ORACLE

# Disaster Recovery using Exadata Cloud

## On-Premises Primary to Standby in Exadata Cloud Service or Gen 2 Exadata Cloud at Customer

March 6, 2020

## DISCLAIMER

This document in any form, software or printed matter, contains proprietary information that is the exclusive property of Oracle. Your access to and use of this confidential material is subject to the terms and conditions of your Oracle software license and service agreement, which has been executed and with which you agree to comply. This document and information contained herein may not be disclosed, copied, reproduced or distributed to anyone outside Oracle without prior written consent of Oracle. This document is not part of your license agreement nor can it be incorporated into any contractual agreement with Oracle or its subsidiaries or affiliates.

This document is for informational purposes only and is intended solely to assist you in planning for the implementation and upgrade of the product features described. 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 in this document remains at the sole discretion of Oracle.

Due to the nature of the product architecture, it may not be possible to safely include all features described in this document without risking significant destabilization of the code.

# TABLE OF CONTENTS

# INTRODUCTION

Oracle's Maximum Availability Architecture (Oracle MAA) is the best practices blueprint for data protection and availability for Oracle databases deployed on private or public clouds. Oracle Data Guard and Active Data Guard support disaster recovery (DR) for databases with recovery time objectives (RTO) and recovery point objectives (RPO) that cannot be met by merely restoring from backup. Customers use these solutions to deploy one or more synchronized replicas (standby databases) of a production database (the primary database) in physically separate locations to provide high availability, comprehensive data protection, and disaster recovery for mission-critical data.

An effective disaster recovery plan can be costly due to the need to establish, equip and manage a remote data center. The Exadata Cloud (including both Exadata Cloud Service and Exadata Cloud at Customer) offers a great alternative for hosting standby databases for customers who do not have a DR site or who prefer to avoid dealing with the cost or complexity of managing a remote data center. Existing production databases remain on-premises and standby databases used for DR are deployed on the Exadata Cloud. This mode of deployment is commonly referred to as a hybrid Data Guard implementation.

Customers may choose to deploy either a Data Guard or an Active Data Guard standby on Exadata Cloud depending upon their requirements. While there are some unique considerations to a hybrid Data Guard configuration, it follows the same Oracle MAA best practices as with any Data Guard deployment. This Oracle MAA blueprint details Oracle MAA Best Practices and provides a procedural overview for deploying DR on the Oracle Cloud using Exadata Cloud Service (ExaCS) and Exadata Cloud at Customer Gen2 (ExaCC). This paper is intended for a technical audience having knowledge of Oracle Database, Data Guard or Active Data Guard, and Oracle Database backup and recovery. This paper also assumes a basic understanding of services offered by the Oracle Cloud.

# DISASTER RECOVERY TO EXADATACLOUD WITH DATA GUARD AND ACTIVE DATA GUARD

The Oracle Cloud offers an extensive set of cloud services tailored to specific customer requirements: Platform as a Service (PaaS), Infrastructure as a Service (IaaS), and Software as a Service (SaaS). Disaster Recovery (DR) for on-premises systems can be configured using the Exadata Cloud Service (ExaCS) or Exadata Cloud at Customer (ExaCC).

Data Guard is included in Oracle Database Enterprise Edition (no separate license is required for on-premises systems) and is supported by all editions (Enterprise, High Performance, and Extreme Performance) of Exadata Cloud Service and Exadata Cloud at Customer. Please bear in mind that Oracle Database Standard Edition does not support Data Guard. Search "Licensing Information" in the Oracle Database Documentation for the selected version to understand the supportability matrix on what is supported in Enterprise Edition versus Standard Edition.

Active Data Guard extends Data Guard capabilities by providing advanced features for data protection and availability such as auto block repair as well as offloading read-only workloads and fast incremental backups from a production database. It should be noted that the use of the term "Data Guard" within this document applies to Active Data Guard as well as Data Guard. Active Data Guard is included in the Oracle Database Extreme Performance Edition, Exadata Cloud at Customer, and Exadata Cloud Service. When used in a hybrid configuration, Active Data Guard must also be licensed for the on-premises system.

*The process described in this document is valid for Exadata Cloud Service (ExaCS) and Exadata Cloud at Customer (ExaCC) Gen 2. The process to deploy hybrid Data Guard in ExaCC Gen 1 is described in Hybrid Data Guard to ExaCC.*

## BENEFITS OF HYBRID DATA GUARD IN THE CLOUD

There are numerous benefits to using a hybrid Data Guard configuration in the Oracle Cloud. The primary benefits are listed below.

1. The cloud data center and infrastructure are managed by Oracle.
2. The Oracle Cloud provides basic system life cycle operations including scaling CPU resources, patching and backup/recovery.
3. Oracle Data Guard provides disaster recovery, data protection and the ability to offload activity for higher utilization and return on investment.
4. A hybrid Data Guard configuration provides the ability to switchover (planned events) or failover (unplanned events) production to the standby database in the cloud during planned maintenance or unplanned outages. Once a failed on-premises database is repaired, the database can be resynchronized with the current production database in the cloud and then production can then be switched back to the on-premises database.
5. Utilization of the same Oracle MAA best practices as on-premises deployment. Use of additional Oracle MAA best practices specific to hybrid Data Guard deployments that are specified in this paper.
    a. When configured with MAA practices a hybrid Data Guard configuration provides:
        i. Recovery Time Objective (RTO) of seconds with automatic failover when configured with Data Guard Fast-Start failover and
        ii. Recovery Point Objective (RPO) less than a second for Data Guard with ASYNC transport or
        iii. RPO zero for Data Guard in SYNC or FAR SYNC configurations

*Data Guard Life Cycle Management e.g. switchover, failover and reinstate is a manual process in Hybrid Data Guard configurations.*

When deploying Exadata Cloud for Disaster Recovery, MAA recommends:

1. Create a DB System target that is symmetric or similar to the on-premises primary database to ensure performance SLAs can be met after a role transition. Use RAC for RAC, Exadata for Exadata, etc.
2. Ensure network bandwidth is capable of handling peak redo rates in addition to existing network traffic. My Oracle Support document 2064368.1 provides additional network bandwidth troubleshooting guidance for assessing and tuning network performance for Data Guard and RMAN.
3. Ensure network reliability and security between on-premises to and from the Cloud environment.
4. Use Active Data Guard for additional auto-block repair, data protection and offloading benefits.
5. Use Oracle Transparent Data Encryption (TDE) for both primary and standby databases. My Oracle Support document 2359020.1 has additional details on TDE behavior in cloud configurations.

*Backups to the object store were not supported for standby databases at the time of publish. However, it is recommended that backups are configured on the cloud database for the time when standby database backup functionality is made available in case the cloud database becomes the primary.*

## Using Standby Database to reduce downtime during Planned Maintenance

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

**Standby-first Patch Apply**

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 enabled for Real Application Clusters (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 (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 a fast fall-back 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 use case for standby in Exadata Cloud is for database rolling upgrade to reduce downtime when upgrading to new Oracle Database releases which are not standby-first compatible. 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 Database Rolling Upgrades for more information on rolling upgrade from release 12.1.0.1 or earlier. A more efficient database rolling upgrade process using the standby database exists for upgrading Data Guard environments as of the 12.1.0.2 release. Refer to Automated Database Upgrades using Oracle Active Data Guard and DBMS_ROLLING and the Using DBMS_ROLLING to Perform a Rolling Upgrade section in the Data Guard documentation.

## SERVICE LEVEL REQUIREMENTS

Hybrid Data Guard 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, 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 compared to the on-premises production system. This occurs when administrators purposefully under-configure standby resources to reduce cost; accepting a 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. Rapid provisioning available with the cloud can enable a middle ground where there is less capacity deployed during steady-state, but the new primary is rapidly scaled-up should a failover be required.

*The reduced resources during steady state in a rapid provisioning approach could impact the ability of recovery to keep the standby database current with the primary thus createing an apply lag and impacting RTO. This approach should only be considered after thorough testing.*

## SECURITY REQUIREMENTS

Oracle MAA best practices recommends using Oracle Transparent Data Encryption (TDE) to encrypt both primary and standby databases to ensure data is encrypted at-rest. Data can be converted *during the migration process* but it is 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. A 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 low compression ratios. Hence, when TDE encryption is used, it is not recommended to use redo transport compression. However, when TDE is used in conjunction with Oracle Database 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.

  Assuming best practices are being followed, if the on-premises database 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.

*In the event TDE is not desired on-premises, refer to My Oracle Support note 2359020.1 for information about encryption behavior between encrypted and unencrypted databases in a Data Guard configuration.*

## DATABASE, OS ENVIRONMENT AND NETWORK PREREQUISITES

**TABLE 1: PREREQUISITES**

| | On-Premises | Oracle Cloud (ExaCS and ExaCC Gen 2) |
|---|---|---|
| Operating System | Linux, Windows or Solaris X86 (My Oracle Support Note 413484.1 for Data Guard cross-platform compatibility) | Oracle Enterprise Linux (64-bit) |
| Oracle Database* | » *Oracle Database Enterprise Edition 11.2.0.4 (64-bit)*<br>» *Oracle Database Enterprise Edition 12.1.0.2 (64-bit)*<br>» *Oracle Database Enterprise Edition 12.2.0.1 (64-bit)*<br>» *Oracle Database Enterprise Edition 18c (64-bit)*<br>» *Oracle Database Enterprise Edition 19c (64-bit)* | Extreme performance / BYOL |
| RAC | RAC or non-RAC | RAC or non-RAC |
| Multitenant | For 12.1 and above, primary database has to be a CDB/PDB database. | Multitenant Database<br>Non-CDB |
| Physical Vs Virtual | Physical or Virtual | Exadata Virtual |
| Database Size | Any Size | Any size. For shape limits please consult Exadata Cloud documentation |
| TDE Encryption | Recommended | Mandatory for cloud databases |

** 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)*

## Cloud Network Prerequisite

Data transfers from on-premises to Oracle Cloud Infrastructure (OCI) use the public network, VPN and/or the high bandwidth option provided by Oracle FastConnect.

In a Data Guard configuration, the primary and standby must be able to communicate bi-directionally. This requires additional network configuration to allow access to ports between the systems.

*Network connectivity configuration is not required for Exadata Cloud at Customer because it is deployed on the on-premises network. Skip to the 'On-Premises Network Configuration' section if using ExaCC.*

## Secure Connectivity

*For Exadata Cloud Service (not required for Exadata Cloud at Customer)*

There are two options to privately connect your cloud network to the on-premises network; FastConnect and IPSec VPN. Each of these methods requires a Dynamic Routing Gateway (DRG) to connect to your Virtual Cloud Network (VCN). Details for creating a DRG can be found in the documentation at this link.

### Oracle FastConnect

OCI FastConnect provides a method to create a dedicated, private connection between your data center and OCI. FastConnect provides higher-bandwidth options and a more reliable and consistent networking experience compared to internet-based connections. More details on FastConnect can be reviewed here.

### IPSec VPN

IPSec stands for Internet Protocol Security or IP Security. IPSec is a protocol suite that encrypts the entire IP traffic before the packets are transferred from the source to the destination. For an overview of IPSec in OCI review the documentation here.

## Public Internet Connectivity

Connectivity between OCI and on-premises can be achieved through the public internet as well. This method is not by default secure and additional steps must be taken to secure transmissions. The remainder of this whitepaper assumes public internet connectivity.

By default, cloud security for port 1521 is disabled. Also, this default pre-configured port in the cloud for either a Virtual Machine (VM) or Bare Metal (BM) has open access from the public internet.

1. If Virtual Cloud Network (VCN) for the standby database doesn't have an Internet Gateway, one must be added. The link below describes how to create an internet gateway:
   https://docs.us-phoenix-1.oraclecloud.com/Content/Network/Tasks/managingIGs.htm
2. Ingress and egress rules must be configured in the VCN security list to connect from/to the on-premises database. The link below provides additional information.
   https://docs.us-phoenix-1.oraclecloud.com/Content/Network/Concepts/securitylists.htm

## On-Premises Network Configuration

In a Data Guard configuration, information is transmitted between primary and standby databases in both directions. This requires basic configuration, network tuning and opening of ports at both primary and standby databases.

It is especially important to ensure the bandwidth exists to support the redo generation rate of the primary database. Follow My Oracle Support note 2064368.1 Assessing and Tuning Network Performance for Data Guard and RMAN to assess and tune the network link between the on-premises and cloud environments.

## On-Premises Prerequisites

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

- Configure name resolution
  - For ExaCC, since the clusters reside on the on-premises network, the on-premises DNS should resolve each cluster and no further configuration should be necessary.
  - For Exadata Cloud Service, name resolution between the clusters needs to be configured. This can be done either through a static file like /etc/hosts, or configuring the on-premises DNS to properly resolve the public IP address of the OCI instance. Also, the on-premises firewall will need to have Access Control Lists properly configured to allow SSH and Oracle Net to be accessed from the on-premises system to OCI.
- Data Guard in a DR situation requires access from the cloud instance to the on-premises database; the primary database listener port must be opened with restricted access from the cloud IP addresses using features like iptables. Since every corporation has different network security policies, the network administrator will need to perform operations similar to the cloud-side network configuration shown in preceding sections
- Prompt-less SSH from Exadata Cloud to the on-premises machine. This is configured both for on-premises to Exadata Cloud during the provisioning process and from the Cloud to on-premises.
- The configuration of the on-premises firewall to allow inbound SSH connectivity from the Exadata Cloud Service to the on-premises machine.
- It is strongly recommended to complete the network assessment in the previous section 'On-Premises Network Configuration'. Setting the appropriate tcp socket buffers sizes is especially important for ASYNC redo transport.
- The RDBMS software must be the same between the primary and standby for instantiation. If the current on-premises Oracle Database version is not available in Exadata Cloud, the primary database must be patched or upgraded to an available cloud bundle patch. The available bundle patches on the cloud can be listed with the command below. The installation of the software is described in the Deployment Process section.

  **As root:**

```
# dbaascli cswlib list
```

- One-off patches and merge patches should also match between the primary and standby databases. Find the applied one-offs with the command below and apply any on-premises one-offs to the cloud database software using the patch documentation.

  **As oracle:**

```
$ORACLE_HOME/OPatch/opatch lspatches
```

- 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 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(*),0,'NOCONFIG') from v$DG_BROKER_CONFIG;
```

- Verify the listener port for the on-premises listener by running the following command from the on-premises machine. This port is needed when configuring redo transport and will be entered into the tns descriptors during the deployment process.

```
$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. The process described in this document assumes the primary database has been configured prior to instantiation. The redo log configuration is especially important prior to instantiation.

## Validating Connectivity between On-Premises and Exadata Cloud Hosts

Once all the networking steps are implemented successfully, run the command below to validate if the connection looks good from all sources to all targets and vice versa. If telnet is successful, proceed to the next step.

ON ON-PREMISE HOST

```
[root@onpremise1 ~]# telnet <TARGET HOST IP ADDRESS> <PORT>
Trying xxx.xxx.xxx.xxx...
Connected to xxx.xxx.xxx.xxx.
Escape character is '^]'.
^C^]q
telnet> q
Connection closed.
```

ON CLOUD HOSTS

```
[root@oci2 ~]# telnet <TARGET HOST IP ADDRESS> <PORT>
Trying xxx.xxx.xxx.xxx...
Connected to xxx.xxx.xxx.xxx.
Escape character is '^]'.
^]q
telnet> q
Connection closed.
```

*netcat `(nc -zv <IP> <PORT>)` can be used in place of telnet*

# DEPLOYMENT PROCESS

The deployment process below assumes the prerequisites have been met. Once the network is configured between the environments, the process of instantiating a Hybrid Data Guard standby database is similar to the process that would be followed in an on-premises configuration. Once complete the cloud database will be visible in the cloud console and will be registered for use with cloud tooling, patching and backup functionality. Data Guard operations such as switchover, failover and reinstate will be manually executed through Data Guard Broker as in an on-premises environment.

*Unless otherwise noted, commands are the same whether the cloud system is Exadata Cloud Service or Exadata Cloud at Customer Gen2.*

*This process assumes ASM and Oracle Managed Files (OMF) is configured on the primary database. The process for configuring Hybrid Data Guard in non-OMF and non-ASM configurations is outside the scope of this document.*

## Prerequisite: Update Cloud Tools

This process requires a minimum dbaastools rpm version of 18.2.3.2.0_190618. Updating to the latest dbaastools rpms are always recommended.

Apply the latest tools rpm with the following:

```
(as root) # dbaascli patch tools apply LATEST
```

## Step 1: Create the Cloud Database

The cloud console create database functionality will be used to create what will become the standby database. Creating the database through the console (then deleting the files and instantiating) will ensure the database is visible in the console and is registered properly for use with the cloud tooling. Before creating the database, the bundle patch of the primary database must be known as well as all one-off patches.

### Select and download the desired RDBMS version and Bundle Patch

Due to space limitations, the Exadata Cloud stores one bundle patch (BP) image for each version locally in the cluster's ACFS storage area. Whichever BP is downloaded at the time the database is created via the console is the BP the new database will use. To download the desired bundle patch for your version:

First list the available versions and bundle patches:

**As root:**

```
# dbaascli cswlib list

DBAAS CLI version 19.4.1.0.0
Executing command cswlib list
############ List of Available BP ##############
-APR2017 (For DB Versions  12201 12102 11204)
-JAN2018 (For DB Versions  12201 12102 11204)
-APR2018 (For DB Versions  12201 12102 11204)
-JUL2018 (For DB Versions  18000 12201 12102 11204)
-OCT2018 (For DB Versions  18000 12201 12102 11204)
-JAN2019 (For DB Versions  18000 12201 12102 11204)
-APR2019 (For DB Versions  18000 12201 12102 11204 19000)
-JUL2019 (For DB Versions  18000 12201 12102 11204 19000)

#### List of Available NONCDB BP ####
-APR2018 (For DB Versions  12201 12102)
-JAN2019 (For DB Versions  12201 12102)
-APR2019 (For DB Versions  12201 12102)
-JUL2019 (For DB Versions  12201 12102)
```

Then download the desired bundle patch:

**As root:**

```
# dbaascli cswlib download --version 12201 --bp JUL2019

DBAAS CLI version 19.4.1.0.0
Executing command cswlib download --version 12201 -bp JUL2019
INFO: CSWLIB update  db image bits
INFO: Log file is: /var/opt/oracle/log/misc/cswlib/cswlib_<date>_<timestamp>.log
INFO: CSWLIB update_bits of  12201 succeded !
```

*The listed bundle patches for each version are the only available images for the Exadata Cloud. If the on-premises database's bundle patch is not available in the cloud, the primary database should be patched or upgraded to one of the available bundle patches before instantiating.*

## Create the Database Using the Cloud Console

Use the Cloud console to create the database in the proper bundle patch by selecting the version downloaded in the prior step. There will not be a bundle patch listed, only the version. For those not familiar with creating a database via the console, refer to the documentation for ExaCS and ExaCC.

*Per Data Guard requirements, the database name (db_name) of the standby must be the same as the primary database while the db_unique_name must be different. The create database screen for ExaCC has a separate field for the unique name that defaulted to be different from the primary. For ExaCS there is no explicit setting for unique name. The tooling will set a db_unique_name different than db_name automatically.*

*When deploying the Cloud database it is recommended to include a backup configuration. Backups to the object store were not supported for standby databases as the time of publish. However, it is recommended that backups are configured for the time when standby databases can be backed up or in case the cloud database becomes the primary.*

## Step 2: Manually Delete the Database Created By the API

Once the database is provisioned and ready, perform the steps below to delete the database files and instantiate the standby from the on-premises database using RMAN. The steps to initiate the RMAN duplicate will be described later in this document.

To delete the provisioned database use the manual method of removing the database files from ASM diskgroups. Do not use DBCA as this will also remove the srvctl registration as well as the oratab entries which should be retained for the standby.

To manually delete the database on the Exadata Cloud Service hosts, run the steps below

1. This process will replace the controlfile which stores the RMAN settings for backup to the Object Store. Therefore these settings need to be saved and replaced after the standby database is instantiated.

2.Create a script to remove all database files

```
SQL> set heading off linesize 999 pagesize 0 feedback off trimspool on
SQL> spool /tmp/delete_ASM_files.sh
SQL> select 'asmcmd rm '||name from v$datafile
union all
select 'asmcmd rm '||name from v$tempfile
union all
select 'asmcmd rm '||member from v$logfile;
SQL> spool off
```

```
SQL> create pfile='/tmp/<standby DB_UNIQUE_NAME>.pfile' from spfile;  #Backup of spfile
```

```
$chmod 777 /tmp/delete_ASM_files.sh
```

3. Save the RMAN settings to be re-applied after instantiation.

The output from the command will only be printed to the log file. It will not be visible on the screen:

```
$ rman target / log='/tmp/rman_setting.log'
RMAN> show all;
RMAN> exit
```

*IMPORTANT: The instantiation process will replace the controlfile with the controlfile from the primary database, thus replacing the RMAN configuration deployed by the Cloud tooling. This step saves the configuration which will be replaced after instantiation. This is especially important if backups were configured.*

4. Shutdown the database

First collect the clusterware configuration of the database for future reference:

```
$ srvctl config database -d <db_unique_name> > /tmp/<standby db_unique_name>.config
```

Finally, stop the database:

```
$ srvctl stop database -d <db_unique_name> -o immediate
```

5. Remove database files

Remove the existing data files, log files and tempfile. The password file will be replaced and the spfile will be reused.

As grid user (sudo from opc user to grid user)

Edit `/tmp/delete_ASM_files.sh` created previously to remove any unneeded lines from sqlplus, leaving only lines beginning with 'asmcmd'.

```
[grid@<host> ~]$ vi /tmp/delete_ASM_files.sh
```

Then save and execute the script

```
[grid@<host> ~]$ . /tmp/delete_ASM_files.sh
```

All files for the starter database have now been removed.

## Step 3: Copy the Password File to the Exadata Cloud

The password file for the Cloud database must be replaced by the password file of the on-premises primary database.

### Check password file location

If Oracle Clusterware is running on the on-premises host, check the password file location.

```
$ srvctl config database -db testdbname
Database unique name: testdbname
Database name:
Oracle home: /u02/app/oracle/product/12.2.0.0/dbhome_2
Oracle user: oracle
Spfile: +DATA/testdbname/spfiledbtestdbname.ora
Password file: +DATA/testdbname/PASSWORD/orapwd<sid> <======== password file location
Domain: domainname.xxxx.xxxx
```

### Copy password file on-premises to Cloud Exadata

Copy password file to all Exadata Cloud nodes.

If on-premises password file location is non-ASM, copy the file as below.

```
$ scp -i <ssh key> $ORACLE_HOME/dbs/orapw<SID> opc@<Public-IP-OCI-HOST>:/tmp
```

If password file location is ASM, switch user to "grid" or the ASM owner, source the environment variables and then copy the password file as below.

on-premises

```
$ sudo su - grid
$ export ORACLE_SID=<ASM ORACLE_SID>
$ export ORACLE_HOME=<GRID_HOME>
$ asmcmd
ASMCMD> cd +<DISKGROUP_NAME>/<DB_UNIQUE_NAME>/PASSWORD
ASMCMD> cp orapw<SID> /tmp
copying +DATA/testdbname/PASSWORD/orapw<sid> -> /tmp/orapw<sid>

scp -i <ssh key> /tmp/orapw<SID> opc@<Public-IP-OCI-HOST>:/tmp
```

### Place the password file in the proper Exadata Cloud location for the standby database.

As opc user on Exadata Cloud host

```
$ chmod 777 /tmp/<password file name>
$ sudo su – grid
```

Place the password file in ASM:

Use `srvctl config database -db <stadndby DB_UNIQUE_NAME>` to find current passwordfile location

As grid user:

```
$ asmcmd pwcopy --dbuniquename <standby DB_UNIQUE_NAME> /tmp/<password file name> <current standby
password file> –f
```

*Error ASMCMD-9453: failed to register password file as a CRS resource can be ignored if you are reusing the previously registered password file location.*

## Step 4: Copying the wallet file to Exadata Cloud

Retrieve the TDE wallet location from the on-premises database with the following query:

```
SQL> select WRL_PARAMETER from v$encryption_wallet;

WRL_PARAMETER
--------------------------------------------------------------------------------
/u01/app/oracle/admin/<db_unique_name>/wallet/
```

Copy the `ewallet.p12` and `cwallet.sso` files from on-premises directory to the `/tmp` directory on Exadata Cloud node1. On the primary it may be necessary to copy the files out of ASM to /tmp as with the password file.

**ON ON-PREMISES HOST**

```
scp -i ~/<ssh_key> <PATH>/ewallet.p12 opc@<Public-IP-OCI-HOST1>:/tmp
scp -i ~/<ssh_key> <PATH>/cwallet.sso opc@<Public-IP-OCI-HOST1>:/tmp
```

Remove old wallet files in `/var/opt/oracle/dbaas_acfs/<standby db_name>/tde_wallet`. Execute the commands only on node1. (Exadata Cloud nodes share storage under `/var/opt/oracle/dbaas_acfs` using ACFS)

**ON CLOUD HOST NODE1**

as opc user

```
$ chmod 777 /tmp/ewallet.p12
$ chmod 777 /tmp/cwallet.sso
$ sudo su – oracle
$ cp /tmp/ewallet.p12 /var/opt/oracle/dbaas_acfs/<standby db_name>/tde_wallet/
$ cp /tmp/cwallet.sso /var/opt/oracle/dbaas_acfs/<standby db_name>/tde_wallet/
$ chmod 600 /var/opt/oracle/dbaas_acfs/<standby db_name>/tde_wallet/*wallet*
```

*If the wallet files of the primary are stored in ASM, the ASM keystore must be merged into a local file system keystore before copying the local keystore files to the cloud. See My Oracle Support Document ID 2193264.1 for details.*

## Step 5: (11g only) Configure Static Listeners

This step is only required for 11.2 databases. A static listener is needed for Data Guard Broker communication in addition to initial instantiation of a standby database. The static listener enables remote connections to an instance while the database is down in order so that the instance can be started. See My Oracle Support Doc ID 1387859.1 for additional details.

As the grid user or software owner, add the following entry to the listener.ora on the Exadata Cloud node 1 after replacing the variables. The listener.ora is located in <grid home>/network/admin

**LISTENER.ORA ON THE FIRST NODE OF THE STANDBY**

```
SID_LIST_LISTENER =
  (SID_LIST =
    (SID_DESC =
      (GLOBAL_DBNAME = <DB_UNIQUE_NAME>_DGMGRL)
      (ORACLE_HOME = <Local Oracle Home>)
      (SID_NAME = <ORACLE SID of the local instance>)
    )
    (SID_DESC =
      (GLOBAL_DBNAME = <DB_UNIQUE_NAME of the Cloud database>)
      (ORACLE_HOME = <Local Oracle Home>)
      (SID_NAME = <ORACLE SID of the local instance>)
    )
 )
```

**LISTENER.ORA ON ALL OF THE REMAINING NODES IN THE CONFIGURATION**

```
SID_LIST_LISTENER =
  (SID_LIST =
    (SID_DESC =
      (GLOBAL_DBNAME = <DB_UNIQUE_NAME>_DGMGRL)
      (ORACLE_HOME = <Local Oracle Home>)
      (SID_NAME = <ORACLE SID of the local instance>)
    )
)
```

Finally reload the listeners (as grid user or clusterware software owner):

```
$ORACLE_HOME/bin/lsnrctl reload
```

## Step 6: Oracle Net Encryption and TNS Entries for Redo Transport

To protect against plaintext or unencrypted tablespace redo from being visible on the WAN place the following entries in the sqlnet.ora file on all on-premises and cloud.

Cloud deployments utilizes the TNS_ADMIN variable to separate tnsnames.ora and sqlnet.ora in shared database homes. Therefore the cloud sqlnet.ora, and by extension tnsnames.ora, for a given database are located in `$ORACLE_HOME/network/admin/<db_name>`. These values should already be set by the deployment tool in cloud configurations.

**SQLNET.ORA ON ON-PREMISES HOST(S)**

```
SQLNET.ENCRYPTION_SERVER=REQUIRED
SQLNET.CRYPTO_CHECKSUM_SERVER=REQUIRED
SQLNET.ENCRYPTION_TYPES_SERVER=(AES256,AES192,AES128)
SQLNET.CRYPTO_CHECKSUM_TYPES_SERVER=(SHA1)
SQLNET.ENCRYPTION_CLIENT=REQUIRED
SQLNET.CRYPTO_CHECKSUM_CLIENT=REQUIRED
SQLNET.ENCRYPTION_TYPES_CLIENT=(AES256,AES192,AES128)
```

```
SQLNET.CRYPTO_CHECKSUM_TYPES_CLIENT=(SHA1)
```

Entries for each database are needed in both primary and standby tnsnames.ora files for proper redo transport. Use the following example, replacing bolded values with values relevant to the configuration.

The TNS descriptors for the databases will be different depending on whether the scan listeners are resolvable from the other system.

The description below assumes the scan name is resolvable and can be used in the TNS descriptor. See Appendix B for an example of the TNS descriptors using ADDRESS_LIST if the scan name cannot be resolved.

#### TNSNAMES.ORA ON ON-PREMISES HOST

Add the following descriptor to the on premises tnsnames.ora files after making necessary replacements.

```
<standby db_unique_name> =
 (DESCRIPTION =
  (SDU=65536) (RECV_BUF_SIZE=134217728)
  (SEND_BUF_SIZE=134217728)
  (ADDRESS_LIST =
    (ADDRESS = (PROTOCOL = TCP)(HOST = <standby scan name>)(PORT = {1521|<port#>}))
  )
  (CONNECT_DATA =
    (SERVER = DEDICATED)
    (SERVICE_NAME = <service name of the standby database>)
  ))
```

#### TNSNAMES.ORA ON CLOUD HOSTS

Add the descriptor for the primary database to cloud tnsnames.ora files after making the necessary changes.

*Change the descriptor name for the Cloud standby to use the db_unique_name rather than the db_name*

```
<primary db_unique_name> =
 (DESCRIPTION =
  (SDU=65536) (RECV_BUF_SIZE=134217728)
  (SEND_BUF_SIZE=134217728)
  (ADDRESS_LIST =
    (ADDRESS = (PROTOCOL = TCP)(HOST = <primary scan name>)(PORT = {1521|<port#>}))
  )
  (CONNECT_DATA =
    (SERVER = DEDICATED)
    (SERVICE_NAME = <primary database service name>])
    (UR=A)
  ))
```

## Step 7: Instantiate the Standby Database

The standby database can be created from the active primary database or from a backup of the primary database. This section describes the method duplicating from the active primary database using Oracle 12.1 or higher feature RMAN 'RESTORE…FROM SERVICE'.

*RDBMS 11.2 does not support RMAN RESTORE FROM SERVICE. Backups based duplication or RMAN DUPLICATE must be used. For details refer to the documentation.*

*For additional details regarding standby instantiation see MOS 2275154.1 Creating a Physical Standby Database in an 11.2, 12.1, 12.2 or later environment.*

*Before copying the primary database ensure the best practices have been implemented in the primary database, especially the creation of the standby redo logs. If the SRLs are not created before the copy they will need to be added to the primary and standby databases separately.*

Start the standby instance (one instance for RAC)

```
$ srvctl stop database -d <standby DB_UNIQUE_NAME> -o immediate

$ rman target /
RMAN> startup nomount

RMAN> restore standby controlfile from service '<primary tns descriptor>';

RMAN> alter database mount;

RMAN> CONFIGURE DEVICE TYPE DISK PARALLELISM 4;
```

← sets the degree or parallelism (number of channels) and can be altered depending on bandwidth.

```
RMAN> restore database from service '<primary tns descriptor>' section size 64G;
```

*The length of time required to complete the copy is dependent upon the size of the database and the available bandwidth. It could take anywhere from minutes for a small database to many hours for a large database.*

```
RMAN> shutdown immediate
```

Restart the standby database

```
$ srvctl start database -d <standby DB_UNIQUE_NAME> -o mount
```

Clear all online and standby redo logs

Before clearing the logs verify:

- DB_CREATE_ONLINE_LOG_DEST_1= <DATA disk group>. Correct if necessary
- DB_CREATE_ONLINE_LOG_DEST_*n* is not set other than n=1.

Clear all logfiles:

```
$ sqlplus "/ as sysdba"
SQL> set pagesize 0 feedback off linesize 120 trimspool on
SQL> spool /tmp/clearlogs.sql
SQL> select distinct 'alter database clear logfile group '||group#||';' from v$logfile;
SQL> spool off
SQL> @/tmp/clearlogs.sql

SQL> select member from v$logfile;
```

*All redo logs should be on the DATA disk group in the standby DB_UNIQUE_NAME directory*

## Step 8: Configure Data Guard broker.

Enable the dg_broker_config_file parameters on primary and standby database.

*For ASM, place broker config files on separate disk groups. For RAC, broker config files must be on shared storage.*

**ON ON-PREMISES AND CLOUD HOSTS**

```
SQL> alter system set dg_broker_config_file1='+DATA/<db_unique_name/dr1.dat';
SQL> alter system set dg_broker_config_file2='+RECO/<db_unique_name/dr2.dat';
```

Start the Data Guard Broker Process on primary and standby database

**ON ON-PREMISES AND CLOUD HOSTS**

```
SQL> alter system set dg_broker_start=true;
SQL> show parameter dg_broker_start
NAME                      TYPE    VALUE
------------------------------------ ----------- ------------------------------
dg_broker_start              boolean     TRUE

SQL> select pname from gv$process where pname like 'DMON%';

PNAME
-----
DMON
DMON
```

Register the database via DGMGRL on primary site

**ON ON-PREMISES HOST**

```
$ dgmgrl sys/<sys password>@<net service name for primary database>

DGMGRL> CREATE CONFIGURATION <configuration_name> AS PRIMARY DATABASE IS <primary db_unique_name>
CONNECT IDENTIFIER IS <Net Service name for primary database>;

DGMGRL> ADD DATABASE <standby db_unique_name> AS CONNECT IDENTIFIER IS <Net Service name for
standby database> MAINTAINED AS PHYSICAL;

DGMGRL> enable configuration;
```

Enable flashback database on the standby:

**ON STANDBY DATABASE**

```
DGMGRL> edit database <standby> set state=apply-off;

SQL> alter database flashback on;

DGMGRL> edit database <standby> set state=apply-on;
```

## Step 9: (11g only) Remove the static listener

As grid user on the Exadata Cloud node 1, remove the static listener SID_DESC created for instantiation and reload the listener.

```
(SID_DESC =
  (GLOBAL_DBNAME = <DB_UNIQUE_NAME of the Cloud database>)
  (ORACLE_HOME = <Local Oracle Home>)
  (SID_NAME = <ORACLE SID of the local instance>)
)
```

Do NOT remove the SID_DESC for `<DB_UNIQUE_NAME>_DGMGRL`

## Step 10: Set RMAN parameters:

Replace the original RMAN configuration with the contents of `/tmp/rman_setting.log` created previously. This will most importantly replace the backup configuration with encryption, compression and retention in addition to the snapshot controlfile location. These values were lost when the standby controlfile was copied from the primary database.

# HEALTH CHECK 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:

- exachk applicable to Exadata Database Machine (suitable for Exadata Cloud Service)
- ORAchk application to all Oracle stack

Each of the automated checks includes 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.

## Monitoring

Regular monitoring of the Data Guard configuration is not provided in a Hybrid Data Guard Configuration and must be done manually. Refer to Monitoring a Data Guard Configuration (Doc ID 2064281.1) for MAA best practice recommendations for monitoring.

# 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. 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
```

*A snapshot standby database cannot be the target of a switchover or failover. A snapshot standby database must first be converted back into a physical standby database before performing a role transition to it.*

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

At any time you can manually execute a Data Guard switchover (planned event) or failover (unplanned event). 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.

## SOFTWARE UPDATES - PATCHING AND UPGRADE

With the process completed as described, the Cloud tooling can be used to patch and upgrade the standby database. The process for patching and upgrading a Hybrid Data Guard configuration is described in My Oracle Support Note 2626861.1: Patching and upgrading a Hybrid Data Guard Standby Database.

## CONCLUSION

Hybrid Data Guard using Exadata Cloud 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 PRACTICES AND 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 and characteristics should be set on both the primary and standby databases.

- ARCHIVELOG mode enabled
- Flashback database on
- FORCE LOGGING enabled
- Use SPFILE
- Use Data Guard Broker
- Online Redo Log characteristics
    - At least 1G in size for non-Exadata databases, 4G for Exadata
    - Minimum of 3 groups per thread
    - Single member groups when using high redundancy storage
    - Reside on DATA disk group when groups are single member
- Standby Redo Log characteristics
    - Identical size as online redo logs
    - For RAC, assign SRL groups to a thread
    - Same number of groups per thread as online redo log groups
    - Single member groups only
    - Reside on DATA disk group
- LOG_BUFFER = 128M for 11.2; 256M for 12.1+
- DB_BLOCK_CHECKING=NONE for primary; MEDIUM or FULL for standby
  *Note: A value of MEDIUM is suggested for the primary however, 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 ***#Exadata Only***
- DB_CREATE_ONLINE_LOG_DEST_1= DATA disk group
- DB_CREATE_ONLINE_LOG_DEST_n other than n=1 should only be set when the DATA disk group is not high redundancy; The setting should be RECO in this case. (Cloud uses High redundancy disk groups)
- DB_CREATE_FILE_DEST uses DATA disk group
- DB_RECOVERY_FILE_DEST uses RECO disk group
- Recyclebin is on

## APPENDIX B: TNSNAMES.ORA SAMPLE USING ADDRESS_LIST

The following examples of TNS descriptors using ADDRESS_LISTs which can be used in situations where the scan name cannot be resolved between the Cloud and on-premises clusters.

### STANDBY DATABASE

*No changes are necessary to the on-premises database descriptor in the on-premises tnsnames.ora*

```
<standby db_unique_name> =
  (DESCRIPTION=
  (SDU=65536) (RECV_BUF_SIZE=134217728)
  (SEND_BUF_SIZE=134217728)
    (ADDRESS_LIST=
     (FAILOVER=on)
     (CONNECT_TIMEOUT=3)(RETRY_COUNT=3)
     (ADDRESS = (PROTOCOL = TCP)(HOST = <standby node1 VIP address>)(PORT = {1521|<port#>}))
  )
     (ADDRESS = (PROTOCOL = TCP)(HOST = <standby node2 VIP address>)(PORT = {1521|<port#>}))
  )
    )
    (CONNECT_DATA=
      (SERVER=DEDICATED)
      (SERVICE_NAME= <standby database service name>)
    )
  )
```

### PRIMARY DATABASE

*Change the name of the Cloud database connect descriptor to use db_unique_name instead of db_name. Otherwise no changes.*

```
<primary db_unique_name> =
  (DESCRIPTION=
  (SDU=65536) (RECV_BUF_SIZE=134217728)
  (SEND_BUF_SIZE=134217728)
   (ADDRESS_LIST=
    (FAILOVER=on)
    (CONNECT_TIMEOUT=3)(RETRY_COUNT=3)
    (ADDRESS = (PROTOCOL = TCP)(HOST = <primary node1 VIP address>)(PORT =
{1521|<port#>}))
  )
    (ADDRESS = (PROTOCOL = TCP)(HOST = <primary node2 VIP address>)(PORT =
{1521|<port#>}))
  )
  (CONNECT_DATA =
```

```
    (SERVER = DEDICATED)
    (SERVICE_NAME = <primary database service name>)
))
```

## CONNECT WITH US

Call +1.800.ORACLE1 or visit oracle.com.
Outside North America, find your local office at oracle.com/contact.

blogs.oracle.com facebook.com/oracle twitter.com/oracle



Disaster Recovery using Exadata Cloud
March, 2020

Author: Andrew Steinorth
Contributing Author: Glen Hawkins