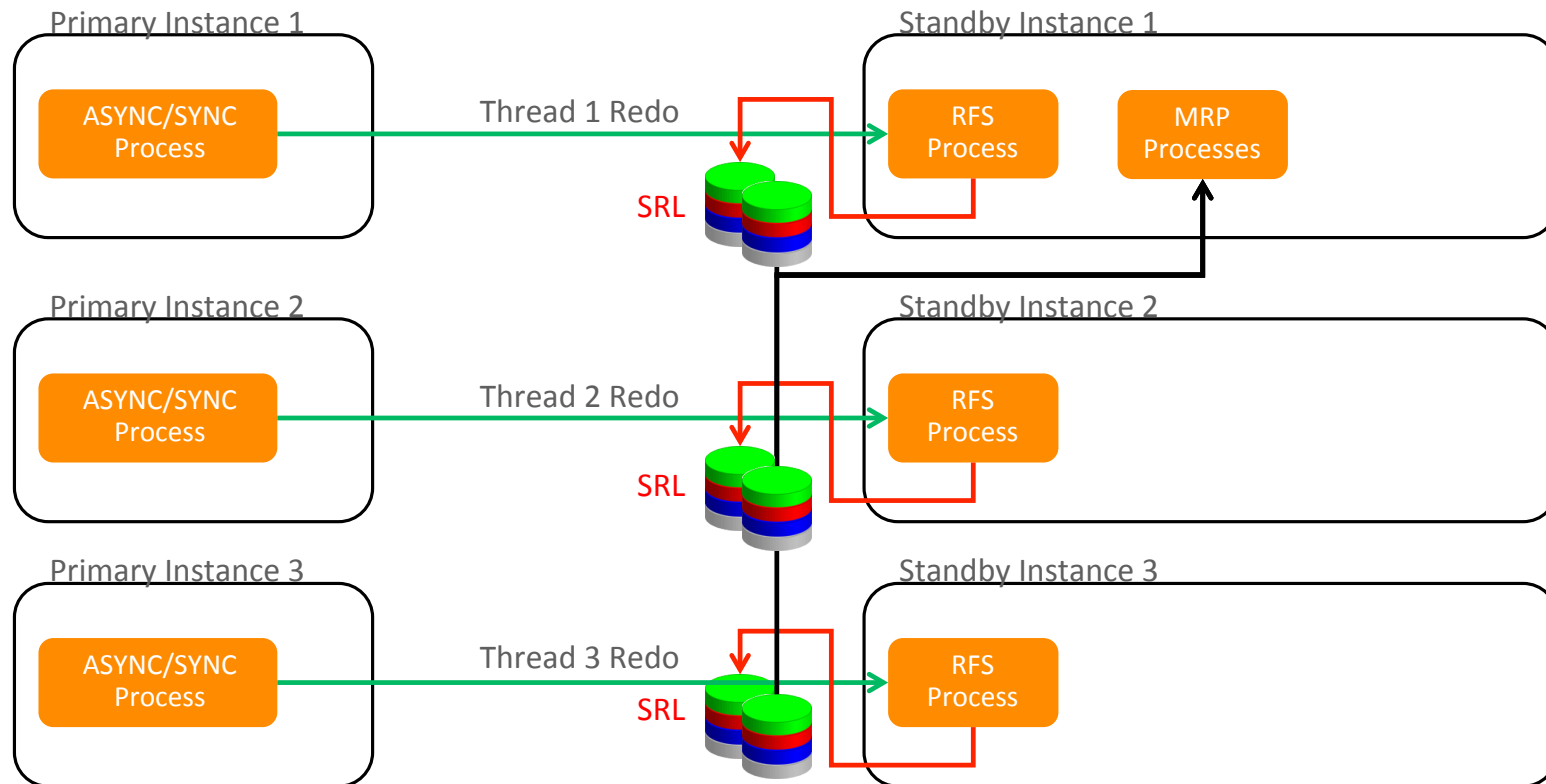
# Oracle Data Guard - Performance

## Performance and Tuning Enhancements

- Multi-Instance Redo Apply
  - Oracle Data Guard Broker Support
- Database In-Memory on an Oracle Active Data Guard Standby Database
- Active Data Guard support for Diagnostic Pack features
- Active Data Guard Support for SQL Tuning Advisor

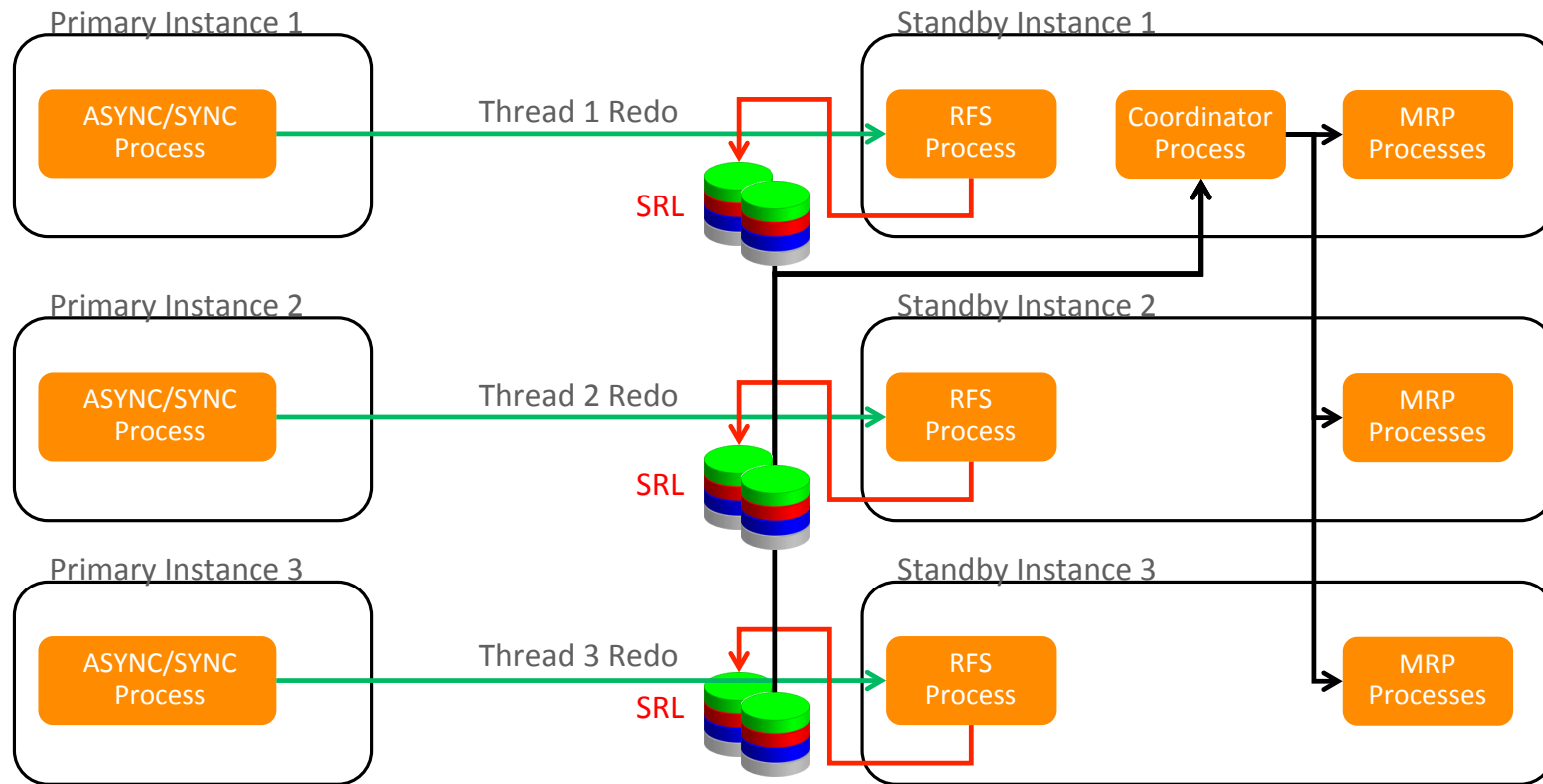


# Single-Instance Redo Apply



# Multi-Instance Redo Apply

NEW IN  
12.2



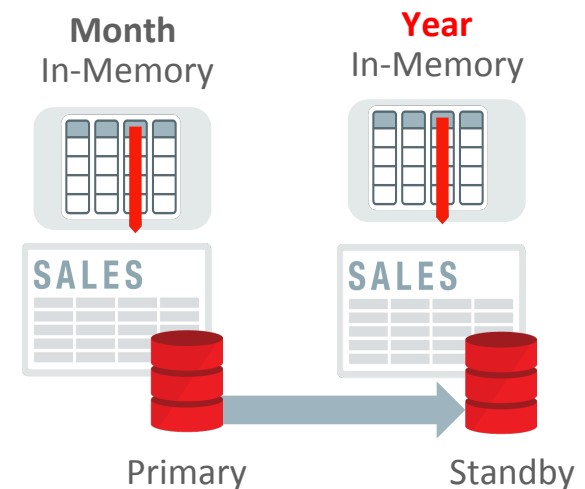
NEW IN  
12.2

## Multi-Instance Redo Apply

- Parallel, multi-instance recovery : standby will keep up
  - Standby recovery - utilizes CPU and IO across all nodes of RAC standby
    - `recover managed standby database disconnect using instances 4;`
  - Have seen 3500MB+/sec apply rate on an 8 node RAC
- Multi-Instance Apply runs on all MOUNTED instances or all OPEN Instances
- Exposed in the Broker with the '**ApplyInstances**' property on a standby database

# Database In-Memory and Active Data Guard

- In-memory DB on **Active** Data Guard
  - Create IMC tables and columns for analytics on Active Data Guard
  - Can populate with different data than production database
- Offload even more to your standby!



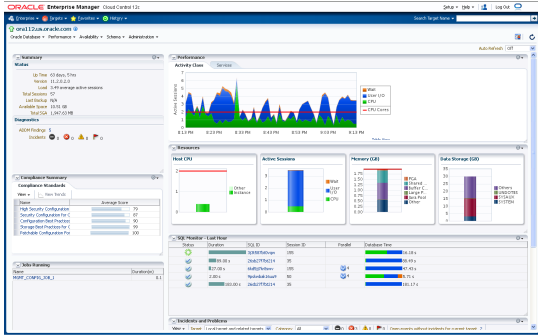
**NEW IN  
12.2**

# Diagnostics and Tuning for Active Data Guard

- Support for Diagnostic Pack (AWR)

Statistic Name	Time (s)	% of DB Time
sql execute elapsed time	37,269.55	98.97
DB CPU	24,419.87	64.84
parse time elapsed	822.64	2.18
hard parse elapsed time	738.97	1.96
PL/SQL execution elapsed time	442.38	1.17
hard parse (sharing criteria) elapsed time	99.67	0.26
PL/SQL compilation elapsed time	10.58	0.03
sequence load elapsed time	5.78	0.02
repeated bind elapsed time	1.19	0.00
failed parse elapsed time	0.48	0.00
connection management call elapsed time	0.34	0.00
hard parse (bind mismatch) elapsed time	0.10	0.00
DB time	37,659.17	
background elapsed time	3,874.17	
background cpu time	2,586.88	

- Support for Tuning Pack features and SQL Plan Analyzer



NEW IN  
12.2

# AWR Support for Remote Snapshots

## On an Active Data Guard Standby

- In Oracle Database 12c Release 2, the AWR framework is enhanced to support capture of remote snapshots from any generic database including Active Data Guard (ADG) databases.
  - A target catalog database collects snapshots from the remote databases (sources)
  - Snapshots can be collected automatically or manually
  - AWR tables on the catalog database accumulate snapshot data from all sources via database links
  - Source databases must be registered on the catalog via new `DBMS_WORKLOAD_REPOSITORY.REGISTER_REMOTE_DATABASE` API

## SQL Tuning Advisor Support for Active Data Guard

- All SQL Tuning Advisor tasks issued at the standby
  - Create tuning task, execute tuning task and implement SQL Tuning Advisor recommendations
  - Test execution (heavy lifting) happens on standby
    - Only minimal write related activity on primary
- Required data fetched from primary over a database link from standby
- Task details and tuning results are stored at primary and the essential data required to construct the report is accessed remotely from primary
- The report is constructed locally at the standby, with no CPU overhead in primary

## Feature list

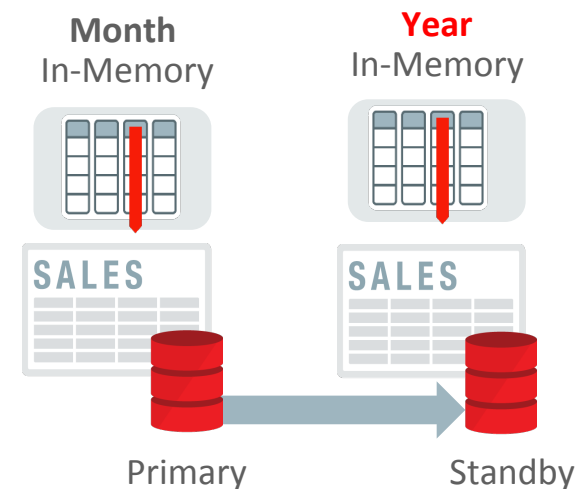
- **Faster query response with faster redo apply**
  - Multi-Instance Redo Apply on ADG now supports In-Memory Columnar & Block-change tracking
- **Extend ADG environments for Data Warehousing**
  - ADG support for Nologging workloads
- **Faster application response after ADG role transition**
  - Preserve Buffer Cache on ADG after role transition
- **Protect Primary database from lost write storage failures**
  - New Lost write detection capability using shadow tablespaces at Primary



## Multi-Instance Redo Apply enhancements

- Multi-Instance Redo Apply allows all standby nodes to participate in recovery
- In-memory DB (IMC) on **Active** Data Guard allows: \*
  - Creation of IMC tables and columns for analytics on Active Data Guard
  - Population with different data than production database
  - Offloading even more to your standby!
- Multi-Instance Redo apply also works with BCT

\* Available only on Exadata and Oracle Cloud Offerings



## Data Guard and No Force Logging\*

**\*Available on Engineered Systems and Oracle Cloud only**

- Extended to provide better support in an Active Data Guard environment without significantly increasing the amount of redo generated.
- Two new modes are added as alternatives to the existing nologging mode
  - Standby Nologging for **Load Performance**
    - Ensures that standbys will receive the nonlogged data changes with the minimum impact to the speed of loading at the primary
      - The standby can transiently have nonlogged blocks. These nonlogged blocks will be automatically resolved by managed standby recovery.
  - Standby Nologging for **Data availability**
    - Ensures all standbys have the data when the primary load commits but at the cost of throttling the speed of loading data at the primary
      - The standbys will never have any nonlogged blocks.

## Creating GTTs on Active Data Guard

- Creating GTTs today will fail when executed on Active Data Guard
- In 18.1.0.0 `CREATE GLOBAL TEMPORARY TABLE` on Active Data Guard will:
  - Create the GTT on the Primary
  - Wait for the GTT to be replicated and applied to the Active Data Guard standby
  - Return control to the user

# Executing DML on Active Data Guard

## Restricted support for DML Re-direction

- DML Re-direction automatically performed from Active Data Guard standby to the Primary without compromising ACID
  - Enabled by setting parameter “`_enable_proxy_adg_redirect=TRUE`” and appropriate connectivity parameters
    - Standby ADG session waits for updates to show up on ADG via redo apply
    - DML data re-directed by an ADG session is only visible to that session before commit
    - DML data re-directed is visible to all sessions after ADG session commits
  - Respects ACID properties of a database for ADG session
  - Targeted for “Read Mostly **Occasional Updates**” applications

## New Broker DGMGRL commands

- Enhanced Scripting and Execution
  - SET ECHO|DEBUG ON|OFF
  - SHOW ALL
    - Shows current DGMGRL settings

## New Broker VALIDATE commands

- VALIDATE DATABASE SPFILE
  - Validate parameters settings between the Primary and a Standby database.
- VALIDATE NETWORK CONFIG
  - Validate the network setup between all databases in the configuration
- VALIDATE STATIC CONNECT IDENTIFIER
  - Verify the static connect identifier setup in Non-CRS databases

# Q & A



## Safe Harbor Statement

The preceding 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.



# Integrated Cloud

## Applications & Platform Services

ORACLE®