White Paper

Oracle Disaster Recovery solution for Dell EMC VxFlex family

Abstract

This paper suggests how best to deploy Data Guard technology with Oracle Real Application Clusters on VxFlex integrated rack for reliable disaster recovery and business continuity.

October 2019

000055

Revisions

Date	Description
October 2019	Initial release

Acknowledgements

This paper was produced by the following:

Author: Sanjay Puttaswamy

Support: Shashikir Chidambara, Neil Gerren

Other:

Table of contents

Executive summary

Enterprise organizations in today's world experience challenges like downtime, application and data growth, and high service levels. VxFlex integrated rack delivers scalability, availability, and efficiency to enterprise customer to address these and other challenges. Disaster recovery and business continuity are also critical to all global enterprises and are among the most important characteristics of database. Oracle Data Guard is one of the most effective solutions to protect any enterprise data and make it available on a 24 x 7 basis despite natural disasters and other unplanned or planned outages.

This paper discusses how best to deploy Data Guard technology with Oracle Real Application Clusters on VxFlex integrated rack for reliable disaster recovery and business continuity.

1 Introduction

This white paper demonstrates that Oracle Data Guard replication can be used to replicate databases between VxFlex integrated rack clusters that have Oracle RAC clusters configured.

1.1 Objective

The purpose of this paper is to outline how customers can deploy Oracle Data Guard for Oracle RAC databases on VxFlex integrated rack to protect their enterprise data during natural disasters and other outages.

1.2 Audience

This white paper is intended for Oracle database administrators, system engineers, partners, and the members of Dell EMC and partner professional service community who are seeking to understand the disaster recovery capability and compatibility of Oracle Data Guard on Dell EMC VxFlex integrated rack using ESXi hypervisors.

The reader of this document must have a working knowledge of Dell EMC VxFlex integrated rack, VMware vSphere technologies, Oracle database technologies, and should have a basic familiarity with storage, compute, and network technologies and topologies.

1.3 Terminology

The following table defines acronyms and terms that are used throughout this document:

Table 1 Terms and definitions

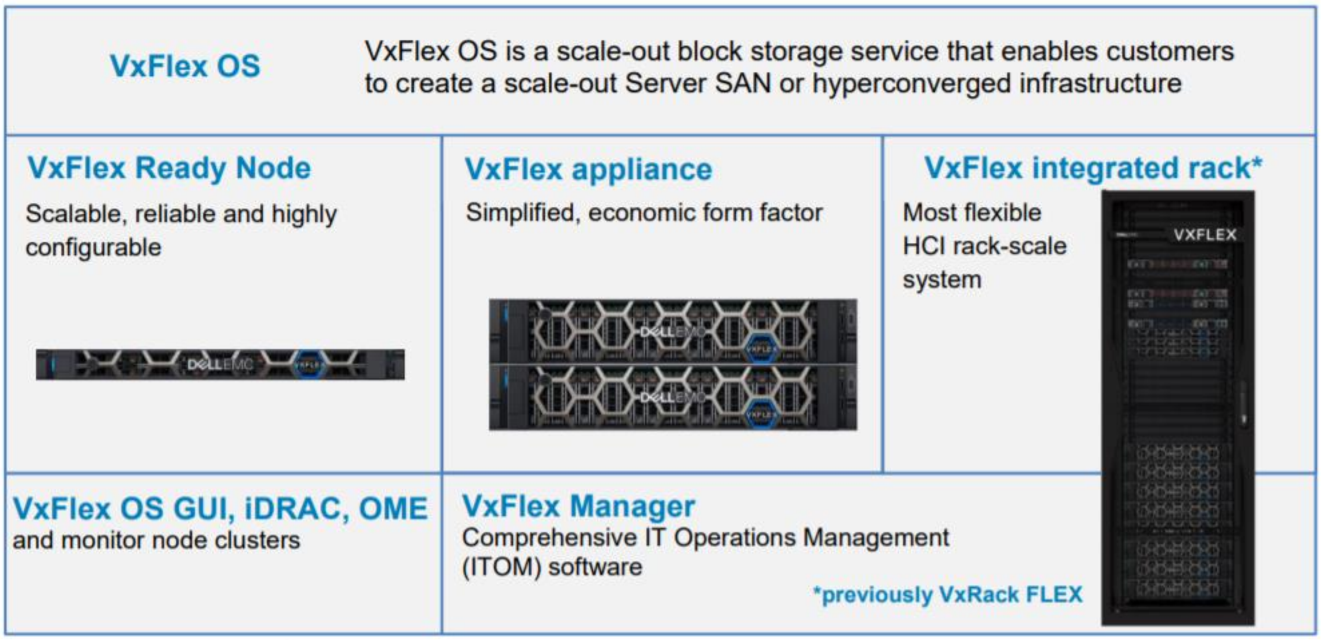
Term	Definition
MDM	Meta Data Manager
SDS	Storage Data Server
SDC	Storage Data Client
SVM	Storage Virtual Machine
OS	Operating System
RCM	Release Certification Matrix
SSD	Solid-State Drive
RAC	Real Application Clusters
ASM	Automatic Storage Management
DG	Data Guard
MRP	Managed Recovery Process
RFS	Remote File Server
ARCH	Archiver Process

2 Product overview

2.1 VxFlex family

VxFlex family helps transforming from a traditional three-tier architecture to a modern data center without any trade-offs, meeting business requirements without compromise. The central software layer for VxFlex family is VxFlex OS, scale-out block storage service that enables customers to create a scale-out Server SAN or hyperconverged infrastructure. The VxFlex family currently include VxFlex Ready Nodes, VxFlex appliance, and VxFlex integrated rack.

Figure 1 VxFlex family



2.1.1 VxFlex integrated rack

VxFlex integrated rack is a rack-scale engineered system, with integrated networking, that provides linear scalability and enterprise-grade availability. VxFlex integrated rack is engineered, manufactured, managed, supported, and sustained as one system for single end-to-end lifecycle support.

VxFlex integrated systems create a server-based SAN by combining virtualization software, which is known as VxFlex OS, with Dell EMC PowerEdge servers to deliver flexible, scalable performance and capacity on demand. Local storage resources are combined to create a virtual pool of block storage with varying performance tiers. The VxFlex integrated rack enables you to scale from a small environment to enterprise scale with over a thousand nodes. In addition, it provides enterprise grade data protection, multi-tenant capabilities, and add-on enterprise features such as QoS, thin provisioning, and snapshots. VxFlex systems deliver the performance and time-to-value required to meet the demands of the modern enterprise data center.

2.1.2 VxFlex appliance

VxFlex appliance is a preconfigured and validated for fast, easy deployment, VxFlex appliance offers a turnkey experience in an economic form factor. With VxFlex appliance, customers benefit from a smaller starting point, with massive scale potential, without having to compromise on performance and resiliency.

2.1.3 VxFlex Ready Nodes

VxFlex Ready Nodes combine Dell EMC PowerEdge servers that are powered by Intel® Xeon® Scalable Processors and VxFlex OS software to create scalable, reliable, and easy-to-deploy building blocks for hyperconverged or server SAN architecture, multi-hypervisor or bare metal environments, and high-performance databases.

2.1.4 VxFlex OS

VxFlex OS is software that creates a server and IP-based SAN from direct-attached storage to deliver flexible and scalable performance and capacity on demand. As an alternative to a traditional SAN infrastructure, VxFlex OS combines HDD, SSD, and NVMe media to create virtual pools of block storage with varying performance tiers. VxFlex OS provides enterprise-grade data protection, multi-tenant capabilities, and add-on enterprise features such as QoS, thin provisioning, and snapshots. VxFlex OS supports physical and virtualized servers, has been proven to deliver significant TCO savings vs. traditional SAN.

VxFlex OS consists of three primary components: Meta Data Manager (MDM), Storage Data Client (SDC) and Storage Data Server (SDS). The MDMs work the brain of the system and are responsible for managing metadata and core functions such as automated rebuild and rebalance, which ensure data access as media and servers fail. The SDC runs like an agent or daemon on a server and acts like a highly available HBA to connect to storage cluster to consume the storage as required by the application workload. The SDSs are daemons that contribute storage to the storage cluster. SDCs communicate directly with the SDSs. When an SDC gets an I/O request from the application, it sees the cached metadata map, and sends the request directly to the SDS, which contains the requested data.

2.1.5 VxFlex Manager

VxFlex Manager is a unified management and automation solution for the VxFlex integrated rack and the VxFlex appliance that enables template-based provisioning of infrastructure and workloads. It provides monitoring, alerting, and health status of hardware and services. It simplifies and automates lifecycle management of diverse and heterogeneous IT resources. It enables users to respond rapidly and reliably to dynamic business needs.

VxFlex Manager is a comprehensive IT Operations Management (ITOM) software purpose that is built for VxFlex to automate and simplify implementation, expansion, and lifecycle management.

VxFlex Manager brings together multiple management consoles, workflow automation, and an intuitive interface that allows customers to monitor, manage, deploy, and maintain physical and virtual resources with the click of a button. Key tenets of the VxFlex Manager architecture include:

- System assurance: compliancy and non-disruptive remediation
- Insights: monitoring, alerting, and health checks
- Implementation simplification: simplified and automated system deployment and workflows
- Node serviceability: single-button take node out of and back into service
- Hypervisor update: single-button upgrade of hypervisor

2.1.6 VxFlex OS management

VxFlex OS management is available using a GUI, CLI, and REST clients. There is a VMware vSphere® plug-in that allows VMware admins to deploy, upgrade, configure, and manage VxFlex OS in an ESXi environment within VMware vSphere.

2.2 Oracle RAC

Oracle Database 12c and 18c Enterprise Edition provide efficient, reliable, and secure data management for mission-critical transactional applications, query-intensive data warehouses, and mixed workloads. Oracle Real Application Clusters (RAC) provides the processing power of multiple, interconnected servers on a cluster; enables access to a single database from multiple servers within the cluster. This design insulates both applications and database users from server failures, while providing performance that scales out on-demand at low cost; and is a vital component of grid computing that enables multiple servers to share a database. Oracle Database 12c and 18c include Automatic Storage Management (ASM) and Oracle Clusterware. Combining the use of ASM and Oracle Clusterware virtualizes storage, database servers, application servers, holistic management, and all the other aspects related to deploying and managing a virtualized IT environment.

Oracle ASM is Oracle's recommended storage management solution that provides an alternative to conventional volume managers, file systems, and raw devices. Oracle ASM is a volume manager and a file system for Oracle Database files that supports single-instance Oracle Database and Oracle Real Application Clusters (Oracle RAC) configurations.

ASMLib, Udev, and the ASM filter drivers are all supported, although, Dell EMC recommends only using the ASM Filter Driver with 12c or greater. Use the solution that best meets your business and technical needs. ASMLIB is generally aligned with Oracle Linux and/or the Unbreakable Enterprise Kernel (UEK), while Udev is customarily used with other UNIX distributions. The new filter driver seeks to prevent unintentional out-of-band writes to devices allocated to your database. It also supports TRIM and provides integration with other storage APIs as they develop.

Oracle ASM uses disk groups to store datafile; an Oracle ASM disk group is a collection of disks that Oracle ASM manages as a unit. Within a disk group, Oracle ASM exposes a file system interface for Oracle Database files. The content of files that are stored in a disk group is evenly distributed to eliminate hot spots and to provide uniform performance across the disks. The performance is comparable to the performance of raw devices.

2.3 Oracle Data Guard

Oracle Data Guard is designed to integrate with other Oracle software to provide disaster recovery for Oracle databases. When the primary database becomes unavailable, Data Guard can quickly switch a standby database to the production role, helping significantly reduce downtime caused during primary site outages. Implementing Data Guard for an Oracle RAC database can provide end-to-end data protection and high-availability capabilities.

A Data Guard–based configuration consists of one primary database and from one to thirty standby databases. All systems in this configuration must run an Oracle image built for the same platform. The primary and standby databases can be single- or multi-instance Oracle RAC databases. Databases within a Data Guard–based configuration connects and communicate with one another through Oracle Net over a network. Data Guard synchronizes the primary and standby databases by automatically transmitting redo data from the primary database to the standby databases and applying redo data on the standby databases.

Each Data Guard standby database can be either a physical standby database or a logical standby database. A physical standby database is a physically identical copy of the primary database, which Data Guard synchronizes through media recovery. A logical standby database contains the same logical information as the primary database, but the physical structure of the database can be different. In this case, Data Guard converts the redo log data into SQL statements and executes these statements on the standby database.

2.3.1 Data Guard Protection modes

A synchronized RAC standby database is meant to be one that meets the minimum requirements of the configured data protection mode and that does not have a redo gap. Oracle Data Guard includes three protection modes based on performance, availability, and data protection.

Maximum Performance mode

This is the default protection mode and provides the highest level of data protection that is possible without affecting the performance or availability of the primary database. With this protection mode, a transaction is committed as soon as the redo data are needed to recover the transaction is written to the local (online) redo log. Using this configuration, the primary database writes its redo stream to the standby redo logs on the standby database asynchronously for the commitment of the transactions that create the redo data.

Maximum Availability mode

This protection mode provides the highest level of data protection that is possible without affecting the availability of the primary database. This protection mode is similar to Maximum Protection where a transaction will not commit until the redo data needed to recover that transaction is written to both the local (online) redo log and to at least one remote standby redo log. Unlike Maximum Protection mode; however, the primary database does not shut down if a fault prevents it from writing its redo data to a remote standby redo log. Instead, the primary database operate in Maximum Performance mode until the fault is corrected and all log gaps have been resolved. After all log gaps have been resolved, the primary database automatically resumes operating in Maximum Availability mode.

Maximum Protection mode

This protection mode offers the ultimate in data protection. It guarantees no data loss occurs in the event the primary database fails. To provide this level of protection, the redo data that are needed to recover each transaction must be written to both the local (online) redo log and to a standby redo log on at least one standby database before the transaction can be committed. In order to guarantee no loss of data can occur, the primary database shut down if a fault prevents it from writing its redo data to at least one remote standby redo log. In a multiple-instance RAC database environment, Data Guard will shut down the primary database if it is unable to write the redo data to at least one properly configured database instance.

Prerequisites for the Data Guard protection modes:

Table 2 Prerequisites

Requirement	Maximum Protection	Maximum Availability	Maximum Performance
Redo Archival Process	LGWR	LGWR	LGWR or ARCH
Network Transmission mode	SYNC	SYNC	ASYNC when using LGWR
Disk Write Option	AFFIRM	AFFIRM	NOAFFIRM
Standby Redo Logs	Yes	Yes	Yes for Real-Time Apply

3 Solution architecture

The solution that is provided in this paper is deployed with ESXi hyperconverged configuration on VxFlex integrated rack.

3.1 Logical architecture

Four nodes VxFlex cluster that is configured at the Production site and another four nodes VxFlex cluster that is configured at the Standby site. Each physical server included two Intel Xeon Skylake 14-core processors, 384 GB RAM, and ten 1.92 TB SSDs. Each VxFlex integrated rack node has a Dell EMC Storage Virtual Machine (SVM) running on it, providing both storage clustering and storage services.

The following figure provides an overview of the logical architecture of Oracle Data Guard configuration on VxFlex integrated rack:

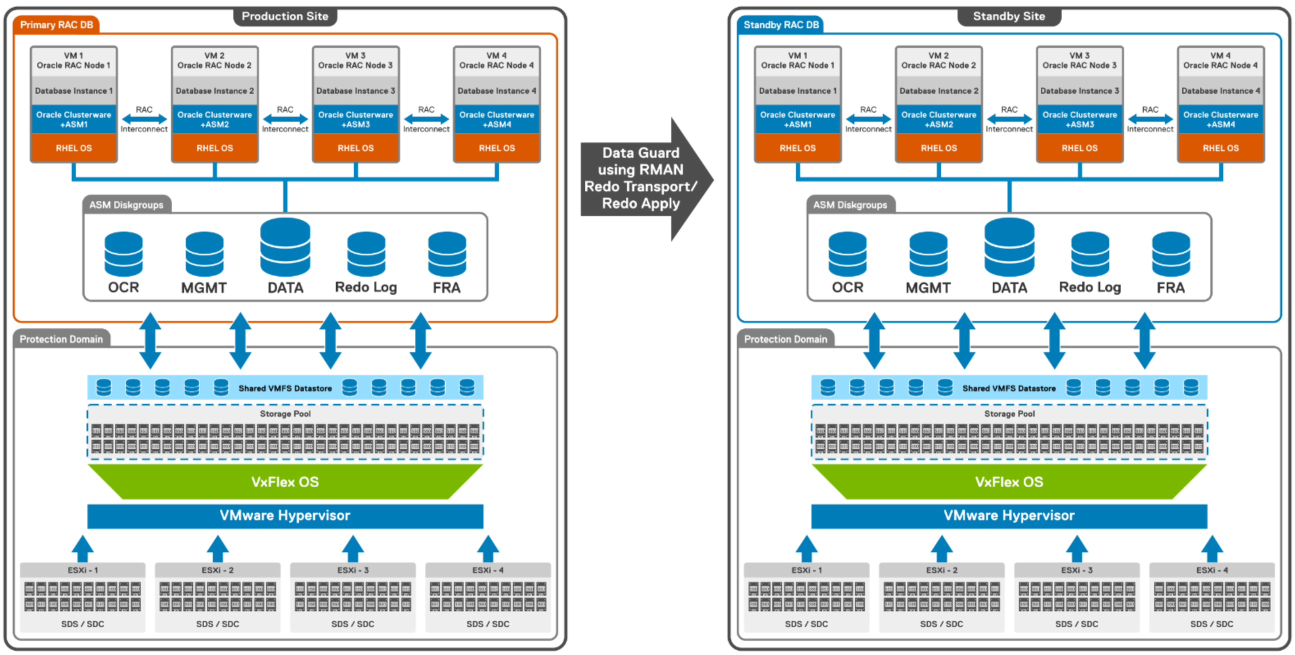


Figure 1 VxFlex integrated rack with Oracle Data Guard

Storage

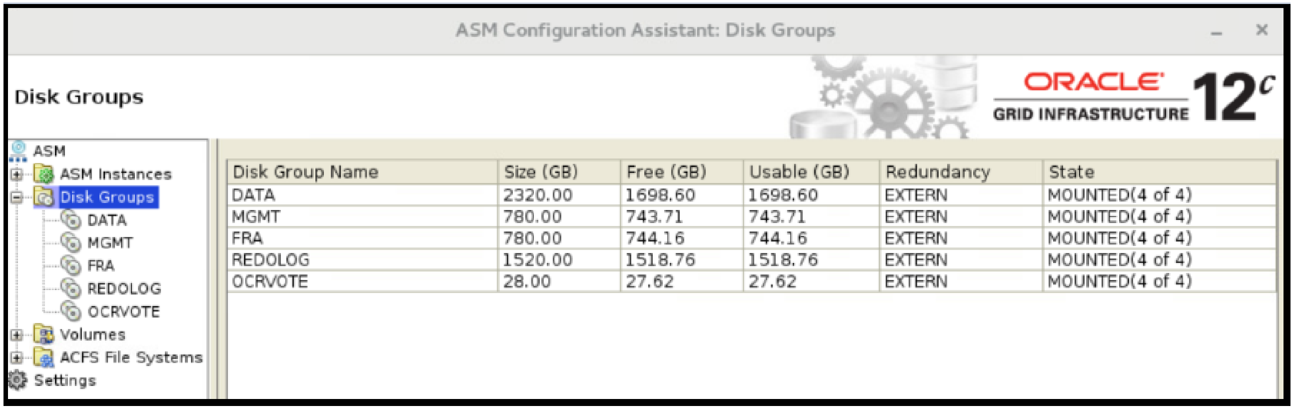
In the VxFlex integrated rack with VMware vSphere 6.5 (HCI) environment, the MDM, and SDS components are installed on a dedicated SVM, whereas the SDC is installed directly on the ESXi host. The VxFlex back-end storage consists of four SDS systems. The Storage Data Server (SDS) aggregates and serves raw local storage in a given node and shares that storage as part of VxFlex OS cluster. A single protection domain is carved out of the SSDs on these four SDS systems. A single storage pool is configured, and multiple volumes are carved out to meet the Oracle RAC database requirements including volumes for data, logging, cluster registry, and flash recovery area. These volumes are mapped to the ESXi cluster and added as datastores and later mapped as disks drive to Oracle database virtual machine using VMware Paravirtual SCSI (PVSCSI) adapters.

Guest VM configuration

When configuring Linux guest VM for Oracle RAC database, ensure that the hardware, network, storage, and software requirements are supported before installing the Oracle RAC. Created four Red Hat Enterprise Linux virtual machines on both primary and standby sites and deployed Oracle Grid Infrastructure on the vSphere clusters. The guest VMs used here consist of 16 virtual CPUs, 128 GB RAM, and thick provisioned disks on VxFlex backend storage. For Oracle RAC, the RAC-instance VMs should run on different ESXi hosts. Affinity rules should be set so that a RAC instance VM always runs on an assigned VxFlex cluster node.

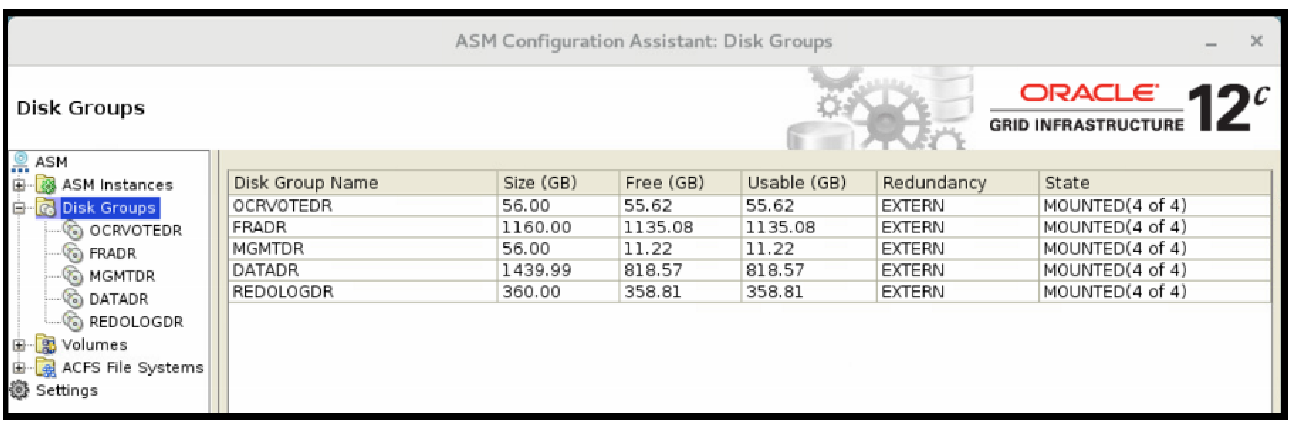
Storage layout for Oracle RAC Database

Oracle ASM volume manager was used as the volume manager for the Oracle database files. The disk groups were created using “EXTERNAL” redundancy. EXTERNAL redundancy was used since VxFlex OS uses mesh-mirroring to protect data by default. Unique ASM disk group names are assigned for the primary and standby RAC database.



Disk Group Name	Size (GB)	Free (GB)	Usable (GB)	Redundancy	State
DATA	2320.00	1698.60	1698.60	EXTERN	MOUNTED(4 of 4)
MGMT	780.00	743.71	743.71	EXTERN	MOUNTED(4 of 4)
FRA	780.00	744.16	744.16	EXTERN	MOUNTED(4 of 4)
REDOLOG	1520.00	1518.76	1518.76	EXTERN	MOUNTED(4 of 4)
OCRVOTE	28.00	27.62	27.62	EXTERN	MOUNTED(4 of 4)

Figure 2 ASM disk groups on RAC primary database



Disk Group Name	Size (GB)	Free (GB)	Usable (GB)	Redundancy	State
OCRVOTEDR	56.00	55.62	55.62	EXTERN	MOUNTED(4 of 4)
FRADR	1160.00	1135.08	1135.08	EXTERN	MOUNTED(4 of 4)
MGMTDR	56.00	11.22	11.22	EXTERN	MOUNTED(4 of 4)
DATADR	1439.99	818.57	818.57	EXTERN	MOUNTED(4 of 4)
REDOLOGDR	360.00	358.81	358.81	EXTERN	MOUNTED(4 of 4)

Figure 3 ASM disk group on RAC standby database

3.2 Network architecture

The following figure provides an overview of the network architecture and design of VxFlex integrated rack for both the RAC primary and RAC standby using Oracle Data Guard.

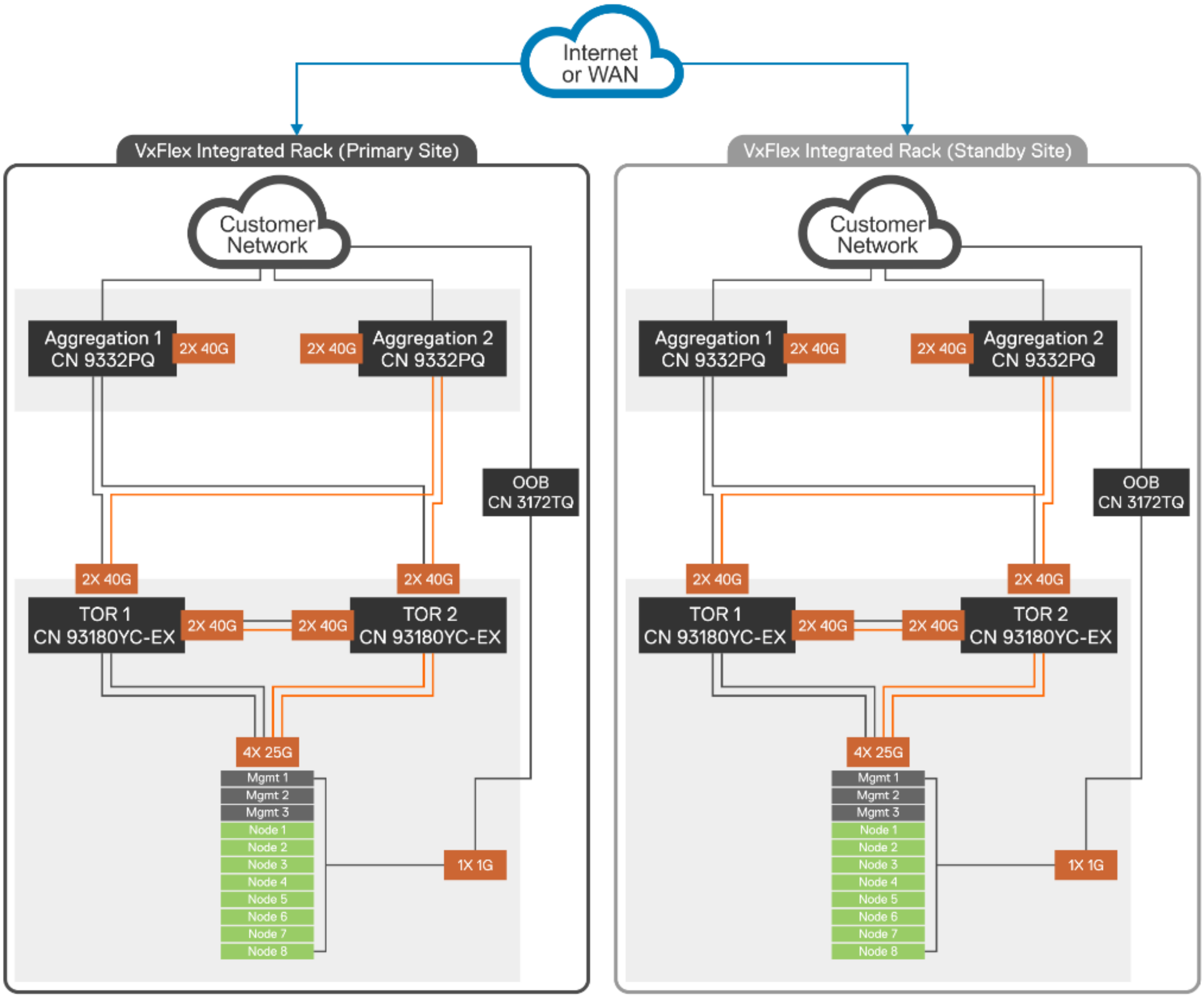


Figure 4 Network architecture

The following table contains the network details of VxFlex integrated rack

Table 3 VxFlex networking details

Components	Description
Cisco Nexus 93180YC-EX	10 Gbps & 25 Gbps TOR switches
Cisco Nexus 9332PQ	40 Gbps Aggregation switches
Cisco Nexus 3172TQ	1 Gbps & 10 Gbps Management switches.
Data Domain	2 x 10 Gbps links
Application traffic	2 x 25 Gbps links
VxFlex storage traffic	2 x 25 Gbps links

Included Components and architecture:

- VxFlex integrated rack uses a pair of Cisco Nexus 93180YC-EX as TOR switches
- VxFlex integrated rack uses a pair of Cisco Nexus 9332PQ as Aggregation switches
- Cisco Virtual Port Channel is configured between the TOR and Aggregation switches
- Cisco Nexus 3172TQ switch is used for OOB traffic with 1GbE dedicated network
- Each node comprises of four 25GbE ports, two ports connected to each TOR (Cisco Nexus 93180YC-EX) switch
- Both TOR (Cisco Nexus 93180YC-EX) switches have uplinks to aggregation (Cisco Nexus 9332PQ) switches for redundancy and network bandwidth aggregation
- Both RAC primary and RAC standby sites are connected through WAN

4 Oracle Data Guard configuration

Oracle Standby database can be created either using traditional or RMAN method. In traditional method, the process involves making a full cold backup of production RAC database on the primary site and copied to the disaster recovery site. In RMAN method, the physical standby database can be created on disaster recovery site without any production database backup.

4.1 Oracle Data Guard using RMAN

This section provides step-by-step instructions to create a RAC standby database in maximum performance mode using asynchronous redo transport and real-time apply.

1. Mandate the following prerequisites are met for RAC primary database and RAC physical standby database:

    - Ensure the force logging enabled at RAC primary database.
      `SQL> ALTER DATABASE FORCE LOGGING;`
    - Ensure that RAC primary database is in Archivelog mode.
    - Create RAC primary database with Oracle ASM.
    - RAC standby database is created on ASM.
    - RAC standby nodes that already installed with Grid Infrastructure and Oracle database software.

Note: Ensure DB_UNIQUE_NAME (must be unique), instance names, TNS, and ASM disk groups at the RAC standby site.

Table 4 Environment details of primary and standby databases

	Primary RAC database	**Standby RAC database**
Operating System	Red Hat Linux 7.5	Red Hat Linux 7.5
Oracle Release	12.2.0.1	12.2.0.1
Nodes	oradb1, oradb2, oradb3, oradb4	oradr1, oradr2, oradr3, oradr4
Database Name	hdb	hdb
Database Unique Name	hdb	stbhdb
Instances	hdb1, hdb2, hdb3, hdb4	stbhdb1, stbhdb2, stbhdb3, stbhdb4
SCAN Name	oradb-scan	oradr-scan
Disk Groups	DATA, REDOLOG, FRA	DATADR, REDOLOGDR, FRADR

2. Configure Oracle Data Guard parameters at primary RAC site as shown in the below table:

`log_archive_config='DG_CONFIG=(hdb,stbhdb)'`
`log_archive_dest_1='location=USE_DB_RECOVERY_FILE_DEST arch reopen=60 max_failure=0 mandatory valid_for=(ALL_LOGFILES,ALL_ROLES) db_unique_name=hdb'`
`log_archive_dest_2='service=stbhdb LGWR ASYNC NOAFFIRM max_failure=10 max_connections=2 reopen=400 valid_for=(online_logfiles,primary_role) db_unique_name=stbhdb'`
`log_archive_max_processes=12`
`fal_server=stbhdb1, stbhdb2, stbhdb3, stbhdb4`
`db_file_name_convert='+DATA','+DATADR'`
`log_file_name_convert='+REDOLOG','+REDOLOGDR'`
`standby_file_management=AUTO`

3. Create Standby Redo Logs to enable the changes occurring in the RAC primary database to be reflected in almost real time in the RAC standby, a concept known as Real Time Apply (RTA). So, we create a standby redo logs on the RAC primary database. There should be minimum of (threads)*(groups per threads + 1) standby redo logs created on the RAC standby database.

4. Copy password file from RAC primary node to RAC standby nodes. Use **pwcopy** command to copy password file from ASM to file system.

5. Configure Oracle Net on the RAC primary and RAC standby systems by adding both the primary and standby database Transparent Network Substrate (TNS) aliases in the **tnsnames.ora** files. A listener static entry must be defined for RAC standby instance on first cluster node because this is required by RMAN active duplication when running on first cluster node.

   Add the following entry in **listener.ora** and reload the listener configuration:

```
SID_LIST_LISTENER =
 (SID_LIST =
   (SID_DESC =
     (GLOBAL_DBNAME = aux)
     (ORACLE_HOME=/u01/app/oracle/product/12.2.0.1/db_1)
     (SID_NAME = stbhdb1)
    )
  )
```

6. Create a parameter file for the standby instance on first standby cluster node. Add a single parameter `DB_NAME='aux'` in the parameter file. Create `AUDIT_DEST_DUMP` directory on each standby cluster node.

7. Set up the appropriate environment variables on the RAC standby node, such as `ORACLE_SID`, `ORACLE_BASE`, and `ORACLE_HOME`. Startup the RAC standby instance on the first standby node in `NOMOUNT` state.

8. Connect to RMAN from RAC primary node to create a RAC standby database using the RMAN command **Duplicate target database for standby from active database**.

```
$ rman

Recovery Manager: Release 12.2.0.1.0 - Production on Thu Jul 4 20:48:27 2019

Copyright (c) 1982, 2017, Oracle and/or its affiliates.  All rights reserved.

RMAN> connect target sys/oracle1@hdb;

connected to target database: HDB (DBID=1837129077)

RMAN> connect auxiliary sys/oracle1@oradr1:1521/aux

connected to auxiliary database: AUX (not mounted)

RMAN> run {
allocate channel p1 type disk;
allocate auxiliary channel s1 type disk;
duplicate target database for standby from active database
spfile
 set db_unique_name='stbhdb'
 set instance_name='stbhdb1'
 set instance_number='1'
 set compatible='12.2.0.1'
 set control_files='+DATADR','+FRADR'
 set db_file_name_convert='+DATA','+DATADR'
 set log_file_name_convert='+REDOLOG','+REDOLOGDR'
 set db_recovery_file_dest='+FRADR'
 set DB_CREATE_ONLINE_LOG_DEST_1='+REDOLOGDR'
 set DB_CREATE_ONLINE_LOG_DEST_2='+FRADR'
 set DB_CREATE_ONLINE_LOG_DEST_3='+MGMTDR'
 set fal_server='hdb'
 set log_archive_dest_1='location=USE_DB_RECOVERY_FILE_DEST arch reopen=60 max_failure=0 mandatory valid_for=(ALL_LOGFILES,ALL_ROLES) db_unique_name=stbhdb'
 set log_archive_dest_2='service=hdb LGWR ASYNC NOAFFIRM max_failure=10 max_connections=2 reopen=400 valid_for=(online_logfiles,primary_role) db_unique_name=hdb';

}
```

9. Verify the database role after successful duplication.

```
SQL> select name, db_unique_name, database_role, open_mode from v$database;

NAME      DB_UNIQUE_NAME                 DATABASE_ROLE    OPEN_MODE
--------- ------------------------------ ---------------- --------------------

HDB       stbhdb                         PHYSICAL STANDBY MOUNTED
```

10. Set `DB_CREATE_FILE_DEST` and instance initialization parameters since that `DB_CREATE_FILE_DEST` parameter cannot be set together with `DB_FILE_NAME_CONVERT` during RMAN active duplication (run script). RAC standby database instance parameters must be configured manually by changing specific RAC parameters in the common SPFILE (`INSTANCE_NUMBER, INSTANCE_NAME`).

11. RAC standby database and its database instances must be added manually to Oracle Cluster Registry (OCR) as new resources with **srvctl** command.

```
$ srvctl add database -db stbhdb -oraclehome $ORACLE_HOME -role physical_standby
-startoption mount -spfile +DATADR/stbhdb/PARAMETERFILE/spfile.269.1012815527
$ srvctl add instance -db stbhdb -instance stbhdb1 -node oradr1
$ srvctl add instance -db stbhdb -instance stbhdb2 -node oradr2
$ srvctl add instance -db stbhdb -instance stbhdb3 -node oradr3
$ srvctl add instance -db stbhdb -instance stbhdb4 -node oradr4

$ srvctl config database -d stbhdb
Database unique name: stbhdb
Database name:
Oracle home: /u01/app/oracle/product/12.2.0.1/db_1
Oracle user: oracle
Spfile: +DATADR/stbhdb/PARAMETERFILE/spfile.269.1012815527
Password file:
Domain:
Start options: mount
Stop options: immediate
Database role: PHYSICAL_STANDBY
Management policy: AUTOMATIC
Server pools:
Disk Groups:
Mount point paths:
Services:
Type: RAC
Start concurrency:
Stop concurrency:
OSDBA group: dba
OSOPER group: oper
Database instances: stbhdb1,stbhdb2,stbhdb3,stbhdb4
Configured nodes: oradr1,oradr2,oradr3,oradr4
Default network number for database services:
Database is administrator managed
```

12. Startup all the RAC standby instance.

```
$ srvctl start database -d stbhdb
$ srvctl status database -d stbhdb
Instance stbhdb1 is running on node oradr1
Instance stbhdb2 is running on node oradr2
Instance stbhdb3 is running on node oradr3
Instance stbhdb4 is running on node oradr4
```

13. Start recovery on RAC standby database after successful RMAN duplicate and the configuration related to cluster. Start the MRP (Media Recovery Process) on any one node of the RAC standby database.

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

14. Verify the log synchronization between RAC primary database and RAC standby database.

RAC primary database:

```
SQL> select open_mode,database_role from v$database;

OPEN_MODE            DATABASE_ROLE
-------------------- ----------------
READ WRITE           PRIMARY

SQL> SELECT MAX(SEQUENCE#),THREAD# FROM V$ARCHIVED_LOG WHERE RESETLOGS_CHANGE# =
(SELECT MAX(RESETLOGS_CHANGE#) FROM V$ARCHIVED_LOG) GROUP BY THREAD#;

MAX(SEQUENCE#)    THREAD#
-------------- ----------
           172          1
           132          2
           146          4
           169          3
```

RAC standby database:

```
SQL> select open_mode,database_role from v$database;

OPEN_MODE            DATABASE_ROLE
-------------------- ----------------
READ ONLY WITH APPLY PHYSICAL STANDBY

SQL> SELECT MAX(SEQUENCE#), THREAD# FROM V$ARCHIVED_LOG WHERE RESETLOGS_CHANGE# =
(SELECT MAX(RESETLOGS_CHANGE#) FROM V$ARCHIVED_LOG) GROUP BY THREAD#;

MAX(SEQUENCE#)    THREAD#
-------------- ----------
           172          1
           132          2
           146          4
           169          3
```

4.2 Data Guard validation

A RAC physical standby database is physically identical to the RAC primary database on a block-for-block basis. Redo Apply uses a media recovery process to apply redo at standby arriving from the primary. This technique works regardless of the data types in the database and for all DML and DDL operations that occurred on the primary site.

4.2.1 Data Guard for DDL operation

The following steps show DDL operation performed on RAC primary gets created on RAC standby by Redo Apply.

1. On RAC primary database, create a tablespace and confirm the tablespace creation.

```
SQL> create tablespace DRTEST datafile '+DATA' size 100m;

Tablespace created.

SQL>  select name from v$tablespace;

NAME
------------------------------
SYSTEM
SYSAUX
UNDOTBS1
TEMP
UNDOTBS2
UNDOTBS3
UNDOTBS4
USERS
TPCCTAB_OL
TPCCTAB
UNDOTBS5
DRTEST

12 rows selected.
```

2. Ensure that RAC standby database in recovery mode.

```
SQL> select open_mode,database_role from v$database;

OPEN_MODE            DATABASE_ROLE
-------------------- ----------------
READ ONLY WITH APPLY PHYSICAL STANDBY
```

3. Verify the RAC standby alert log for data file added during media recovery.

```
RAC Standby database alert.log:
------------------------------
2019-07-23T04:27:28.837424-04:00
Successfully added datafile 13 to media recovery
Datafile #13: '+DATADR/STBHDB/DATAFILE/drtest.273.1014352049'
```

4. Confirm the DRTEST tablespace creation at the RAC standby database site.

```
SQL> select name from v$tablespace;

NAME
------------------------------
SYSTEM
SYSAUX
UNDOTBS1
TEMP
UNDOTBS2
UNDOTBS3
UNDOTBS4
USERS
TPCCTAB_OL
TPCCTAB
UNDOTBS5
DRTEST

12 rows selected.
```

4.2.2 Data Guard for DML operation

Create objects from RAC primary database instance and check those objects in RAC standby database.

1. Create a table in RAC primary database and insert data to the table and commit the operation.

```
SQL> select open_mode,database_role from v$database;

OPEN_MODE            DATABASE_ROLE
-------------------- ----------------
READ WRITE           PRIMARY

SQL> create table tab_primary (no number, name varchar2(10)) tablespace DRTEST;

Table created.

SQL> insert into tab_primary values (1,'VxFlex');

SQL> insert into tab_primary values (2,'VMware');

SQL> insert into tab_primary values (3,'OracleRAC');

SQL> commit;

Commit complete.
```

```
SQL> select * from tab_primary;

        NO NAME
---------- ----------
         1 VxFlex
         2 VMware
         3 OracleRAC
```

2. Check the object with rows from RAC standby database.

```
SQL> select open_mode,database_role from v$database;

OPEN_MODE            DATABASE_ROLE
-------------------- ----------------
READ ONLY WITH APPLY PHYSICAL STANDBY

SQL> select * from tab_primary;

        NO NAME
---------- ----------
         1 VxFlex
         2 VMware
         3 OracleRAC
```

4.2.3 Data Guard Switchover

A Switchover is a two-way process in Oracle Data Guard and is a role reversal between the RAC primary database and its RAC standby database. In role transition, the current RAC primary database starts behaving as a RAC standby and on the other hand, the current RAC standby database starts behaving as a RAC primary database. Bystander standbys receive redo from the new primary database.

1. Verify the RAC standby database is ready for switchover. The “Verify” functionality does not perform the switchover but does verify if the switchover can be performed to the target RAC standby database.

```
SQL> alter database switchover to stbhdb verify;

Database altered.
```

   If there is an error message during switchover then, verify to fix the errors before proceeding to switchover operation.

2. Run the Switchover command on the RAC primary database.

```
SQL> alter database switchover to stbhdb;

Database altered.
```

   The above command switches first the RAC primary database to the RAC standby database and on the RAC standby server, switches the RAC standby database to RAC primary database.

3. Check the status of the databases. The former RAC standby database (stbhdb) is now behaving as a RAC primary database.

```
SQL> SELECT name, db_unique_name, log_mode, open_mode, database_role FROM v$database;

NAME      DB_UNIQUE_NAME  LOG_MODE     OPEN_MODE            DATABASE_ROLE
--------- --------------- ------------ -------------------- ----------------
HDB       stbhdb          ARCHIVELOG           MOUNTED           PRIMARY
```

Open the new RAC primary database.

```
SQL> alter database open;

Database altered.
```

On the other hand, the former RAC primary database (hdb) is behaving as a RAC physical standby database.

```
SQL> SELECT name, db_unique_name, log_mode, open_mode, database_role FROM v$database;

NAME      DB_UNIQUE_NAME  LOG_MODE     OPEN_MODE            DATABASE_ROLE
-------- --------------- ------------ -------------------- ----------------

HDB        hdb          ARCHIVELOG     MOUNTED          PHYSICAL STANDBY
```

4. Check the status of the databases. The former RAC standby database (stbhdb) is now behaving as a RAC primary database.

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

SQL> SELECT process, thread#, sequence#, status FROM v$managed_standby
 WHERE client_process='ARCH' or process='MRP0' ORDER BY process;

PROCESS      THREAD#  SEQUENCE# STATUS
--------- ---------- ---------- ------------
ARCH               3        528 CLOSING
ARCH               2        453 CLOSING
ARCH               4        570 CLOSING
ARCH               4        573 CLOSING
ARCH               1        578 CLOSING
ARCH               3        524 CLOSING
ARCH               1        579 CLOSING
ARCH               2        452 CLOSING
MRP0               4        575 APPLYING_LOG
```

The above output shows that Managed Recovery Process is now running on the new RAC standby database “hdb” with sequence 575 being applied.

4.2.4 Data Guard Failover

Failover is a process where RAC primary database goes down or lost due to some reasons and need to convert existing RAC standby database to start behaving as RAC primary database.

1. Shut down the RAC primary database to test failover process.

   ```
   SQL> shutdown abort
   ```

   Note: If the RAC primary database can be mounted, it may be possible to flush any unsent archived and current redo from the RAC primary database to the RAC standby database. If this operation is successful, a zero data loss failover is possible even if the RAC primary database is not in a zero data loss data protection mode.

2. The below statement flushes any unsent redo from the primary database to the standby database and waits for that redo to be applied to the standby database.

   ```
   SQL> alter system flush redo to hdb;
   ```

   Ensure only one instance is up and mounted at the primary site. And also ensure the target RAC standby database is in recovery mode (Redo Apply).

3. Resolve any archived redo log gaps. Query the `V$ARCHIVE_GAP` view on the RAC standby database to determine if there are any redo gaps on the RAC standby database.

   ```
   SQL> select thread#, low_sequence#, high_sequence# from v$archive_gap;
   ```

4. Stop Redo Apply on RAC standby database.

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

5. Switch the RAC physical standby database to the primary role.

   ```
   SQL> alter database failover to hdb;
   ```

6. Open the new primary database.

   ```
   SQL> alter database open;
   ```

5 Conclusion

This white paper demonstrates that Oracle Data Guard replication can be used to replicate databases between VxFlex integrated rack clusters that have Oracle RAC clusters configured. The validation tests that is described in this white paper ensures that database replication is not being constrained by any bottlenecks. The Oracle Data Guard switchover, failover, or database media recovery performed on VxFlex intergrade rack provides higher availability in the case of an unplanned or planned outage, which help enterprises to meet the Service Level Agreements associated with recovery time objectives.

A Configuration details

The following table contains the hardware and software components details used both on RAC primary and RAC standby site.

Table 5 Component table

Component	Description
VxFlex integrated rack	**8 x VxFlex Nodes (R640 servers):** • VxFlex OS version: R2_6.1 • ESXi version: 6.5 • CPU: 2 x 14 core Intel(R) Xeon(R) Gold 6132 (2.60 GHz) • Memory: 384-GB RAM ((12 x 32-GB DIMMs) • Storage: 10 x 1.92 TB SSDs (12 Gbps)
Network	2 NIC cards, each having 2 ports 25 GbE connection
Oracle Guest VM configuration	**4 x Linux VMs:** • Operating system version: Red Hat Enterprise Linux Server release 7.5 (Maipo) • VM version: 13 • vCPU: 16 • Memory: 128 GB
SVM configuration	**4 x Linux VMs:** • Operating system version: SUSE Linux Enterprise 12 • VM version: 11 • vCPU: 8 • Memory: 10 GB
Oracle software	• Oracle 12c Release 2 Grid Infrastructure and ASM • Oracle 12c Release 2 RAC database

B Technical support and resources

- Dell.com/support is focused on meeting customer needs with proven services and support.
- Dell EMC VxFlex product overview and White papers on DellEMC.com provides expertise that helps to ensure customer success on Dell EMC VxFlex family.

B.1 Related resources

- Oracle RAC on VxFlex Integrated Rack provides best practices to deploy Oracle RAC on VxFlex family.
- Oracle Data Guard Concepts and Administration provides concepts and the creation, maintenance and monitoring of Oracle Data Guard configurations.