ORACLE
Cloud Infrastructure

Hybrid Data Guard to ExaCC

Production Database on Premises and Disaster Recovery on Exadata Cloud @ Customer

ORACLE WHITE PAPER | SEPTEMBER 2019

ORACLE

Introduction

Oracle Maximum Availability Architecture (MAA) is Oracle's best practices blueprint based on proven Oracle high availability technologies, end-to-end validation, expert recommendations and customer experiences. This paper provides best practices for configuring and maintaining hybrid Data Guard configuration with Exadata on-premises and Exadata Cloud@Customer or ExaCC. This paper assumes a basic understanding of the Exadata Cloud at Customer configuration.

Data Guard and Active Data Guard provide disaster recovery (DR) for databases with recovery time objectives (RTO) that cannot be met by restoring from backup. Customers use these solutions to deploy one or more synchronized replicas (standby databases) of a database (the primary database) to provide high availability, comprehensive data protection, and disaster recovery.

An effective disaster recovery plan can be costly due to the need to establish, equip and manage a remote data center. The Oracle Cloud at Customer offers a great alternative for hosting standby databases for customers to reduce the cost or complexity of managing Exadata hardware and infrastructure. Existing production databases remain on-premises and standby databases used for DR are deployed on the Exadata Cloud@Customer. This mode of deployment is commonly referred to as a hybrid cloud implementation.

Disaster Recovery to the Cloud with Data Guard and Active Data Guard

Active Data Guard is included with the Cloud at Customer PaaS license and extends Data Guard capabilities by providing advanced features for data protection and availability as well as offloading read-only workload and fast incremental backups from a production database. From on-prem licensing perspective, Active Data Guard is included in the Extreme Performance Edition. When used in a hybrid configuration, Active Data Guard must also be licensed for the on-premises system.

Oracle Maximum Availability Architecture recommends:

1. ExaCC target system is symmetric or similar to the on-premises Exadata system to ensure you meet the same performance SLAs after a role transition
2. Ensure network bandwidth is enough to handle peak redo rates
3. Ensure network reliability and security between on-premises to and from ExaCC.
4. Use Active Data Guard for additional auto-block repair, data protection and offloading benefits

Benefits of Hybrid Standby in the Cloud

1. Cloud data center and infrastructure is managed by Oracle
2. Cloud provides basic system life cycle operations including bursting and shrinking of compute resources of the VM hosting the Standby.
3. Oracle Data Guard provides disaster recovery, data protection and ability to offload activity for higher utilization and return on investment.
4. When configured with MAA practices and using Data Guard Fast-Start failover, customers can achieve Recovery Target Objective (RTO) of seconds and Recovery Point Objective (RPO or potential data loss) less than a second with ASYNC transport or zero with SYNC or FAR SYNC transport configurations.

Using Standby Database to reduce downtime during Planned Maintenance

There are several options for utilizing a standby database on the cloud for reducing planned downtime of the primary production database:

Standby-first Patch Apply (This colored section are benefits, however title says Database, OS Requirements and Network Prerequisites. This will flow better if the benefits were moved out to a benefits section)

Many patches may be applied first to a physical standby for thorough validation. Customers who wish to minimize downtime will frequently patch the standby first, then switch production to the standby database, and then patch the original primary. If the primary and standby are RAC and the software update is RAC rolling, a switchover is not required; however, it is still recommended to update the software on the standby-first for additional validation and protection. Data Guard physical replication is supported between primary and standby running at mixed patch versions for patches that are standby-first eligible. This is documented in the patch readme. The customer may also choose to run for a period of time with mixed patch versions between primary and standby to enable fast fallback to the unpatched version should there be any unanticipated problems with the patch. See My Oracle Support Note 1265700.1, “Oracle Patch Assurance - Data Guard Standby-First Patch Apply” for more details on patches eligible for the standby-first process.

Database Rolling Upgrade

Another beneficial Data Guard use case is database rolling upgrade with transient logical standby or DBMS_Rolling solution when standby-first solution is not applicable. The transient logical process is used in Oracle 11g and Oracle 12c to temporarily convert a physical standby database to a logical standby, upgrade the logical standby to the new version, validate and when ready execute a Data Guard switchover. After the switchover completes, the original primary database is converted to a synchronized physical standby also operating at the new release. Refer to Oracle 11g Database Rolling Upgrades Made Easy or Oracle 12c DBMS_Rolling for more information. A more efficient database rolling upgrade process using the standby database exists for Data Guard environments 12.2 and higher. Refer to Using DBMS_ROLLING to Perform a Rolling Upgrade section in the Data Guard documentation.

Data Guard Life Cycle Management e.g. switchover, failover and reinstate is a manual process in Hybrid Data Guard configurations. There is no cloud console support for instantiation, management or maintenance.

Service Level Requirements

Hybrid cloud deployments are by definition user-managed environments. The administrator must determine service level expectations for availability, data protection, and performance that are practical for a given configuration and application. Service Levels must be established for each of three dimensions relevant to disaster recovery that are applicable to any Data Guard configuration:

- **Recovery Time Objective (RTO)** describes the maximum acceptable downtime should an outage occur. This includes the time required to detect the outage and to failover both the database and application connections so that service is resumed.
- **Recovery Point Objective (RPO)** describes the maximum amount of data loss that can be tolerated. Achieving the desired RPO depends upon:
  - Available bandwidth relative to network volume.
  - The ability of the network to provide reliable, uninterrupted transmission.
  - The Data Guard transport method used: either asynchronous for near-zero data loss protection or synchronous for zero data loss protection.
- **Data Protection:** With Active Data Guard and MAA configuration, customers can configure the most comprehensive block corruption detection, prevention and auto-repair.
- **Performance:** Database response time may be different after failover if less capacity – compute, memory, I/O, etc, are provisioned at the standby system than in the on-premises production system. This occurs when administrators purposefully under-configure standby resources to reduce cost; accepting reduced service level while in DR mode. MAA best practices recommend configuring symmetrical capacity at both primary and standby so there is no change in response time after failover. Compute bursting available with the cloud can enable a middle ground where there is less capacity deployed steady-state, but the new primary is rapidly scaled-up should a failover be required.

Security Requirements

Oracle MAA best practice recommends using Oracle Transparent Data Encryption (TDE) to encrypt both primary and standby databases to ensure all data is encrypted at-rest. This requires customers to have TDE license for the primary database. If you have an advanced security option for your on-prem, that includes TDE. TDE license is already included with ExaCC. Data can be converted during the instantiation process but it's highly recommended to convert to TDE prior to migration to provide the most secure Data Guard environment. Refer to Oracle Database Tablespace Encryption Behavior in Oracle Cloud (Doc ID 2359020.1) for more information. VPN connection or Oracle Net encryption is also required for encryption-in-flight for any other database payload (e.g. data file or redo headers) that are not encrypted by TDE.

Using TDE to protect data is an important part of improving the security of the system. Users should, however, be aware of certain considerations when using any encryption solution, including:

- Additional CPU overhead: Encryption requires additional CPU cycles to calculate encrypted and decrypted values. TDE, however, is optimized to minimize the overhead by taking advantage of database caching capabilities and leveraging hardware acceleration within Exadata. Most TDE users see little performance impact on their production systems after enabling TDE. If performance overhead is a concern, please refer to the Oracle Database Advanced Security Guide.
- Lower data compression: Encrypted data compresses poorly because it must reveal no information about the original plaintext data. Thus, any compression applied to TDE encrypted data will have lower compression ratios. Hence, when TDE encryption is used, compression is not recommended to use with redo transport. However, when TDE is used in conjunction with Oracle databases compression technologies such as Advanced

Compression or Hybrid Columnar Compression, compression is performed before the encryption occurs, and the benefits of compression and encryption are both achieved.

- Key management: Encryption is only as strong as the key used to encrypt. Furthermore, the loss of the encryption key is tantamount to losing all data protected by that key. If encryption is enabled on a few databases, keeping track of the key and its lifecycle is relatively easy. As the number of encrypted databases grows, managing keys becomes an increasingly difficult problem. For users with a large number of encrypted databases, it is recommended that Oracle Key Vault be used on-premises to store and manage TDE master keys.

Database, OS Environment and Network Prerequisites

If On-Premises is not already enabled with TDE, please follow the master note **Master Note For Transparent Data Encryption (TDE) (Doc ID 1228046.1)** to enable TDE and create wallet files.

*** Oracle Database version on primary and standby databases must match during initial instantiation. For database software updates that are standby-first compatible, the primary and standby database Oracle Home software can be different. Refer to Oracle Patch Assurance - Data Guard Standby-First Patch Apply (Doc ID 1265700.1)*

Network Prerequisites

Since the ExaCC is connected by the TOR switch to the on-premises network and DNS is configured to resolve addresses from the on-premises to ExaCC networks there is no additional networking required for ExaCC Hybrid Data Guard.

On-Premises Prerequisites

The following prerequisites must be met before instantiating the standby database:

- Prompt-less SSH from ExaCC to on-premises must be configured.
- The Oracle Home for the on-premises database must be the same Oracle patchset as the standby database. If the ExaCC environment is on a different bundle patch level than the on-premises database it is recommended to patch the source environment to the same database bundle patch level as the database home in the ExaCC environment. (The command "$ORACLE_HOME/OPatch/opatch lspatches" can be executed to check the patches installed on both Source and Target environments)
- The on-premises database must be a Container Database (CDB). Non-CDBs are not currently supported in cloud environments.
- The steps outlined in this document assume that the on-premises primary database is not already part of an existing Data Guard broker configuration. If there is an existing broker configuration for the on-premises database it is assumed that the administrator has prior knowledge of the broker and knows how to add the new standby database to an existing broker configuration. A value other than 'NOCONFIG' for the following query implies an existing broker configuration.

```
SQL> select decode(count(1),0,'NOCONFIG') from v$DG_BROKER_CONFIG;
```

- Use the default listener named LISTENER. The steps outlined in this document assume the default listener name LISTENER is used. To verify run the following command from the on-premises machine. The expected result is shown.

```
$lsnrctl show current_listener | grep 'Current Listener' Current Listener is
 LISTENER
```

» Verify the listener port by running the following command from the on-premises machine. The expected result is shown.

```
$ lsnrctl stat| grep 'Connecting to'
Connecting to (ADDRESS=(PROTOCOL=tcp)(HOST=)(PORT=(1521)))
```

Implement MAA Best Practice Parameter Settings on the Primary Database

See Appendix A for a list of best practices. Completing this process on the primary database before instantiation is recommended. Especially configuring both online and standby redo logs which will be duplicated during the instantiation process.

Deployment Process

The Deployment process below assume the prerequisites have been met. The process of instantiating a standby database on an ExaCC is very similar to the process that would be followed in an on-premises configuration. The ExaCC environment specific configuration items are described by the remainder of this process.

Step 1: On Premises Configuration

The assumption is the On-Premises database has been configured per MAA best practices. See Appendix A for parameter settings. Below, a list of additional configuration items.

Set Maximum TCP socket size

Check the maximum operating system TCP socket sizes for the on premises system as well as the cloud instance with the following command run as root. (TCP socket sizes for the ExaCC are 128MB)

ON ON-PREMISES HOST

```
# /sbin/sysctl -a | egrep net.core.[w,r]mem_max
net.core.wmem_max = 4194304
net.core.rmem_max = 4194304
```

ON EXACC

```
# /sbin/sysctl -a | egrep net.core.[w,r]mem_max
net.core.wmem_max = 4194304
net.core.rmem_max = 4194304
```

If necessary, adjust all socket size maximums to 128MB or 134217728. For on premises systems consult your operating system guide for details about how to accomplish this. For the cloud instance edit the /etc/sysctl.conf file settings for net.core.wmem_max and net.core.rmem_max. If the values between on premises and ExaCC do not match, the network protocol will negotiate the lower of the two values. Therefore, the values between sites is not required to match though that is recommended in order to attain optimal transport performance.

```
net.core.rmem_max = 134217728 net.core.wmem_max = 134217728
```

Step 2: Create an ExaCC Database Using the Console

Creating a database through the console will establish a new RDBMS home which will then be used by the standby database. Create a database using the ExaCC console with a dummy database name which does not match the primary database. Upon completion of the instantiation of the standby database this dummy database can be

removed leaving the Oracle Home in place. Please follow Creating a Database Deployment in the Administering Oracle Database Exadata Cloud Service guide.

Step 3: Configure $ORACLE_HOME/network/admin required for Exadata Cloud

The ExaCC environment keeps separate tnsnames.ora and sqlnet.ora files for each database in a RDBMS home by setting the TNS_ADMIN environment variable both in the user's environment and as a clusterware environment variable for each database. If this variable is not set correctly in both locations, unexpected results will be experienced with Transparent Data Encryption and any SQL*Net connections.

1. Create the TNS_ADMIN directory $ORACLE_HOME/network/admin/<STANDBY db_unique_name> on each node of the ExaCC

```
$ mkdir -p /u02/app/oracle/product/18.0.0.0/dbhome_3/network/admin/<STANDBY
db_unique_name>
```

2. Copy the existing sqlnet.ora file from the created ExaCC database's TNS_ADMIN directory to the newly created TNS_ADMIN directory.

```
$ cp /u02/app/oracle/product/18.0.0.0/dbhome_3/network/admin/<db_unique_name of
created db>/sqlnet.ora
/u02/app/oracle/product/18.0.0.0/dbhome_3/network/admin/<STANDBY
db_unique_name>/sqlnet.ora
```

3. Edit the copied sqlnet.ora file and edit ENCRYPTION_WALLET_LOCATION and WALLET_LOCATION

```
ENCRYPTION_WALLET_LOCATION =
 (SOURCE=
  (METHOD=FILE)
   (METHOD_DATA=
    (DIRECTORY=/var/opt/oracle/dbaas_acfs/<STANDBY db_unique_name>/tde_wallet)))

WALLET_LOCATION =
(SOURCE=(METHOD=FILE)(METHOD_DATA=(DIRECTORY=/u02/app/oracle/admin/<STANDBY
db_unique_name>/db_wallet)))
```

4. Create the encryption wallet directory. Since the directory is on ACFS it is shared across all nodes of the cluster therefore it only needs to be created on one node.

```
$ mkdir -p /var/opt/oracle/dbaas_acfs/<STANDBY db_unique_name>/tde_wallet
```

5. Create the SSL wallet directory. Since this directory is on the local mount point, it must be created on all nodes of the cluster.

```
$ mkdir -p /u02/app/oracle/admin/<STANDBY db_unique_name>/db_wallet
```

Step 4: Copy the TDE wallet

Make sure that $ORACLE_HOME/network/admin/sqlnet.ora contains the following line wallet file location is defined as ENCRYPTION_WALLET_LOCATION parameter in sqlnet.ora

SQLNET.ORA on on-premise host

```
ENCRYPTION_WALLET_LOCATION =
 (SOURCE=
  (METHOD=FILE)
   (METHOD_DATA=
    (DIRECTORY=/var/opt/oracle/dbaas_acfs/<STANDBY db_unique_name>/tde_wallet)))
```

Copy the ewallet.p12 and cwallet.sso files on-Premises to the above directory on ExaCC host.

NOTE: Find the location of the wallet in the primary on-premises database by querying the v$encryption_wallet view

ON ON-PREMISES HOST

```
scp -i ~/<ssh_key> ewallet.p12 opc@<ExaCC Host>:/tmp
scp -i ~/<ssh_key> cwallet.sso opc@<ExaCC Host>:/tmp
```

ON EXACC HOST

```
$ chmod 777 /tmp/ewallet.p12
$ chmod 777 /tmp/cwallet.sso
$ sudo su – oracle
$ cp /tmp/ewallet.p12 /var/opt/oracle/dbaas_acfs/<STANDBY db_unique_name>/tde_wallet
$ cp /tmp/cwallet.sso /var/opt/oracle/dbaas_acfs/<STANDBY db_unique_name>/tde_wallet
chmod 600 /var/opt/oracle/dbaas_acfs/<STANDBY db_unique_name>/tde_wallet
```

Step 5: Instantiate the Standby Database

The standby database can be instantiated from a backup or the existing active primary database. MOS Note 2283978.1 should be used to instantiate databases with RDBMS 12.1 or higher from the active primary database using RMAN RESTORE...FROM SERVICE method. This is the most straightforward method for instantiating a standby and is the MAA recommended method.

NOTE: Ensure the oracle user's environment has TNS_ADMIN set appropriately so the correct tnsnames.ora is picked up during RMAN restore commands.

NOTE: Be sure to set database parameter diagnostic_dest='/u02/app/oracle' in the standby database pfile.

For RDBMS version 11.2 the RMAN DUPLICATE method must be used to instantiate from an active primary database. See MOS 1617946.1 for details.

Once the database is registered in Oracle clusterware be sure to set the TNS_ADMIN directory and a clusterware environment variable.

```
$ srvctl setenv database -d <STANDBY db_unique_name> -env
'TNS_ADMIN=/u02/app/oracle/product/18.0.0.0/dbhome_3/network/admin/<STANDBY
db_unique_name>'
```

Step 6: Remove 'dummy' database

Once the standby database is properly instantiated the dummy database created in step 2 can be removed following the process defined in the documentation.

Completed Instantiation

At the end of this instantiation,

- Hybrid DG config has been configured
- Cloud infrastructure is managed and non-DB life cycle operations (e.g. bursting) can be executed
- DG operations and DB management for this “standby database” is manual

Validate DR Readiness

Best practice is to use Active Data Guard to offload read-only workload to the standby database to provide continuous, application-level validation that the standby is ready for production. This provides a level of assurance in addition to continuous Oracle block-level validation performed by Data Guard apply processes. It is also a best practice to periodically place the standby in read/write mode (using Data Guard Snapshot Standby) to validate its readiness to support read-write production workloads. A snapshot standby may also be used for a final level of pre-production functional and performance testing of patches and upgrades since the DR system is sized similarly to the production system. A Snapshot Standby continues to receive redo from the primary database where it is archived for later use, thus providing data protection at all times. Recovery time (RTO), however, will be extended by the amount of time required to convert the Snapshot Standby back to the standby database if a failover is required while testing is in progress. Note that additional storage is required for the fast recovery area when a standby is in snapshot mode (to hold archived redo received from the primary production database for later use and current redo and flashback logs generated by the snapshot standby). Steps for converting a standby to a snapshot standby and back are listed in the section below. Please refer to Oracle documentation for additional details on Data Guard Snapshot Standby. Optionally, you may perform an actual switchover or failover operation to the cloud for a complete end-to-end DR test; for more details see Failover/Switchover to the Cloud.

Converting Standby Database to a Snapshot Standby

A snapshot standby is a fully updatable standby database that is created from a physical standby database. On snapshot standby databases, redo data is received but not applied until the snapshot standby database is converted back to a physical standby database.

The benefits of using a snapshot standby database include the following:

1. It provides an exact replica of a production database for development and testing purposes while maintaining data protection at all times. You can use the Oracle Real Application Testing option to capture primary database workload and then replay it for test purposes on the snapshot standby.
2. It can be easily refreshed to contain current production data by converting to a physical standby and resynchronizing.

Follow the steps below to convert a physical standby database to a snapshot standby

Convert the standby to a snapshot standby and validate

Via Data Guard broker issue the following commands

```
DGMGRL> convert database 'stby' to snapshot standby;
DGMGRL> SHOW CONFIGURATION;
Configuration - DRSolution
Protection Mode: MaxPerformance Databases:
prmy - Primary database stby - Snapshot standby database
Fast-Start Failover: DISABLED
Configuration Status: SUCCESS
```

NOTE: A snapshot standby database must first be converted back into a physical standby database before performing a switchover.

Convert the snapshot standby back into a physical standby database

Via Data Guard broker issue the following commands

```
DGMGRL> CONVERT DATABASE 'stby' to PHYSICAL STANDBY;
```

Failover/Switchover to the Cloud

You can manually execute a Data Guard role transition at any time. Customers may also choose to automate Data Guard failover by configuring Fast-Start failover. Switchover and failover reverse the roles of the databases in a Data Guard configuration – the standby in the cloud becomes primary and the original on-premises primary becomes a standby database. Refer to Oracle MAA Best Practices for additional information on Data Guard role transitions.

Switchovers are always a planned event that guarantees no data is lost. To execute a switchover perform the following in Data Guard Broker

```
DGMGRL> validate database stby;
Database Role: Physical standby database Primary Database: pri
Ready for Switchover: Yes
Ready for Failover: Yes (Primary Running)
DGMGRL> switchover to <target standby>;
```

A failover is an unplanned event that assumes the primary database is lost. The standby database is converted to a primary database immediately; after all available redo from the primary has been applied. After a failover the old primary database must be reinstated as a physical standby which is made simpler with flashback database and Data Guard broker enabled. To execute a failover and reinstatement execute the following in Data Guard Broker.

```
DGMGRL> failover to stby;
Performing failover NOW, please wait...
Failover succeeded, new primary is "stby"
Execute startup mount on one instance of the old primary before reinstating.
SQL> shutdown abort
SQL> startup mount
DGMGRL> reinstate database pri
Reinstating database "pri", please wait...
```

For more information on role transitions using the Data Guard Broker see the broker documentation for Oracle Database 11g or 12c.

Switch back to On-Premises

The same role transition procedure mentioned in the failover/switchover process is applied again when you are ready to move production back to the on-premises database.

Health Checks and Monitoring

After the standby is instantiated, a health check should be performed to ensure the Data Guard databases (primary and standby) are compliant with Oracle MAA best practices. It is also advisable to perform the health check on a monthly basis as well as before and after database maintenance. There are several methods for checking the health of a Data Guard configuration:

Oracle MAA Scorecard

Oracle provides several automated health check tools that can be downloaded from My Oracle Support specific for the type of hardware platform:

- ORAchk applicable to generic platform (suitable for Database Cloud Service)
- exachk applicable to Exadata Database Machine (suitable for Exadata Cloud Service)

Each of the automated checks include an Oracle MAA Scorecard that reports on a number of key Data Guard configuration best practices in addition to many other checks.

Oracle strongly recommends the use of these automated tools for comprehensive health check of not only the Data Guard configuration but the system as a whole. The health checks are regularly updated with current information. Be sure to download the latest version of the health checks applicable to your platform.

Data Guard Specific Queries (Applicable from Oracle Database 11g onward)

A set of Data Guard specific queries are provided in MOS 2064281.1

Data Guard VALIDATE DATABASE (Applicable from Oracle Database 12c onward)

The Data Guard Broker VALIDATE DATABASE command is highly recommended for the most comprehensive Data Guard specific health check. VALIDATE DATABASE performs an extensive configuration check and validates the configuration's readiness for switchover or failover.

Example:

```
DGMGRL> validate database APPDBB;

  Database Role:     Physical standby database
  Primary Database:  pri

  Ready for Switchover:  Yes
  Ready for Failover:    Yes (Primary Running)
```

See the Data Guard broker documentation for more information on the extensive checks performed by the VALIDATE DATABASE command

Client Failover

Automating client failover, the process by which clients are reconnected to the active primary database after a failure, includes relocating database services to the new primary database as part of a Data Guard failover, notifying clients that a failure has occurred in order to break them out of TCP timeout, and redirecting clients to the new primary database. Configuration details are thoroughly covered in the papers MAA Best Practices for Continuous Availability. Please consult the link for full details.

Conclusion

Hybrid Data Guard using Exadata Cloud at Customer systems is an economical method to achieve Disaster Recovery readiness. Utilizing Maximum Availability Architecture best practices ensures the best solution for data protection and availability.

Appendix A: MAA Best Practice Parameter Settings

The following settings are recommended to follow MAA best practices in order to provide maximum availability and protection of the data. These parameters should be set on both the primary and standby databases.

- ARCHIVELOG enabled
- Flashback database on
- FORCE LOGGING enabled
- Use SPFILE
- Use Data Guard Broker
- COMPATIBLE uses 4 decimals and is the same on both databases
- DB_FILES=1024
- Online Redo Log characteristics
    - Only multiplexed on normal redundancy storage; single member groups when using high redundancy storage
    - Minimum of 3 online log groups per thread
    - Reside on DATA disk group
- Standby Redo Log characteristics
    - Identical size as online redo logs
    - For RAC, assign SRL groups to a thread
    - Single member only
    - Same number of groups per thread as online redo log groups
    - Reside on DATA disk group
- LOG_BUFFER = 128M for 11.2; 256M for 12.1+
- DB_BLOCK_CHECKING=OFF *Note: this setting could affect performance and should be enabled only after proper testing of the application.*
- DB_BLOCK_CHECKSUM=TYPICAL
- STANDBY_FILE_MANAGEMENT=AUTO
- DB_LOST_WRITE_PROTECT=TYPICAL
- DB_FLASHBACK_RETENTION_TARGET=minimum of 120
- FAST_START_MTTR_TARGET=300
- USE_LARGE_PAGES=ONLY if hugepages are configured and properly sized on the on-prem system
- CLUSTER_INTERCONNECTS set per gv$cluster_interconnects
- PARALLEL_THREADS_PER_CPU=1

- DB_CREATE_ONLINE_LOG_DEST_1= DATA disk group
- DB_CREATE_ONLINE_LOG_DEST_n other than 1 should only be set when DATA is not high redundancy
- DB_CREATE_FILE_DEST uses DATA disk group
- DB_RECOVERY_FILE_DEST uses RECO disk group
- Recyclebin is on

ORACLE

Oracle Corporation, World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065, USA

Worldwide Inquiries
Phone: +1.650.506.7000
Fax: +1.650.506.7200

CONNECT WITH US


blogs.oracle.com/oracle


facebook.com/oracle


twitter.com/oracle


oracle.com

Integrated Cloud Applications & Platform Services

White Paper Title
September 2019
Author: Kazuhiro Ikeda, Sebastian Solbach
Contributing Authors: Ramachandran Pandrapattahil