Reference Architecture

HPE REFERENCE ARCHITECTURE FOR ORACLE DATABASE 12C ON HPE SIMPLIVITY 380 GEN10

CONTENTS

EXECUTIVE SUMMARY

Oracle® Database 12c is critical to the operation of many businesses. Unexpected downtime, poor performance, or data loss can lead to lost revenue, diminished user productivity and customer dissatisfaction. HPE SimpliVity hyperconverged infrastructure provides a highly resilient and scalable operating environment for Oracle Database 12c across a wide variety of use cases.

HPE SimpliVity 380 Gen10 is an enterprise-grade hyperconverged platform that is designed and optimized for the virtual machine (VM) environment, speeds up application performance, improves efficiency and resiliency, and restores VMs in seconds.

The HPE SimpliVity System is a 2U rack-mounted building block that delivers server, storage, and networking services all-in-one. All of this comes at a fraction of the cost, and an extreme reduction in complexity, compared to a traditional infrastructure stack. It is the first hyperconverged platform to include native data protection, data efficiency, performance acceleration, and global unified management, all of which are delivered with HPE SimpliVity core enabling technology, the Data Virtualization Platform (DVP).

This paper documents the following solutions in an HPE SimpliVity Federation configuration. A federation is a collection of one or more HPE SimpliVity Clusters and the main construct within which data is managed.

HPE SimpliVity Cluster is a collection of one or more HPE SimpliVity hyperconverged nodes, typically located at the same physical site connected over a standard Ethernet network collectively providing a single storage pool to the hypervisor on each node.

- Oracle Database 12c - Disaster recovery solution using Oracle Data Guard and HPE Serviceguard for Linux® across production and recovery data centers
- Oracle Database 12c - High availability solution using Oracle Real Application Clusters (RAC)
- Oracle Database 12c - Application-consistent Oracle Database VM backup and restore using HPE SimpliVity built-in data protection
- Oracle Database 12c - Automation of Oracle Database VM backup using HPE SimpliVity REST API

The testing validated the core benefits of the SimpliVity OmniStack solution, namely:

- Accelerated Data Efficiency: OmniStack performs inline data deduplication, compression and optimization on all data at inception across all phases of the data lifecycle, all handled with fine data granularity of just 4KB-8KB. On average, SimpliVity customers achieve 40:1 data efficiency while simultaneously increasing application performance.
- Built-In Data Protection: OmniStack includes native data protection functionality, enabling business continuity and disaster recovery for critical applications and data, while eliminating the need for special-purpose backup and recovery hardware or software. OmniStack inherent data efficiencies, minimize I/O and WAN traffic, reducing backup and restore times from hours to minutes.
- Global Unified Management: OmniStack VM-centric approach to management eliminates manually intensive, error-prone administrative tasks. System administrators are no longer required to manage LUNs and volumes; instead, they can manage all resources and workloads centrally, using familiar interfaces such as VMware vCenter®.

Target audience: The document is aimed for Database Administrators (DBAs), who are responsible for managing database availability and performance, and their infrastructure counterparts who are responsible for the successful ongoing operation of the infrastructure. This includes individuals who work with VMware®, data center operations, and data protection.

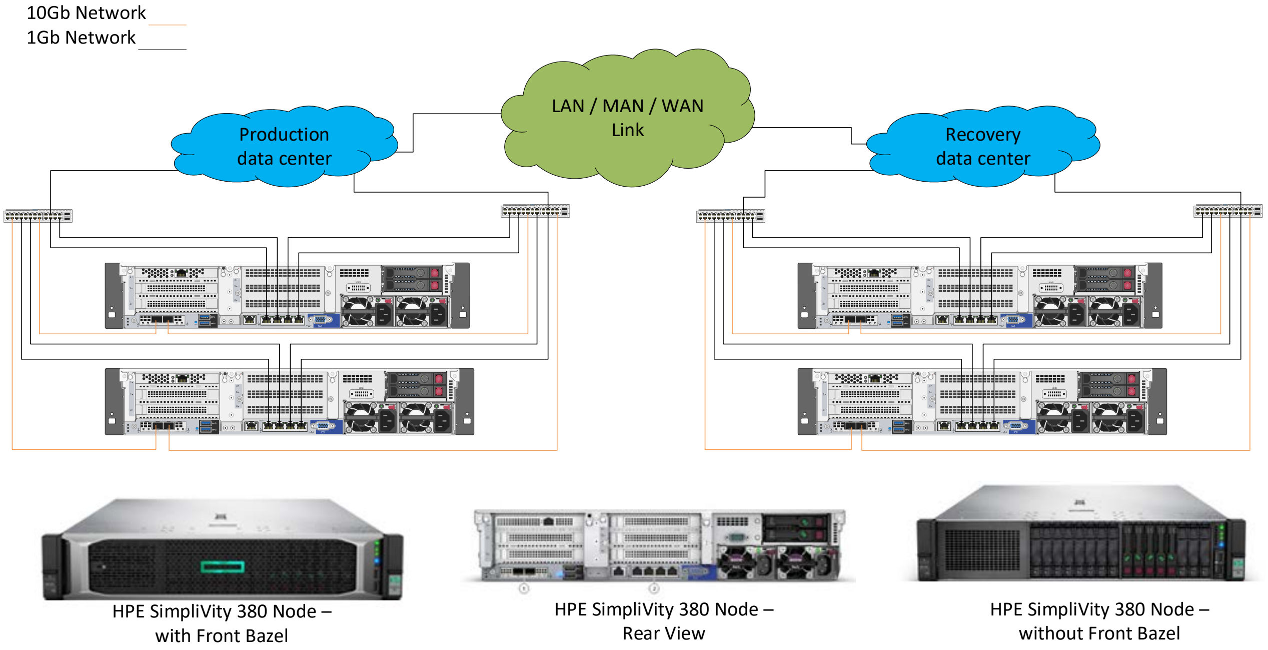
Document purpose: The purpose of this document is to describe a Reference Architecture, highlighting recognizable benefits to technical audiences.

This Reference Architecture describes solution testing performed in March 2019.

SOLUTION OVERVIEW

This solution leverages HPE SimpliVity 380 Gen10 Nodes, VMware vSphere® version 6.5, Red Hat® Enterprise Linux (RHEL) version 7.5, and Oracle Database 12c Enterprise edition. The HPE SimpliVity hyperconverged servers provide tremendous value with resiliency and availability built into the core architecture. If a host's hardware were to experience a problem, the VMs residing on the host would be restarted on a remaining host in the federation by VMware HA.

Four HPE SimpliVity 380 Gen10 Nodes were configured with both 10GbE and 1GbE interfaces. Redundant cablings (Figure 1) were used to protect against interface failures. In a production environment, the best practice is to deploy multiple network switches in a redundant configuration to protect against the loss of a single network switch.



HPE SimpliVity 380 Node – with Front Bazel

HPE SimpliVity 380 Node – Rear View

HPE SimpliVity 380 Node – without Front Bazel

FIGURE 1. HPE SimpliVity Federation – 2 plus 2 node network connectivity diagram

A common deployment for HPE SimpliVity 380 Gen10 is a two-site environment, in which HPE SimpliVity 380 Gen10 systems are deployed in each of two data centers, and connected within the same federation. This configuration provides high availability and local backup recoverability at each site. It also delivers elegant disaster recovery and simplified management as the SimpliVity systems and all associated VMs are managed from a single user interface pane within the VMware vCenter Server management Web Client.

SOLUTION COMPONENTS

The section briefly describes the key components of the solution.

- HPE SimpliVity 380 Gen10 Node
- Oracle Database 12c
- HPE Serviceguard for Linux
- Oracle Real Application Cluster
- Oracle Data Guard

HPE SimpliVity 380 Gen10 Node

HPE SimpliVity 380 Gen10, based on the HPE ProLiant DL380 Gen10 Servers is a compact, scalable 2U rack-mounted building block that delivers server, storage, and networking services. Adaptable for diverse virtualized workloads, the secure 2U HPE ProLiant DL380 Gen10 delivers world-class performance with the right balance of expandability and scalability. It also provides a complete set of advanced functionality that enables dramatic improvements to the efficiency, management, protection, and performance - at a fraction of the cost and complexity of today's traditional infrastructure stack.

FIGURE 2. HPE SimpliVity 380 Gen 10 Node

The HPE SimpliVity 380 Gen10 supports Intel® Xeon® scalable processors with 8 to 22 cores selectable with 1 or 2 CPU options, memory ranging from 144GB up to 1.5TB per node, 1Gb 4 Port Lights-out management (LOM) embedded, 10Gb or 25Gb 2 Port FlexibleLOM and the following storage options. Every HPE SimpliVity 380 Gen10 Node comes with the HPE OmniStack Accelerator Card (OAC) for data deduplication and compression.

HPE SimpliVity comes with two all-flash storage options, 4000 and 6000 Series with different capacity points. Refer to the HPE SimpliVity 380 Gen10 QuickSpecs document for more details and the latest information.

NOTE
Dual socket configurations may be required to achieve some capacity points.

The 4000 series all-flash is ideal for customers who have read-intensive or typical read/write mixed workloads. The 6000 series all-flash is ideal for customers who have high performance mixed workloads with read and/or write intensive requirements. The 6000 series is recommended for Oracle Database solution.

The XS and S solutions are very suitable for small and medium-sized business (SMB) and remote office/branch office (ROBO) customers who have lower compute needs and are looking for an affordable hyperconverged solution. The XL configuration is ideal for customers with high storage capacity workloads or who need a backup hub for distributed environments. Additional PCI Adapters can be added using a secondary riser and it can support up to 3 PCI adapters such as 1Gb, 10Gb, 25Gb Ethernet adapters and 16Gb, 32Gb Fibre Channel host bus adapter (HBA) cards for external storage connectivity.

Oracle Real Application Clusters

Oracle Database with the Oracle Real Application Clusters (RAC) option allows multiple database instances running on different servers to access the same physical database stored on shared storage. The database spans multiple systems, but appears as a single unified database to the application. This provides a scalable computing environment, where capacity can be increased by adding more nodes to the cluster. While all servers in the cluster must run the same OS and the same version of Oracle, they need not have the same capacity, which allows adding servers with more processing power and memory when more performance is required. This architecture also provides high availability, as RAC instances running on multiple nodes provide protection from a node failure.

HPE Serviceguard for Linux

HPE Serviceguard for Linux (SGLX) is a software based high availability and disaster recovery solution that increases the availability and uptime of your business critical applications and minimizes the impact of unplanned outages. SGLX packages applications and other services with their associated resources and monitors the entire package for any failure. Each package is monitored for faults related to hardware, software, OS, virtualization layer, virtual machine guests, network, and storage. When any failure is detected, SGLX shuts down the application quickly and smartly, relocate the application or service to another system with the necessary resources, to bring it into production again.

SGLX uses Quorum Server arbitration mechanism to prevent data corruption and loss in case of a split-brain situation among cluster nodes. Quorum server also supports a feature called Smart Quorum, which helps in increasing the availability of critical workloads in case of split-brain situation. The solutions in this Reference Architecture use virtual machines (VMs) as cluster nodes and a Quorum server running outside the SGLX cluster and Smart Quorum is by default enabled. SGLX also minimizes planned downtime using its Live Application Detach (LAD) and rolling upgrade feature to perform maintenance on clusters and install upgrades for any OS and application without downtime. Take advantage of the Cluster Verification technology to find and fix cluster configuration issues before they advance and cause unplanned downtime.

This Reference Architecture has captured the test results of HPE Serviceguard integration with Oracle Data Guard on Linux using a 2-node cluster configuration spread across two sites. The Serviceguard manages and completely automates the role switchover process of Oracle Data Guard databases between primary and physical standby node. SGLX is utilized to start, stop, and monitor the databases and administer the replication between primary and standby databases. SGLX also performs automatic role management to recover from failures. In case of failures the solution automatically recovers the Oracle database by promotion of the standby database instance to primary. This mode of recovery is much faster compared to a restart DB based solution. The databases can be located on the same premises or in geographically dispersed data centers. SGLX performs recovery point objective (RPO) sensitive automatic role management to recover from failures.

Oracle Data Guard

Oracle- Data Guard is part of Oracle Database Enterprise Edition software that provides a comprehensive set of services that create, maintain and monitor one or more standby databases on different server hardware located either at local or remote locations (for the purpose of disaster recovery) connected via LAN (Local Area Network) or WAN (Wide Area Network). It enables a production-level Oracle database (primary database) to survive disaster and data corruptions. Oracle Data Guard supports three types of standby databases, namely physical standby database, logical standby database and snapshot standby database. For more details, refer to Oracle Data Guard Concepts and Administration documentation.

- Physical standby database: Provides a physically identical copy of the primary database, with on-disk database structures that are identical to the primary database on a block-for-block basis. The database schema, including indexes, are the same. A physical standby database is kept synchronized with the primary database, through Redo Apply, which recovers the redo data received from the primary database and applies the redo to the physical standby database. This Reference Architecture document has validated a physical standby solution using SGLX.
- Logical standby database: Contains the same logical information as the production database, although the physical organization and structure of the data can be different. The logical standby database is kept synchronized with the primary database through SQL Apply, which transforms the data in the redo received from the primary database into SQL statements and then executes the SQL statements on the standby database.
- Snapshot standby database: Like a physical or logical standby database, a snapshot standby database receives and archives redo data from a primary database. Unlike a physical or logical standby database, a snapshot standby database does not apply the redo data that it receives until the snapshot standby is converted back into a physical standby database, after first discarding any local updates made to the snapshot standby database. A snapshot standby database is best used in scenarios that require a temporary, updatable snapshot of a physical standby database.

Oracle Active Data Guard is an additional licensed feature that enables read-only access to a physical standby database for queries, sorting, reporting, and web-based access and so on while continuously applying changes received from the production database.

Oracle Data Guard provides three modes of data protection known as Maximum Availability, Maximum Performance and Maximum Protection.

- Maximum Availability: In this mode, transactions do not commit until all redo data needed to recover those transactions has either been received in memory or written to the standby redo log on at least one synchronized standby database. If the primary database cannot write its redo stream to at least one synchronized standby database, it operates as if it were in maximum performance mode to preserve primary database availability until it is again able to write its redo stream to a synchronized standby database.

- Maximum Performance: This is the default protection mode and provides the highest level of data protection that is possible without affecting the performance of a primary database. This is accomplished by allowing transactions to commit as soon as all redo data generated by those transactions has been written to the online log. Redo data is also written to one or more standby databases, but this is done asynchronously with respect to transaction commitment, so primary database performance is unaffected by delays in writing redo data to the standby database(s).
- Maximum Protection: This protection mode ensures that no data loss occurs if the primary database fails. To provide this level of protection, the redo data needed to recover a transaction must be written to both the online redo log and to the standby redo log on at least one synchronized standby database before the transaction commits. To ensure that data loss cannot occur, the primary database shuts down, rather than continuing to process transactions, if it cannot write its redo stream to at least one synchronized standby database.

The databases used in this Reference Architecture used a physical standby database and the protection mode was set to maximum availability. In this environment, integrating Oracle Data Guard with HPE Serviceguard for Linux using HPE Serviceguard Toolkit for Oracle Data Guard on Linux provides the following advantages:

- Provides high availability and disaster recovery with fully automatic and fast recovery for Oracle databases
- Monitoring for primary and secondary DB instances and the Oracle Data Guard processes
- Smart Quorum feature decides which site will survive in an event of a split between the sites based on the workload criticality
- Recovery Point Objective sensitive recovery
- Built-in monitoring capabilities to check system resources, such as network, volume groups, files systems, etc. and failover in the case of failure of any of these components
- Protect and recover the application stacks in addition to databases
- Failure and status notifications
- Advanced features to minimize downtime for planned maintenance

Hardware

TABLE 1. Solution components - Hardware

HW	Sizing
HPE SimpliVity 380 Gen10 Node	4 Nodes
Memory	768 GB per node
CPU	Intel Xeon Gold 6142 CPU @ 2.60 GHz 2 CPU sockets with 16 cores per socket 64 logical processors with Hyper Threading enabled
Networking	Storage and federation: 10GbE switches Management: 1GbE switches
HPE OmniStack	3.7.6.157

Software

TABLE 2. Solution components - Software

SW	Release
Serviceguard for Linux	12.30
VMware vSphere	6.5 U2
Red Hat Enterprise Linux	7.5
HPE SimpliVity Plug-in for vSphere Web Client	15.39.20

Application software

TABLE 3. Solution components – application

SW	Release
Oracle Database Software and Grid Infrastructure	Oracle Database 12c Enterprise Edition 12.2.0.1
Oracle Data Guard	12.2.0.1, included in Oracle Database Software
Oracle Real Application Cluster	12.2.0.1, included in Oracle Database Software

BEST PRACTICES AND CONFIGURATION GUIDANCE FOR THE SOLUTION

This section describes various best practices and configuration guidance for Oracle database on HPE SimpliVity solutions.

Network configuration

This section describes the network configuration that was deployed in this Reference Architecture.

HPE SimpliVity deployment networks

The HPE SimpliVity deployment network usually contains 4 networks known as, Management network, VM network, federation network and Storage network. Federation and Storage networks always use 10Gbe interface and Management & VM networks may either use 10Gbe network or 1Gbe network based on application workload needs.

NOTE

HPE SimpliVity 380 Gen10 node supports 10/25GbE – 2 Port FlexLOM as well. If a production workload needs higher bandwidth, 10/25GbE could be ordered instead of 10GbE – 2 Port FlexLOM.

Figure 3 depicts the configuration used in this Reference Architecture, where the Oracle database client side network is connected to 10Gbe network port group called VM Network – 2 and the SGLX heartbeat and Oracle RAC private network is configured on a separate port group with 1Gbe NIC teaming interface, called as PvtNet. Oracle recommends a dedicated private network for Oracle RAC setup and preferably a 10Gb network, if heavy workloads are going to run on the Oracle RAC database.

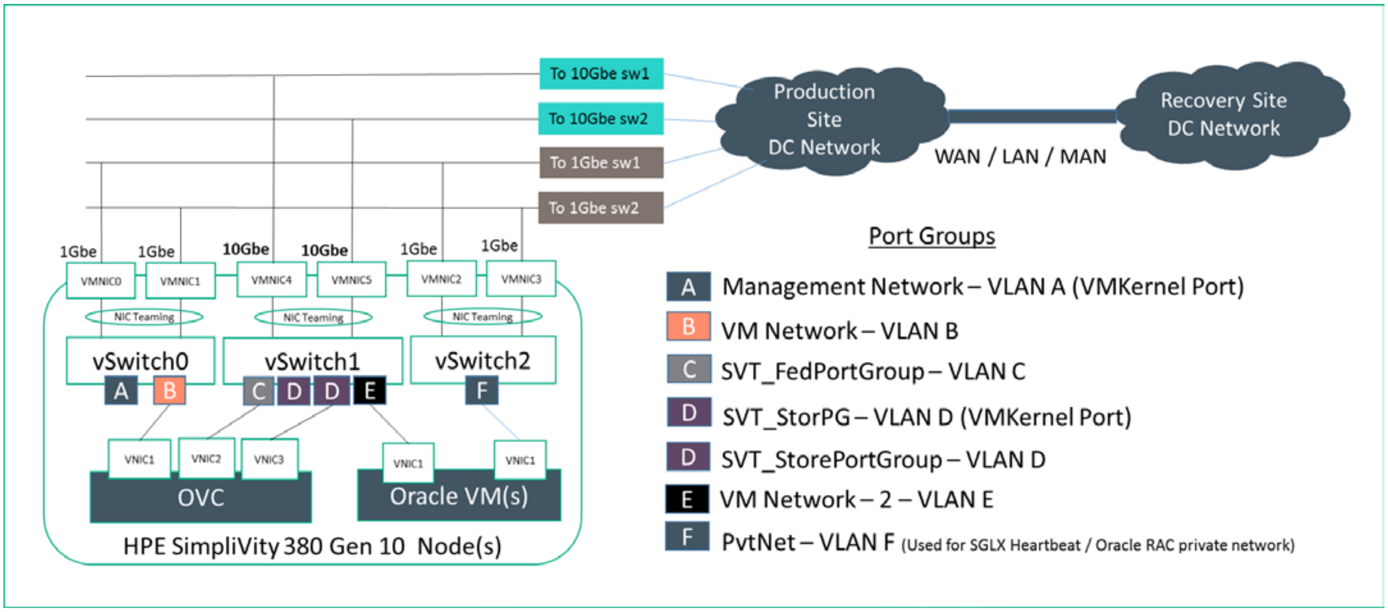

To 10Gbe sw1
To 10Gbe sw2
To 1Gbe sw1
To 1Gbe sw2
Production Site DC Network
WAN / LAN / MAN
Recovery Site DC Network
1Gbe
1Gbe
10Gbe
10Gbe
1Gbe
1Gbe
VMNIC0
VMNIC1
VMNIC4
VMNIC5
VMNIC2
VMNIC3
NIC Teaming
NIC Teaming
NIC Teaming
vSwitch0
vSwitch1
vSwitch2
A
B
C
D
D
E
F
VNIC1
VNIC2
VNIC3
VNIC1
VNIC1
OVC
Oracle VM(s)
HPE SimpliVity 380 Gen 10 Node(s)
Port Groups
A Management Network – VLAN A (VMKernel Port)
B VM Network – VLAN B
C SVT_FedPortGroup – VLAN C
D SVT_StorPG – VLAN D (VMKernel Port)
D SVT_StorePortGroup – VLAN D
E VM Network – 2 – VLAN E
F PvtNet – VLAN F (Used for SGLX Heartbeat / Oracle RAC private network)

FIGURE 3. HPE SimpliVity 380 Gen10 Node – VMware ESXi network configuration

The Storage network handles below traffic:

- VMware datastore access to VMs.

The federation network handles below traffic:

- Communication between HPE SimpliVity nodes within an HPE SimpliVity Cluster.
- VM replication between HPE SimpliVity nodes within an HPE SimpliVity Cluster.
- If network route is available, VM level backup traffic goes between HPE SimpliVity Clusters via the federation network within an HPE SimpliVity Federation.

NOTE
Backup traffic by default is transported over the federation network if a network route is available between the federation networks of HPE SimpliVity hyperconverged nodes. If there is no route available, which is generally the case with remote data centers, the backups are transported over the management network. Therefore, based on the backup schedules and rate of change of data on the VMs being protected, sufficient bandwidth should be available on the management network.

HPE SimpliVity node - NIC Teaming

NIC teaming helps to increase network capacity for the virtual switch hosting the team and provides passive failover if the hardware fails or it loses power. To use NIC teaming, you must uplink two or more adapters to a virtual switch and the following settings need to be set for every NIC team in the setup at vCenter > Simplivity Federation > Click Hosts to open Objects tab > Right click on host and click Settings > Click on Networking subtab > Click the name of the switch > Click the pencil icon to open the Edit Settings dialog box:

- Load balancing: Route based on the original virtual port
- Network failover detection: Link status only
- Notify switches: Yes
- Failback: Yes

In this configuration, a VM will have its network running on a designated physical NIC only and can failover to another NIC within the NIC teaming, in case of physical NIC or switch failure, and it does not require any special physical switch side configuration. Network bandwidth for a VM network is limited to a single physical NIC within the NIC team.

HPE SimpliVity – Arbiter service network

The Arbiter is a service that runs on a Windows system and acts as a witness to maintain quorum for an HPE SimpliVity Cluster to ensure data availability and data consistency, should an HPE SimpliVity node fail or become inaccessible. HPE SimpliVity servers communicate with the Arbiter over the management network using both UDP and TCP on port 22122. Round trip latency between the Arbiter and the SimpliVity nodes should be no more than 300 ms.

A single Arbiter can witness all HPE SimpliVity Clusters in an HPE SimpliVity Federation. The Arbiter is a dependency for HPE SimpliVity Clusters but if an Arbiter fails all SimpliVity features continue to function and workloads remain available. All nodes within the same HPE SimpliVity Cluster must communicate with the same Arbiter.

The Arbiter service can be installed on a physical server or virtual machine and there is no need to install it on a dedicated server. The Arbiter service can be safely installed on a server running other services such as vCenter, Active Directory Server, DNS, etc., but it cannot be run on an HPE SimpliVity datastore in an HPE SimpliVity Cluster which it is witnessing.

HPE Serviceguard – Data and Heartbeat network

To avoid single point of failure, HPE Serviceguard for Linux recommends you to deploy a highly available network configuration with redundant heartbeats and data networks. Use VMware NIC teaming at the host level for all networks used by the applications that run on VMs and do not use NIC teaming at the guest level. Heartbeat roundtrip network latency, measured using ping should be no more than 200 ms.

NOTE

While applying the cluster configuration, Serviceguard just verifies the network requirement at cluster node (VM) level only and does not verify it at hypervisor level. Therefore, you may see a warning message about hearbeat network not being redundant, while applying the cluster configuration with a single heartbeat network connected to a vSwitch with NIC teaming configured. You can safely ignore the message and continue the cluster configuration.

You may also configure every subnet including the data network as Serviceguard heartbeat network. This would avoid above error message and also improves the heartbeat network redundancy.

HPE Serviceguard Quorum network

The Serviceguard Quorum Server provides arbitration services for Serviceguard clusters when a cluster partition is discovered. Ideally the Quorum Server and the clusters must communicate over a subnet that does not handle other traffic. This helps to ensure that the Quorum Server is available when it is needed. If this is not practical, or if the communication must cover a long distance (for example, if the Quorum Server is serving an Extended Distance cluster, such as Oracle Data Guard), heavy network traffic or network delays could cause Quorum Server timeouts. You can reduce the likelihood of timeouts by increasing the Quorum Server timeout interval; use the `QS_TIMEOUT_EXTENSION` parameter in the cluster configuration file.

If a subnet that connects the Quorum Server to a cluster is also used for the cluster heartbeat, configure the heartbeat on at least one other network, so that both Quorum Server and heartbeat communication are not likely to fail at the same time.

Oracle Data Guard redo transport network

Primary and standby database connect over a TCP/IP network using Oracle Net Services, and there are no restrictions on where the databases are physically located as long as they can communicate with each other.

Based on Oracle documentation Best Practices for Synchronous Redo Transport, Oracle has seen customers having greater success with synchronous transport when round trip time (RTT) network latency is less than 5 ms with OLTP workload that generated redo at 30MB/s. Oracle database performance impact may increase proportionally as the RTT and the rate of redo log generation increases. These are samples from testing and it is highly recommended to do the testing with real production data before drawing any specific conclusions on the impact of synchronous replication.

Refer to Oracle documentation How To Calculate The Required Network Bandwidth Transfer Of Redo In Data Guard Environments (Doc ID 736755.1) for more details on determining the required bandwidth for your environment.

The following are the best practices recommendations based on the Oracle documentation Best Practices for Synchronous Redo Transport.

- Primary and standby database systems must have the same set of configuration with respect to OS, kernel parameter settings, database parameters, vCPU, memory, disk layout and network connectivity. This is to ensure that in case of the primary database failure, application and clients can connect to the standby database without any performance impact.
- Configure Flashback database on both primary and standby databases. This enables rapid role transitions and reduces the time taken for re-establishing roles after switchover or failover.
- In this Reference Architecture, redo logs are sent over the VM Network from the primary to the standby database. Some high volume applications may require a dedicated network for redo log transfer. HPE SimpliVity 380 Gen10 node with 2-processor supports up to three additional PCI adapters using a secondary riser card. Using this additional option, a dedicated vSwitch may be created for this purpose.
- Set the TCP send and receive socket buffer size to 3 times of the Bandwidth Delay Product (BDP). BDP is product of the network bandwidth and latency. The size of the socket buffers is recommended to be set at the Oracle Net level, rather than the operating system (OS) level.

- Online redo logs and standby redo logs should use redo log size = 4GB, or more than or equal to peak redo rate/minute x 20 minutes. To extract peak redo rates, refer to Automated Workload Repository (AWR) reports during peak workload periods such as batch processing, quarter or year-end processing.
- Online redo logs and standby redo logs should be of the same size. The additional standby redo log eliminates the possibility of a standby database waiting on standby redo log. It is critical for performance that standby redo log groups only contain a single member.
- Oracle testing has shown that adjusting the size of the Oracle Net setting for the session data unit (SDU) to its maximum value of 65535 can improve performance of SYNC transport.
- Fast Sync (SYNC NOAFFIRM) enables all sessions waiting for a remote RFS (the Data Guard process that receives redo at the standby database) write to proceed as soon as the write is submitted, not when the write completes.

Oracle RAC network requirement

When installing Oracle RAC, at least two network interfaces and four IP addresses are required for each node in the RAC cluster.

- Public Interface: Used for normal network communications to the nodes and database clients, and it requires a dedicated network interface.
- Private Interface: This is used as the cluster interconnect. This IP address must be separate from the public network and it must be configured on a dedicated physical interface.
- Virtual Interface: Used for failover and RAC management. The virtual IP address should be just configured in /etc/hosts and the DNS server only, and not manually configured in the NIC interface. It will be configured internally on the Public interface by an Oracle process.
- SCAN (Oracle Single Client Access Name) Interface: This is an Oracle RAC feature that provides a single name for clients to access Oracle Databases running in a cluster. Usually 3 IPs are used and DNS provides name resolution to IPs in a round robin fashion.

Cluster configuration

This section briefly describes the cluster configurations which were deployed as part of the solution.

HPE SimpliVity cluster and VMware vSphere HA configuration

An HPE SimpliVity cluster can contain one to sixteen HPE SimpliVity nodes and unlimited standard ESXi hosts that may or may not share a SimpliVity datastore. You need at least two HPE SimpliVity nodes to create a SimpliVity Federation, share data, and ensure High Availability (HA) for continuous operation. A single node configuration is also supported and it is mostly deployed at disaster recovery site. Refer to HPE SimpliVity 380 Gen10 QuickSpecs document for the latest information.

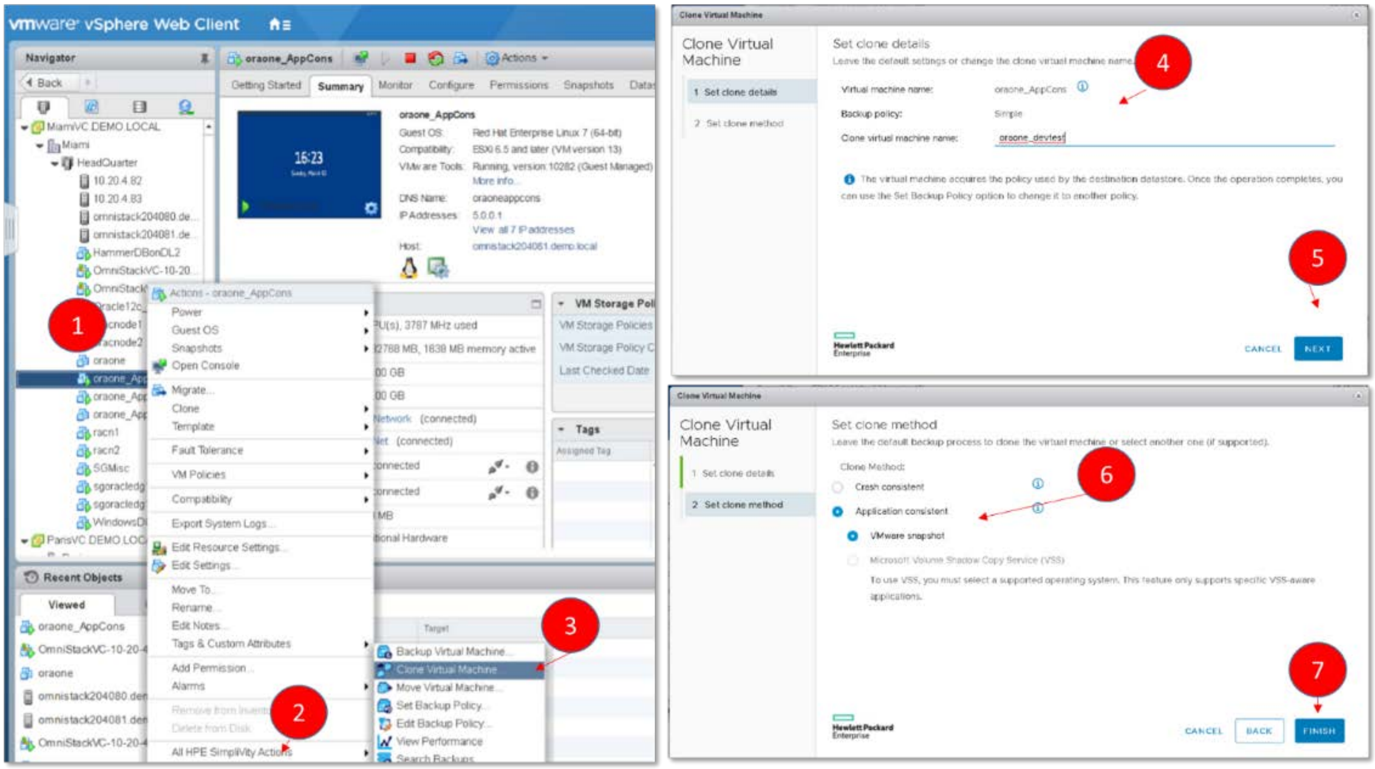
VMware vSphere HA should be enabled to protect against host and virtual machine failure.

If you use vSphere Distributed Resource Scheduler™ (DRS) with HPE SimpliVity node in a cluster, you should also use SimpliVity Intelligent Workload Optimizer (IWO). IWO works with vSphere DRS to move virtual machines closer to their data source (host with the source virtual machine). This improves load balancing efficiency, reduces the number of hops to read and write data, and lowers latency.

Leave vSphere Distributed Power Management™ (DPM) off. DPM is an optional feature of vSphere DRS that optimizes power. However, if you use it, it could shut down the Virtual Controllers to save power. An HPE SimpliVity node cannot function without the Virtual Controller.

USE CASE 1: DEPLOYING PRE-PRODUCTION VM RUNNING ORACLE DATABASE IN SECONDS FOR TEST AND DEVELOPMENT ENVIRONMENTS

Pre-production Oracle workloads benefit from OmniStack's inherent data efficiencies and Global Unified Management capabilities. With SimpliVity, the system administrator can clone VMs in just seconds, with a few mouse clicks, to easily spin up Oracle QA, test or dev environments. OmniStack performs inline deduplication, compression, and optimization at inception, before data hits the disk, eliminating redundancy and overhead, making optimal use of storage capacity. If application consistent cloning is desired, it is recommended to put the database into backup mode using pre-freeze and post-thaw scripts. Refer to Use Case 4 in this Reference Architecture document for more details on pre-freeze and post-thaw scripts and an example.



Clone Virtual Machine

1 Set clone details

2 Set clone method

Set clone details

Leave the default settings or change the clone virtual machine name.

Virtual machine name: oraone_AppCons

Backup policy: Simple

Clone virtual machine name: oraone_devtest

The virtual machine acquires the policy used by the destination datastore. Once the operation completes, you can use the Set Backup Policy option to change it to another policy.

Set clone method

Leave the default backup process to clone the virtual machine or select another one (if supported).

Clone Method:

○ Crash consistent

● Application consistent

● VMware snapshot

○ Microsoft Volume Shadow Copy Service (VSS)

To use VSS, you must select a supported operating system. This feature only supports specific VSS-aware applications.

FIGURE 4. HPE SimpliVity Cloning - Application consistent cloning of production database

Follow these steps to clone a VM using HPE SimpliVity Clone feature and it allows you to either create an application consistent clone or crash consistent clone.

1. Right click on VM to be cloned
2. Click on All HPE SimpliVity Actions
3. Click on Clone Virtual Machine
4. Enter name for Clone virtual machine
5. Click Next
6. Select Application consistent radio button and then select VMware snapshot
7. Click Finish.

USE CASE 2: ORACLE DATABASE 12C - DISASTER RECOVERY SOLUTION USING ORACLE DATA GUARD AND HPE SERVICEGUARD FOR LINUX

Oracle Data Guard (ODG) is the Oracle database disaster recovery solution to protect production databases from failures, disaster and human errors. Using HPE Serviceguard for Linux (SGLX) along with Oracle Data Guard, server administrators can manage and completely automate the failover process of Oracle database in ODG configuration. SGLX is responsible for starting, stopping, and monitoring the databases and administer the replication between primary and physical standby databases. SGLX also performs automatic role management to recover from failures. In case of failures, the solution automatically recovers the Oracle database by promotion of the standby database instance to primary. The databases can be located on the same premises or in geographically dispersed data centers. SGLX performs recovery point objective (RPO) sensitive automatic role management to recover from failure.

Figure 5 shows the SGLX with ODG configuration which was validated in this Reference Architecture.

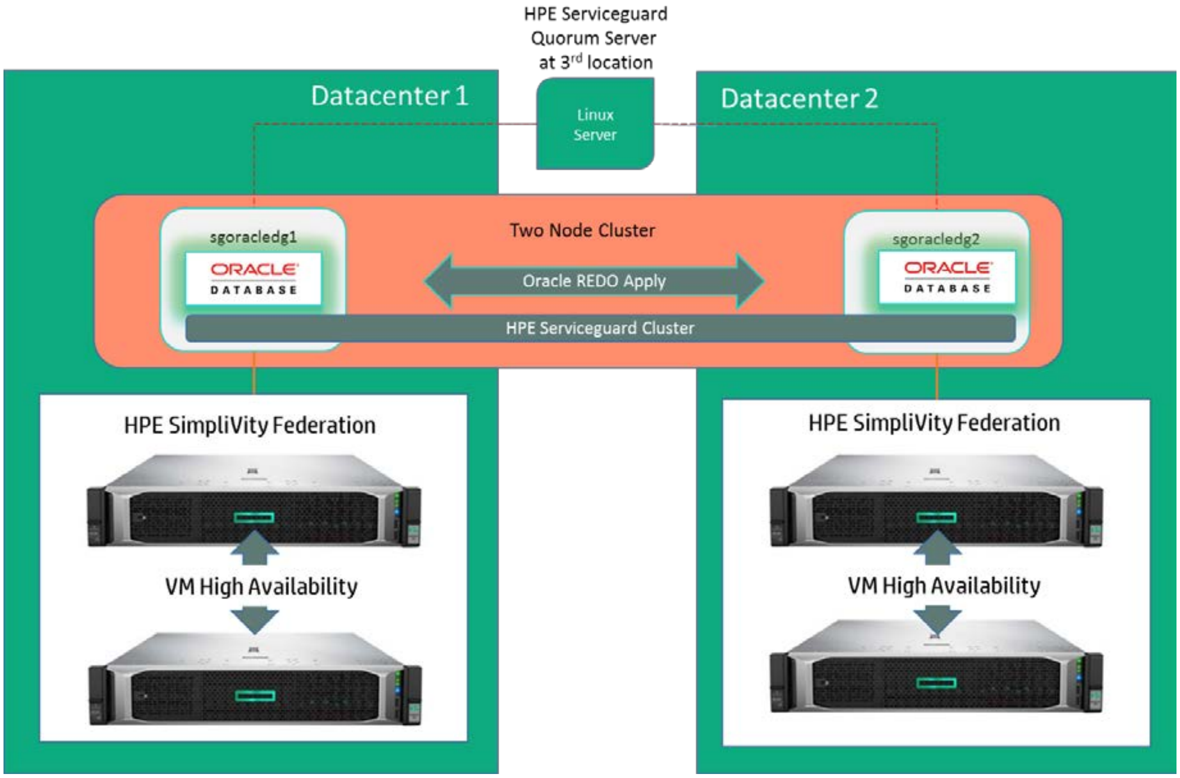


FIGURE 5. HPE Serviceguard and Oracle Data Guard based disaster recovery solution

Oracle database and ODG installation and configuration

1. Create a VM at production site (Datacenter 1 (DC 1)) and at recovery site (Datacenter 2 (DC 2))
2. Install RHEL 7.5 operating system and fulfill all the pre-requisites for Oracle database installation
3. On both the VMs, install Oracle Database 12c software
4. Create Oracle database instance on VM at production site using dbca
5. Configure Oracle Data Guard using Oracle documentation
6. Following table shows the database configuration used in this Reference Architecture

TABLE 4. Oracle database – Oracle Data Guard and SGLX cluster configuration

Item description	Primary database at Production Site	Physical Standby database at Recovery Site
Site Name	Miami POC (Datacenter 1)	Paris POC (Datacenter 2)
Hostname	sgoracledg1	sgoracledg2
Database Name (db_name)	orcl	orcl
Database Unique Name (db_unique_name)	orcl	orcls
Instance Name (SID)	orcl	orcl
Data Network (Database Client Network)	VM Network - 2 (Portgroup at vSwitch1)	VM Network - 2 (Portgroup at vSwitch1)
Serviceguard Heartbeat Network	PvtNet (Portgroup at vSwitch2)	PvtNet (Portgroup at vSwitch2)
DATA filesystem ($ORACLE_BASE/oradata)	LVM Lvol: 500GB Filesystem: ext4	LVM Lvol: 500GB Filesystem: ext4
Flash Recovery Area (FRA) filesystem ($ORACLE_BASE/flash_recovery_area)	LVM Lvol: 500GB Filesystem: ext4	LVM Lvol: 500GB Filesystem: ext4

NOTE
Both ODG and SGLX support ASM based storage for Oracle database. This Reference Architecture validated ODG database in a non ASM configuration. For better performance, ASM based storage for database is recommended.

1. Mount the SGLX installation ISO image and install SGLX software using cminstaller.
2. Install and configure HPE Serviceguard Quorum server at 3rd location. Quorum server software is located in the SGLX installation ISO image.
3. Deploy a 2 Node Serviceguard cluster using `cmdeploycl` command.
4. Ensure Oracle Data Guard Primary and Physical Standby databases are up and running
5. Deploy Serviceguard packages for ODG in the cluster using cmoradgworkload command or Serviceguard Manager GUI. This command will automatically detect ODG configuration and create three packages called as DG1_DC1_<SID>, DG1_DC2_<SID> and DG1_RM. For example, DG1_DC1_orcl, DG1_DC2_orcl and DG1_RM. DG1_DC1_<SID> runs only on node that is located on Datacenter 1 (DC1) and DG1_DC2_<SID> runs only on node that is located on Datacenter 2 (DC 2). DG1_RM runs on the node whichever is playing Primary database role.
6. Now, DBA can use SGLX commands to start and stop Oracle Data Guard databases.

Failover test

Prior to performing failover test, the Primary database was running on node at DC 1 and Physical Standby database on node at DC2. Primary database node was directly powered off using vCenter, simulating database server failure. SGLX detected the node failure, failed over the package DG1_RM to node at DC2, and reconfigured the Physical Standby database to Primary database within ~24 Seconds.

As shown in Figure 6, total recovery time was ~24 seconds. Note that this test was done with minimal load on the database and failover time may vary based on the amount of redo log apply happening on the physical standby database. By configuration wise, failover time could be further minimized by reducing the MEMBER_TIMEOUT parameter of SGLX cluster.

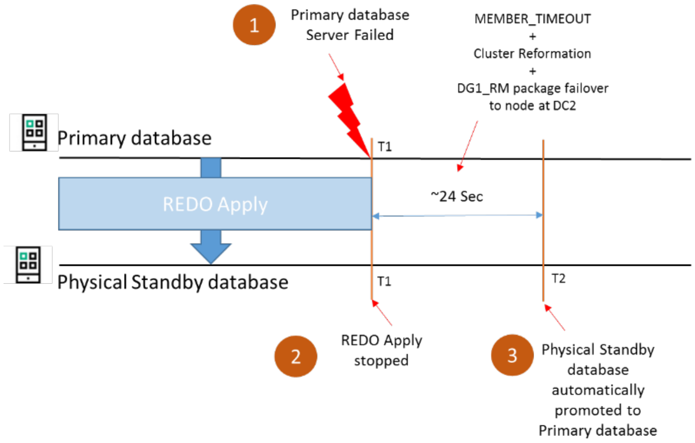

1
Primary database Server Failed
MEMBER_TIMEOUT + Cluster Reformation + DG1_RM package failover to node at DC2
Primary database
T1
REDO Apply
~24 Sec
Physical Standby database
T1
T2
2
REDO Apply stopped
3
Physical Standby database automatically promoted to Primary database

FIGURE 6. SGLX and ODG – Failover due to cluster node failure

The default value for this parameter is 14 seconds, but the minimum value for this parameter is 3 seconds for a cluster with more than one heartbeat subnet.

NOTE

Setting lower MEMBER_TIME values enables faster detection times, but it might make the cluster more sensitive to transient network issues, if any.

Serviceguard for ODG provides support for Recovery Point Objective (RPO) sensitive failover using RPO_LIMIT parameter. When the data replication fails (RFS process failure) for a duration longer than the configured RPO limit whose value is in seconds, the role management package will not failover to the node where the standby DB is running to avoid data loss. The RPO supports email notification, if an email ID is configured. You receive notification if replication is not happening for the duration configured as RPO_LIMIT, and also receive notification when the replication comes back online. By default, the RPO is set to zero (0), which means zero data loss. For more details on the configuration, refer to HPE Serviceguard Toolkit for Oracle Data Guard on Linux User Guide. Figure 7 depicts a failure scenario where Remote Failure Server (RFS) process failure at T1 time at Physical Standby and RPO_LIMIT set to 60 sec has been breached at T2 time. Between T2 and T3 time, Physical Standby will not be promoted to Primary, if Primary server fails. At T3 time, RFS process starts running and replication resumes.

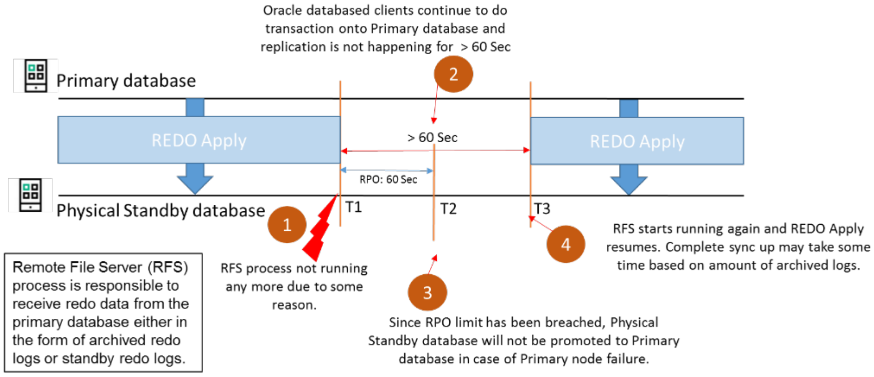

Oracle databased clients continue to do transaction onto Primary database and replication is not happening for > 60 Sec
2
Primary database
REDO Apply
> 60 Sec
REDO Apply
RPO: 60 Sec
Physical Standby database
1
T1
T2
T3
4
RFS starts running again and REDO Apply resumes. Complete sync up may take some time based on amount of archived logs.
Remote File Server (RFS) process is responsible to receive redo data from the primary database either in the form of archived redo logs or standby redo logs.
RFS process not running any more due to some reason.
3
Since RPO limit has been breached, Physical Standby database will not be promoted to Primary database in case of Primary node failure.

FIGURE 7. SGLX and ODG – Recovery Point Objective sensitive failover

USE CASE 3: ORACLE DATABASE 12C HIGH AVAILABILITY SOLUTION USING ORACLE REAL APPLICATION CLUSTER (RAC)

When RPO and RTO SLAs for unplanned outages are above five minutes, and routine maintenance windows for the Oracle Database and operating system are in place, consider leveraging VM-level high availability through VMware's HA feature. If a host fails, the VMs that were running on that host are then restarted on the remaining hosts, assuming VM restart takes less than five minutes. But, when RPO and RTO SLAs for unplanned outages are below five-minute threshold, or no maintenance window for planned outages exist, leveraging Oracle Real Application Cluster (RAC) functionality on top of SimpliVity will help your organization reduce the downtime windows. RAC also helps to scale out performance by adding more nodes to the cluster.

In a RAC environment, two or more database instances on different servers leverage shared storage to access a single database. A private network facilitates the communication between the cluster nodes. Rolling upgrades can be implemented for necessary maintenance operations, which helps to reduce the application downtime. Refer to Oracle Real Application Clusters Administration and Deployment Guide for more details on deployment.

Shared storage for Oracle RAC must meet high availability requirements and files stored on the disk must be protected by data redundancy. HPE SimpliVity Data Virtualization Platform (DVP) is built on top of RAID volumes configured on SSDs and disk failures are taken care of by RAID. In addition to this, DVP replicates each VM folder, called replica set containing total two copies (primary and standby) of each VM at two different SimpliVity nodes. Upon SimpliVity node failure, standby copy will become primary copy transparently. In this Reference Architecture, two different shared disk configuration, listed below have been tested.

- In a two node cluster configuration, one node shares its data disks to the other node. In this configuration, all the data disks are located under one VM folder. When the node which is sharing its disk fails, DVP makes the failed VM's standby copy as primary in the replica set and the surviving node can still access the shared disks from the new primary copy.
- In a two node cluster, both nodes share their data disks to each other and Oracle Automatic Storage Management (ASM) mirroring is used to mirror data between these disks (known as Failgroups), thus maintaining a redundant copy of the database in each VM folder, rather than on a single VM folder. ASM mirroring provides additional redundancy in addition to HPE SimpliVity replica set. But, it will have a performance impact due to dual writes for every data block.

Failure scenarios

In both the scenarios discussed above, it was observed that when one of the RAC node or one of the SimpliVity node fails, the other RAC node continue to access the disks, which belong to the failed node via DVP. Note that upon evicting a node from the cluster, Oracle RAC instance recovery happens and it would interrupt service in all Oracle RAC node, because Oracle RAC must remaster internal services and restore the database by using the current state of the redo log file. The Oracle service interruption noticed was around ~30 seconds and ongoing transaction resumed after 30 seconds. The amount of disruption can vary based on workloads involved. Figure 8 depicts both the configuration.

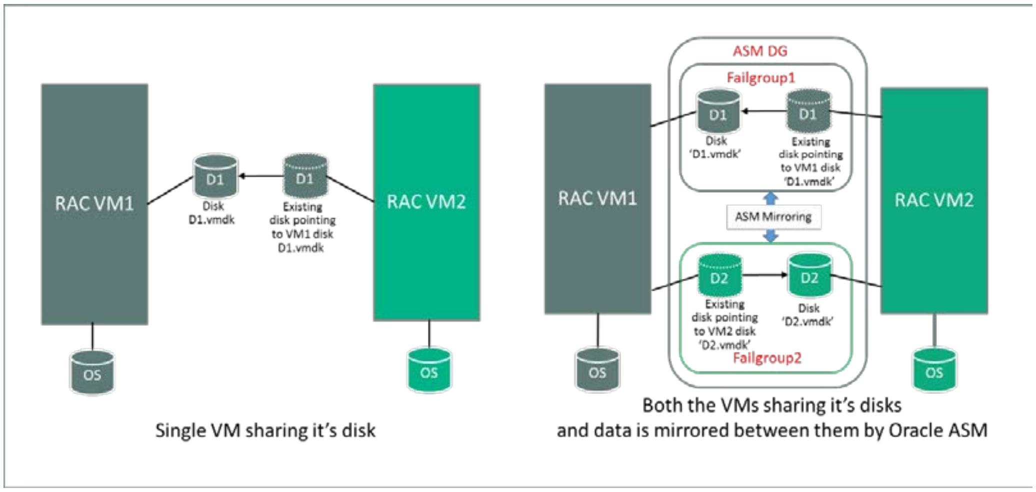


FIGURE 8. Oracle RAC – Two node cluster using single node sharing its disks and two node cluster using Oracle ASM Failgroup based mirroring

Oracle Database 12c High Availability Solution – Two Node Oracle RAC Cluster using shared disks from one of the RAC cluster nodes

This section describes the implementation of two node RAC setup where a RAC node is sharing its data disks to the other RAC node. Figure 9 depicts a two node SimpliVity Federation configuration.

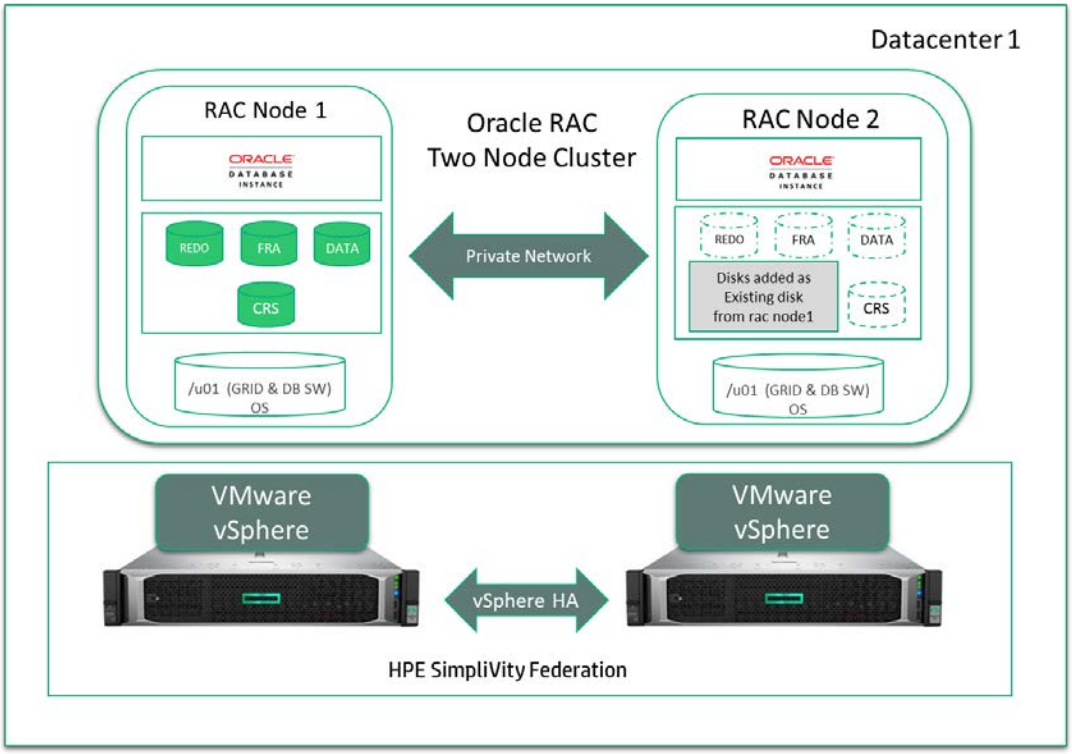


FIGURE 9. Oracle RAC – Two Node Oracle RAC Cluster using shared disk from one of the RAC cluster node

Follow these steps to create RAC cluster.

1. Create a VM on each SimpliVity node at the production site (Datacenter 1).
2. On the VMs, add three new SCSI controllers (SCSI controller 1, SCSI controller 2, SCSI controller 3) and configure all of them as VMware Paravirtual. SCSI controller 0 is used for boot volume only.
3. Oracle RAC requires a dedicated network adapter for intra-cluster communication. Therefore, configure at least two network interfaces for public and private networks and configure them as type VMXNET3. The private network should be non-routable and dedicated for RAC communication. These port groups could even be bound to dedicated physical network adapters, if required to maintain absolute peak performance.
4. Configure storage as given in the table below and set the following parameters for every disk shown in the table.

   Disk Mode: Independent-Persistent

   Sharing: No sharing (see note below)

NOTE
Oracle RAC requires shared storage among RAC nodes. For sharing a disk across RAC nodes, the disk's "Sharing" parameter has to be set to "Multi-writer". However, vCenter GUI throws an error "Incompatible device backing specified for device'0'" when Sharing value is set to Multi-writer for the thin provisioned volumes, and SimpliVity supports only thin provisioned volumes. There is a known issue and workaround documented in the VMware article https://kb.vmware.com/s/article/2147691 for NFS based datastore. However, in a SimpliVity environment, the workaround described in this article does not work. Therefore, as a workaround for SimpliVity, the Multi-writer value must be manually set later in the VM's .vmx file after configuring VMs using vCenter.

TABLE 5. ASM disk group configuration for 2 node clusters

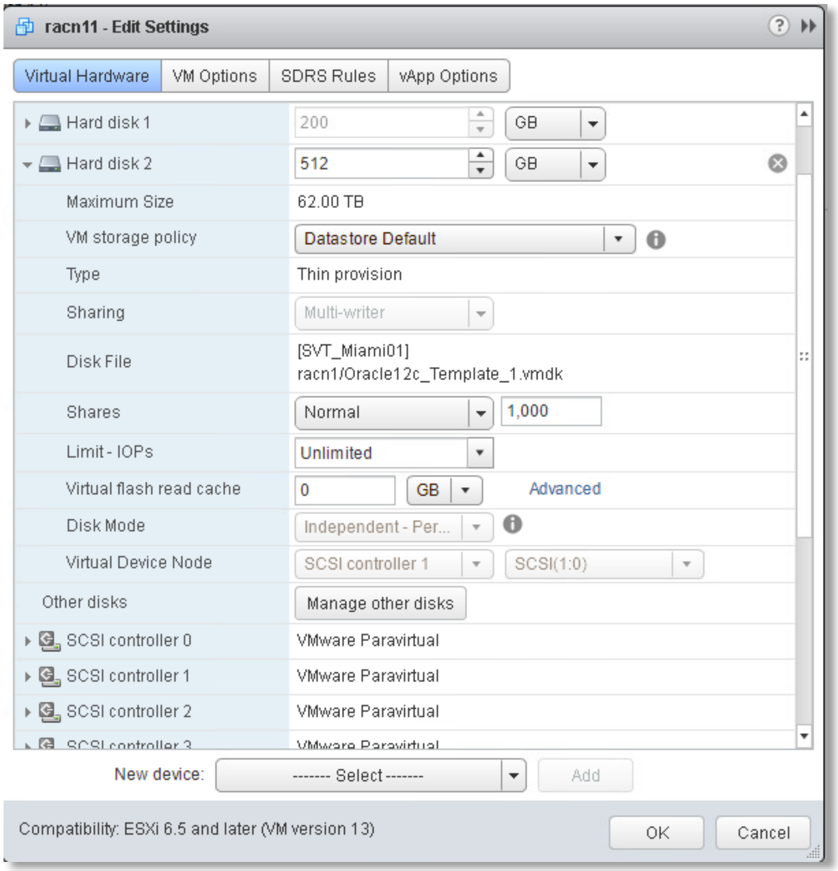
ASM DG	SCSI Ctl / ASM Disk / Size	Details
DATA	SCSI(1.0) / DATAN1D1 / 500GB	racnode1: Add this disk as local disk racnode2: Add this disk as an Existing disk
REDOa	SCSI(2:1) / REDOAN1D1 / 200GB	racnode1: Add this disk as local disk racnode2: Add this disk as an Existing disk
REDOa	SCSI(2:2) / REDOAN1D2 / 200GB	racnode1: Add this disk as local disk racnode2: Add this disk as an Existing disk
REDOb	SCSI(3:1) / REDOBN1D1 / 200GB	racnode1: Add this disk as local disk racnode2: Add this disk as an Existing disk
REDOb	SCSI(3:2) / REDOBN1D2 / 200GB	racnode1: Add this disk as local disk racnode2: Add this disk as an Existing disk
OCRVD	SCSI(3:3) / OCRVDN1D1 / 300GB	racnode1: Add this disk as local disk racnode2: Add this disk as an Existing disk

5. Logon to ESXi host @ SimpliVity node1 as root and go to racnode1 directory where racnode1 VMDK files are located. Open .vmx file and add scsi#:#.sharing as "multi-writer" for every disk except the OS volume. Repeat this task on all the RAC nodes.

   For example:

   ```
   scsi1:0.deviceType = “scsi-hardDisk”
   scsi1:0.fileName = “/vmfs/volumes/3dc72ade-5efd29da/racnode1/racnode_1.vmdk”
   scsi1.0.mode = “independent-persistent”
   scsi1:0.sharing = “multi-writer”
   sched.scsi1:0.shares = “normal”
   sched.scsi1:0.throughputCap = “off”
   scsi1:0.present = “TRUE”
   ```

Figure 10 shows RAC node disk configuration after above changes are made in ESXi host.



Hard disk 1	200	GB
Hard disk 2	512	GB
Maximum Size	62.00 TB	
VM storage policy	Datastore Default	
Type	Thin provision	
Sharing	Multi-writer	
Disk File	[SVT_Miami01] racn1/Oracle12c_Template_1.vmdk	
Shares	Normal	1,000
Limit - IOPs	Unlimited	
Virtual flash read cache	0 GB	Advanced
Disk Mode	Independent - Per...	
Virtual Device Node	SCSI controller 1	SCSI(1:0)
Other disks	Manage other disks	
SCSI controller 0	VMware Paravirtual	
SCSI controller 1	VMware Paravirtual	
SCSI controller 2	VMware Paravirtual	
SCSI controller 3	VMware Paravirtual	

Compatibility: ESXi 6.5 and later (VM version 13)

FIGURE 10. RAC Node disk configuration

6. Power on the VM and install RHEL 7.5 operating system.
7. Verify all the shared disks are visible on all the RAC nodes.
8. Complete all the standard prerequisites for bringing up the ASM instance, GRID and Oracle Database, i.e., /etc/hosts entries, swap size, required RPM packages for ASM, GRID and the Oracle database, the Oracle environment settings, oracle user account, permissions etc. All nodes in the Oracle RAC cluster must be able to communicate with each other and with external clients using TCP/IP. The following are interfaces and IP address assignments for the RAC nodes.

TABLE 6. Oracle database – RAC Cluster configuration

Name	2 Node Cluster	Description
Domain Name	orainfra.local	Register all the host names in DNS.
Public IPs	racnode1 10.0.0.1 racnode2 10.0.0.2	Communication between clients and the nodes in the cluster is across the public network. A dedicated VNIC adapter configured for the public network in each node.
Private IPs	racnode1-prv 8.0.0.1 racnode2-prv 8.0.0.2	For communication between instances running on all nodes, a private network is required. This private network connects only the nodes in the cluster and cannot be accessed from outside the cluster. All nodes need a separate network adapter configured for this private network.

Name	2 Node Cluster	Description
VIP	racnode1-vip 10.0.0.11 racnode2-vip 10.0.0.12	To enable high availability and failover, a virtual IP (VIP) address is also required for each node. A VIP address is moved between nodes in case of a failure by an Oracle RAC process called Cluster Ready Services (CRS). To support a virtual IP address, all nodes require an unused IP address that is compatible with the public network's subnet and subnet mask. This IP will be configured by Oracle process and no manual configuration required.
SCAN (Single Client Access Name) IP	orcl-scan.orainfra.local 10.0.0.21 10.0.0.22 10.0.0.23	SCAN is a feature used in Oracle RAC environments that provides a single name for clients to access any Oracle Database running in a cluster. The IP addresses must be on the same subnet as your default public network in the cluster. A DNS server must be set up to provide round-robin access to the IPs resolved by the SCAN entry. The Oracle Client typically handles failover of connection requests across SCAN listeners in the cluster.
Data Network (Database Client Network)	VM Network - 2 (Portgroup at vSwitch1)	10Gbe network. For heavy workload environment, use a dedicated 10Gbe network rather than sharing the storage federation network.
Oracle RAC Private Network	PvtNet (Portgroup at vSwitch2)	1Gbe network. For heavy workload environment, use dedicated 10Gbe network.

9. Create a partition on all the shared disks.

10. Bring up ASM instance and create asm disks at racnode1.

    a. Configure ASM

    ```
    # oracleasm configure -i
    # /usr/sbin/oracleasm init
    ```

    b. Create ASM disks

    ```
    # oracleasm createdisk <ASM_DISK_NAME> <Dev>
    ```

    Example:

    ```
    # oracleasm createdisk DATAN1D1 /dev/sdb1
    ```

    c. Reboot and check ASM instance starts automatically and is able to see all the disks

    ```
    # oracleasm listdisks
    # oracleasm status
    Checking if ASM is loaded: yes
    Checking if /dev/oracleasm is mounted: yes
    ```

11. On other node, perform the following:

    a. Configure ASM and Disks

    ```
    # oracleasm configure -i
    # /usr/sbin/oracleasm init
    # oracleasm scandisks
    # oracleasm listdisks
    # oracleasm status
    Checking if ASM is loaded: yes
    Checking if /dev/oracleasm is mounted: yes
    ```

12. On racnode1, install Oracle grid infrastructure and it will automatically install it on the other node as well. While installing GRID infrastructure, configure OCR disk group OCRVD using External redundancy.

13. Create DATA, FRA, REDOa and REDOb disk groups using asmca.

14. Install Oracle database software using option “Oracle Real Application Clusters database installation” and configure “orcl” database instance. While installing the database software, OUI will automatically distribute the binaries to all the participating RAC nodes and configure database instance.
15. Verify GRID and Oracle database installation by logging into sqlplus and check the instance status.

Oracle Database 12c High Availability Solution – Two Node Oracle RAC Cluster using ASM mirroring between RAC nodes

This section describes the implementation of a two node RAC setup and how to deploy the ASM mirroring using Normal redundancy type. Figure 11 depicts the two node SimpliVity Federation configuration.

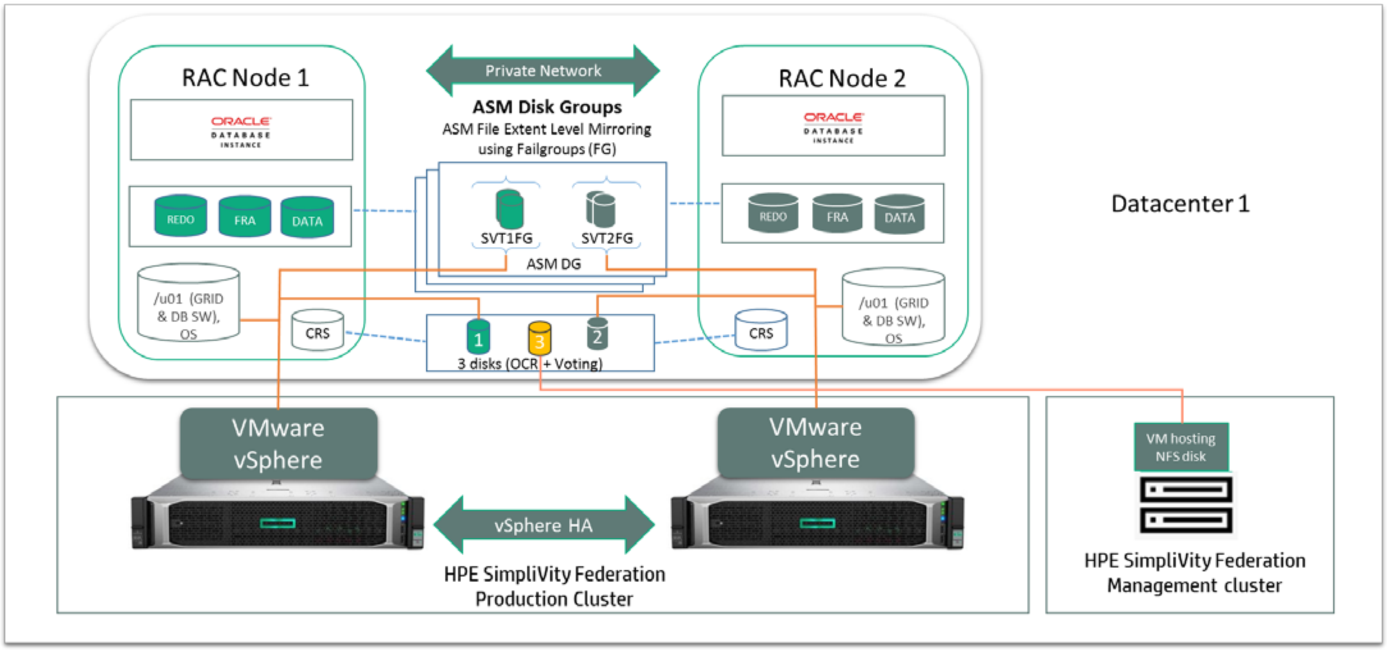


FIGURE 11. Oracle RAC – Two node cluster using Oracle ASM Failgroups

Follow these steps to create a RAC cluster:

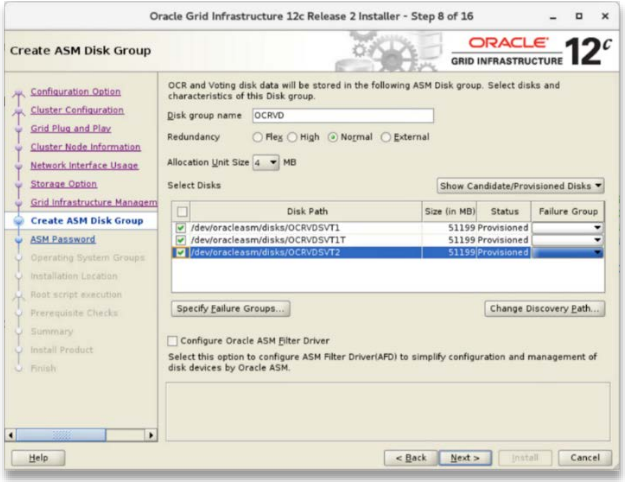
1. Follow the same procedure for creating VMs, installing and configuring OS and prerequisite for Oracle database as described in the previous scenario, where a single VM is sharing it’ disks. Use below table for disk configuration for the current scenario and configure additional NFS based quorum disk as well.

TABLE 7. ASM Mirror disks configuration for 2 node clusters

Used for ASM DG / Failgroup	SCSI Ctl / ASM Disk / Size	Details
DATA / SVT1FG	SCSI(1.#) / DATAN1D1 / 500GB	racnode1: Add this disk as local disk racnode2: Add this disk as an Existing disk
DATA / SVT2FG	SCSI(1.#) / DATAN2D1 / 500GB	racnode2: Add this disk as local disk racnode1: Add this disk as an Existing disk
REDOa / SVT1FG	SCSI(2:#) / REDOAN1D1 / 200GB	racnode1: Add this disk as local disk racnode2: Add this disk as an Existing disk
REDOa / SVT2FG	SCSI(2:#) / REDOAN2D1 / 200GB	racnode2: Add this disk as local disk racnode1: Add this disk as an Existing disk
REDOb / SVT1FG	SCSI(2:#) / REDOBN1D1 / 200GB	racnode1: Add this disk as local disk racnode2: Add this disk as an Existing disk

Used for ASM DG / Failgroup	SCSI Ctl / ASM Disk / Size	Details
REDOb / SVT2FG	SCSI(2:#) / REDOBN2D1 / 200GB	racnode2: Add this disk as local disk racnode1: Add this disk as an Existing disk
OCRVD	SCSI(3:#) / OCRVDSVT1 / 300GB	racnode1: Add this disk as local disk racnode2: Add this disk as an Existing disk
OCRVD	SCSI(3:#) / OCRVDSVT2 / 300GB	racnode2: Add this disk as local disk racnode1: Add this disk as an Existing disk
OCRVD	SCSI(3:#) / OCRVDSVT1T / 300GB	This disk will be used temporarily, instead of NFS disk, while configuring OCR diskgroup at the time of installing GRID software. Oracle Universal Installer (OUI) GUI does not allow you to configure NFS based Quorum disk in OCR diskgroup while installing the GRID software. racnode1: Add this disk as local disk racnode2: Add this disk as an Existing disk
OCRVD (Quorum Disk)	NFS share size of 310GB exported from Linux system running in management SimpliVity cluster.	Note: This step needs to be performed after installing GRID software. Refer to Oracle documentation Using standard NFS to support a third voting disk on a stretch cluster configuration for more details. On all RAC nodes: Mount 310GB NFS share on directory called /voting_disk with ownership of grid:asmdba and create a disk image called nfsvote using below command: dd if=/dev/zero of=/voting_disk/nfsvote bs=1M count=10000 Use below command to find out the “Size in MB” of other disks participating in OCRVD DG and use the same value as count. # kfod status=TRUE asm_diskstring="/dev/oracleasm/disks/*" disks=ALL If nfsvote disk size is not same as other disks within OCRVD, then INS-30521 message will be reported.

2. On racnode1, install Oracle grid infrastructure and it will automatically install it on the other node as well. While installing GRID infrastructure, configure OCRVD using Normal redundancy for 2 node cluster. Figure 12 shows 2 node RAC cluster OCR DG configuration.



Oracle Grid Infrastructure 12c Release 2 Installer - Step 8 of 16

Create ASM Disk Group

OCR and Voting disk data will be stored in the following ASM Disk group. Select disks and characteristics of this Disk group.

Disk group name: OCRVD

Redundancy: ○ Flex ○ High ● Normal ○ External

Allocation Unit Size: 4 MB

Select Disks: Show Candidate/Provisioned Disks

☐	Disk Path	Size (in MB)	Status	Failure Group
☑	/dev/oracleasm/disks/OCRVDSVT1	51199	Provisioned	
☑	/dev/oracleasm/disks/OCRVDSVT1T	51199	Provisioned	
☑	/dev/oracleasm/disks/OCRVDSVT2	51199	Provisioned	

☐ Configure Oracle ASM Filter Driver

Select this option to configure ASM Filter Driver(AFD) to simplify configuration and management of disk devices by Oracle ASM.

FIGURE 12. Oracle RAC – Two node cluster OCR DG configuration

3. After the GRID installation is completed, make sure the NFS volume is mounted at racnode1 and racnode2 and able to see the nfsvote disk.

4. Run asmca as grid user and add nfsvote disk as Quorum disk into OCRVD disk group and then remove OCRVDSVT1 from it.

5. Create DATA, REDOa and REDOb disk groups using failgroups. For example, in DATA disk group, SVT1FG represents disks from racnode1 and SVT2FG represents disks from racnode2.

```
create diskgroup DATA normal redundancy
failgroup SVT1FG disk
‘/dev/oracleasm/disks/DATAN1D1’name DATAN1D1
failgroup SVT2FG disk
‘/dev/oracleasm/disks/DATAN2D1’name DATAN2D1;
create diskgroup REDOa normal redundancy
failgroup SVT1FG disk
‘/dev/oracleasm/disks/REDOAN1D1’name REDON1D1
failgroup SVT2FG disk
‘/dev/oracleasm/disks/REDOAN2D1’name REDON2D1;
create diskgroup REDOb normal redundancy
failgroup SVT1FG disk
‘/dev/oracleasm/disks/REDOBN1D1’name REDON1D1
failgroup SVT2FG disk
‘/dev/oracleasm/disks/REDOBN2D1’name REDON2D1;
```

6. Install Oracle database software using option “Oracle Real Application Clusters database installation” and configure “orcl” database instance. While installing the database software, OUI will automatically distribute the binaries to all the participating RAC nodes and configure the database instance.
7. Verify the GRID and Oracle database installation by logging into sqlplus and checking the instance status.

Capacity and sizing

During the performance phase of testing, configuration options were chosen to maximize database performance. Two performance configurations were tested. The first configuration used 2 RAC nodes, with one RAC node per HPE SimpliVity node. The second configuration used 4 RAC nodes, with one RAC node per HPE SimpliVity node.

For Oracle configuration, per instance basis, the following are recommended:

- Disable automatic memory management, if applicable
- Set buffer cache memory size large enough, per your implementation, to avoid physical reads. For this testing, the buffer cache size was set to 128GB
- Create two large redo log file spaces of 200GB to minimize log file switching and reduce log file waits. One redo log file was placed in REDOa and the other was placed in REDOb
- Create an undo tablespace of 200GB
- Set the number of processes to a level that will allow all intended users to connect and process data. During our testing we use 3000 for this parameter setting
- Set the number of open cursors to a level that will not constrict Oracle processes. This was set to 3000.

Workload description for Oracle RAC scalability tests

The Oracle workload was tested using HammerDB, an open-source tool. The tool implements an OLTP-type workload with small I/O sizes of a random nature. The transaction results have been normalized and are used to compare test configurations. Other metrics measured during the workload come from the operating system and/or standard Oracle Automatic Workload Repository (AWR) statistics reports.

The OLTP test, performed on a 200GB database, was both moderately CPU and highly I/O intensive. The environment was tuned for maximum user transactions. After the database was tuned, the transactions per minute (TPM) were recorded for a varying number of Oracle connections. Because customer workloads vary so much in characteristics, the measurement was made with a focus on maximum transactions.

Oracle Enterprise Database version 12.2 was used in this test configuration.

Analysis and recommendations for Oracle RAC scalability

Testing was conducted with one and two node RAC configuration. All of these runs were on HPE SimpliVity 380 Gen 10 nodes with identical hardware configurations.

Figure 13 shows that CPU utilization for each configuration as the number of Oracle connections increased. As the second RAC node was added, the utilization decreased because the load is shared across two SimpliVity nodes. If more processing power is needed, additional compute nodes could be added instead of SimpliVity nodes. Adding more nodes will improve computing and memory capacity. Adding more nodes will not improve disk I/O as the disks are shared by only one node.

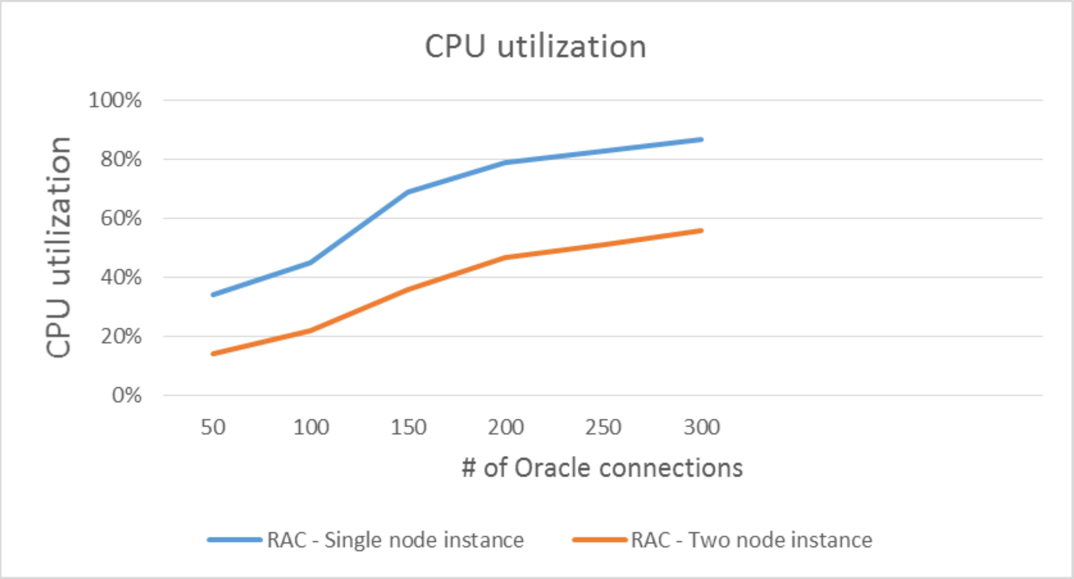
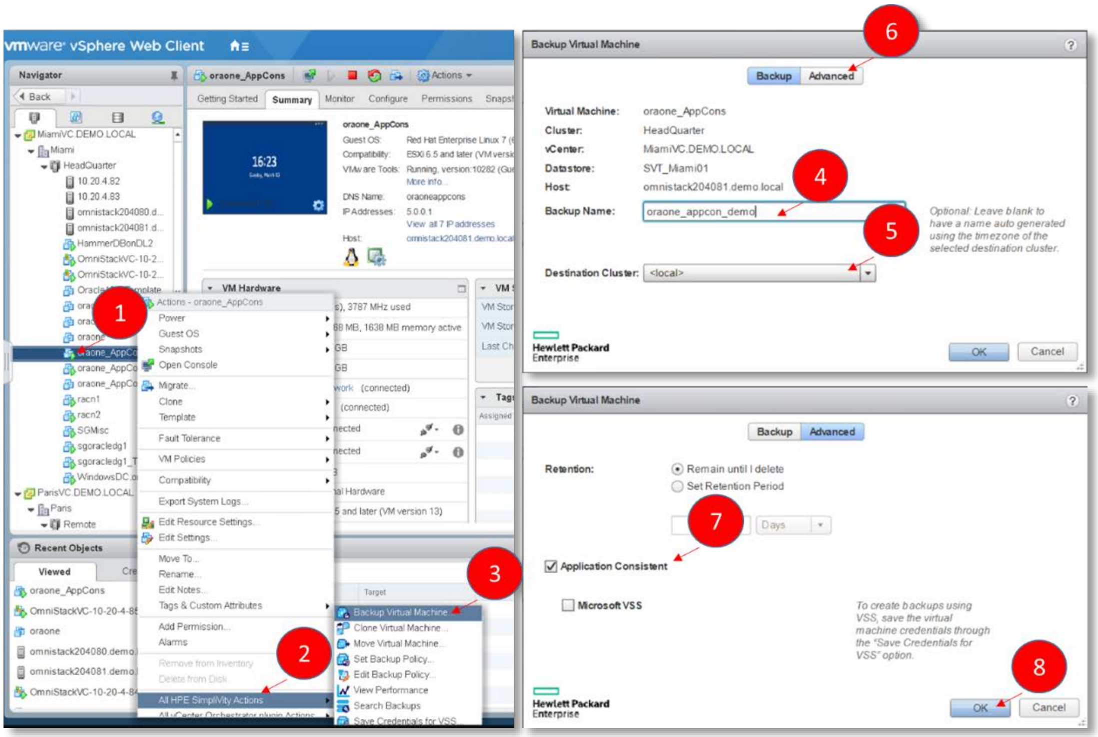


FIGURE 13. CPU utilization Vs number of Oracle connections

USE CASE 4: APPLICATION-CONSISTENT ORACLE DATABASE VM BACKUP AND RESTORE USING HPE SIMPLIVITY BUILT-IN DATA PROTECTION

An application-consistent backup is a backup of application data that allows the application to achieve a quiescent and consistent state. This type of backup captures the contents of the memory and any pending writes that occurred during the backup process. For example, to ensure that the backup includes all the data at a specific point in time, the backup allows the pending I/O operations to finish before committing them to the database. When the backup or snapshot is complete, the software notifies the database application to resume. A restoration of an application-consistent backup requires no additional work to restore the database application.

Figure 14 shows the manual way of taking an application consistent backup. This can also be automated using a backup policy.



Backup Virtual Machine

Backup | Advanced

Virtual Machine: oraone_AppCons
Cluster: HeadQuarter
vCenter: MiamiVC.DEMO.LOCAL
Datastore: SVT_Miami01
Host: omnistack204081.demo.local
Backup Name: oraone_appcon_demo

Optional: Leave blank to have a name auto generated using the timezone of the selected destination cluster.

Destination Cluster: <local>

Backup Virtual Machine

Backup | Advanced

Retention: ● Remain until I delete ○ Set Retention Period

Days

☑ Application Consistent

□ Microsoft VSS

To create backups using VSS, save the virtual machine credentials through the "Save Credentials for VSS" option.

FIGURE 14. HPE SimpliVity – Application consistent backup

Follow the steps to take an application consistent backup of a VM running Oracle Database.

1. Right click on VM
2. Click on All HPE SimpliVity Actions
3. Click on Backup Virtual Machine
4. Enter backup name
5. Select destination cluster
6. Click on Advanced button
7. Check Application Consistent checkbox and do not select Microsoft VSS as it is meant for Windows VMs
8. Click OK

When Application Consistent is selected, the process uses the default VMware snapshot method to create the backup. It preserves the state of the virtual machine and includes all the data at a specific point in time. However, if the application is experiencing high rates of I/O, this can take a long time to complete and may affect database performance. Therefore, it is recommended to take Application Consistent backup in low workload hours.

For Linux-based VMs, VMware tools or open-vm-tools can be used to perform specific operations before and after a SimpliVity application consistent backup is initiated for a VM. On a RHEL VM, the script /usr/sbin/pre-freeze-script is executed when the snapshot is created and /usr/sbin/post-thaw-script is executed when the snapshot is finalized. Ensure that these scripts are executable by the VMware tools user. ARCHIVELOG mode for the Oracle database must be set to ON. Note when crash consistent SimpliVity backup (when not selecting Application Consistent checkbox) is initiated, these scripts are not executed.

Following is the sample **pre-freeze-script**:

```
/usr/sbin/pre-freeze-script
#!/bin/sh
logger “pre-freeze-script called by open-vm-tools”
if [[ $EUID -eq 0 ]]; then
exec su oracle -c /home/oracle/scripts/pre-freeze.sql
fi
/home/oracle/scripts/pre-freeze.sql
#!/bin/bash
export ORACLE_HOME=/u01/app/oracle/product/12.2.0.1.0/db_1
$ORACLE_HOME/bin/sqlplus ”sys/Password1234@oradb as sysdba” <<EOF
spool /home/oracle/scripts/pre-freeze.log;
alter database begin backup;
spool off;
EOF
```

Following is the sample **post-thaw-script**:

```
/usr/sbin/post-thaw-script
#!/bin/sh
logger “post-thaw-script called by open-vm-tools”
if [[ $EUID -eq 0 ]]; then
exec su oracle -c /home/oracle/scripts/post-thaw.sql
fi
/home/oracle/scripts/post-thaw.sql
#!/bin/bash
export ORACLE_HOME=/u01/app/oracle/product/12.2.0.1.0/db_1
$ORACLE_HOME/bin/sqlplus ”sys/Password1234@oradb as sysdba” <<EOF
spool /home/oracle/scripts/post-thaw.log;
alter database end backup;
spool off;
EOF
```

The ALTER DATABASE BEGIN BACKUP command puts the database into hot backup mode. In backup mode, the database copies whole changed data blocks into the redo stream. After you take the database out of backup mode with the ALTER DATABASE END BACKUP statement, the database advances the data file checkpoint system change number (SCN) to the current database checkpoint SCN. When restoring a data file backed up in this way, the database asks for the appropriate set of redo log files to apply if recovery is needed. The redo logs contain all changes required to recover the data files and make them consistent.

Oracle RAC backup

It is recommended to use RMAN in RAC configurations for application consistency as a best practice. Application consistent backups are not recommended as they could potentially interfere with RAC node communications.

USE CASE 5: CRASH-CONSISTENT ORACLE DATABASE VM BACKUP USING HPE OMNISTACK REST API

HPE SimpliVity crash-consistent backup can be used for database applications. However, because they do not capture data in memory or any pending I/O operations, restoring data from a crash-consistent backup requires extra work, such as database recovery (RECOVER DATABASE) may be required using archived redo logs, before the database can be brought back online.

Oracle recommends to put the database in hot backup mode when RMAN is not used for backup operation. Hot backup mode is enabled using ALTER DATABASE BEGIN BACKUP and disabled using ALTER DATABASE END BACKUP. An automation script can be used to put the database in hot backup mode and initiate backup using HPE OmniStack REST API. Refer to Appendix A: Crash-consistent Oracle Database VM backup using HPE OmniStack REST API based script for more details on automation script.

Oracle database VM backup can be taken by putting the database in quiesced state using ALTER SYSTEM QUIESCE RESTRICTED command as well. In this state, Oracle database background processes may still perform updates for internal purposes even while the database is quiesced. Refer to Oracle documentation for more details. After VM backup has been performed, run ALTER SYSTEM UNQUIESCE command to come out of quiesced state.

SUMMARY

Oracle Database 12c is critical to the operation of many businesses. Unexpected downtime, poor performance, or data loss can lead to lost revenue, diminished user productivity and customer dissatisfaction. HPE SimpliVity hyperconverged infrastructure provides a highly resilient and scalable operating environment for Oracle Database 12c across a wide variety of use cases. This Reference Architecture has validated the following use cases:

- Disaster recovery solution for Oracle Database 12c using Oracle Data Guard integration with HPE Serviceguard for Linux and the advantages of integrating Serviceguard with Oracle Data Guard
- High availability solution using Oracle Real Application Cluster (RAC) deployment using one of the RAC cluster nodes sharing its data disk to other RAC nodes, and Oracle ASM Mirroring based deployment
- Application consistent cloning of Oracle Database VM using HPE SimpliVity cloning feature
- Application consistent backup of Oracle Database VM using HPE SimpliVity backup and restore feature

Our testing highlighted some of the key advantages of the HPE SimpliVity solution including:

- Accelerated data efficiency: OmniStack's inherent data efficiencies enabled rapid backup and restoration
- Built-in data protection: native backup and restore capabilities eliminated the need for special-purpose data protection applications or appliances
- Global unified management: OmniStack's VM-centric approach simplified administrative tasks such as configuring backup policies

In a real-world scenario the distinct data centers depicted in the Reference Architecture would be geographically distributed to enable disaster recovery and business continuity. Individual nodes within a data center would be scaled in an incremental fashion to meet specific customer requirements.

Implementing a proof of concept

As a matter of best practice for all deployments, Hewlett Packard Enterprise recommends implementing a proof of concept using a test environment that matches as closely as possible the planned production environment. In this way, appropriate performance and scalability characterizations can be obtained. For help with a proof of concept, contact a Hewlett Packard Enterprise Service representative (hpe.com/us/en/services/consulting.html) or your Hewlett Packard Enterprise partner.

APPENDIX A: CRASH-CONSISTENT ORACLE DATABASE VM BACKUP USING HPE OMNISTACK REST API BASED SCRIPT

The following script is a reference script, which implements crash-consistent backup and also enables hot backup mode for oracle database before backup is initiated, and disables hot backup mode after backup finished.

```
#!/bin/bash

# Usage

# ./svt_oravmbackup.sh <ovcip> <vmname> <backupname> <omniclustername> <ovclogin> <ovcpasswd>

#

# Pre-Requisite

# =============

# Install JSON Command Line Processor on the system where you are going to run this script.

# wget -O jq https://github.com/stedolan/jq/releases/download/jq-1.6/jq-linux64

# chmod +x ./jq

# cp jq /usr/bin

#

if [ ”$#” -ne 6 ]; then

    echo ”Please provide all parameters”

       echo ”Usage: svt_oravmbackup.sh ovcip vmname backupname omniclustername ovclogin ovcpasswd ”

       echo ”Example: svt_oravmbackup.sh 10.20.4.84 orasvr orasvr_bkp HeadQuarter
”administrator@vsphere.local” ”Password1234” ”

       exit

fi

ovc_ip=$1

vm_name=$2

bkp_name=$3

dest_name=$4   # The UID (omnistack_cluster_id) of the omnistack_cluster_name that stores the new
backup

ovc_login=$5

ovc_passwd=$6
```

```
arcchk=$(sqlplus / as sysdba <<EOF

select log_mode from v\$database;

exit;

EOF

)

echo $arcchk | grep -w NOARCHIVELOG

if [ $? = 0 ]

 then

 echo ”ARCHIVELOG mode is not enabled. Cannot proceed”

 exit

fi

# Get the access token first from OVC where VM is running

# Output Example: ”Authorization: Bearer abd4f32f-2535-4739-a3f2-24640c62945a”

#

oauth=”Authorization: Bearer ”$(curl -s -k https://simplivity@$ovc_ip/api/oauth/token -d
grant_type=password -d username=$ovc_login -d password=$ovc_passwd | jq ’.access_token’ | sed
’s/”//g’)

echo ”Auth done!”


# Get the vm id using vm name supplied in the argument

#

vm_id=$(curl -s -k -X GET -H ”$oauth” --header ”Accept: application/json”
”https://simplivity@$ovc_ip/api/virtual_machines?show_optional_fields=false&name=$vm_name&limit=500&of
fset=0&case=sensitive” | jq ’.virtual_machines[].id’ | sed ’s/”//g’)

echo ”Retrieved VM ID:” $vm_id


# Get the omnistack_cluster_id of omnistack_cluster_name, which user supplied as an argument

#

dest_id=$(curl -s -k -X GET -H ”$oauth” --header ”Accept: application/json”
”https://simplivity@$ovc_ip/api/omnistack_clusters?show_optional_fields=false&name=$dest_name&limit=50
0&offset=0&case=sensitive” | jq ’.omnistack_clusters[].id’ | sed ’s/”//g’)

echo ”Omnistack Cluster ID:” $dest_id


# Enabling Hot Backup Mode

#
```

```
bbchk=$(sqlplus / as sysdba <<EOF

alter database begin backup;

exit;

EOF

)

echo $bbchk | grep -w ”Database altered.”

if [ $? = 0 ]

 then

 echo ”Begin Backup successful!”

fi


# Display the current backup mode state to Admin

#

echo ”Backup mode has been enabled (STATUS: ACTIVE) before taking Simplivity Backup.”

sqlplus / as sysdba <<EOF

select * from v\$backup;

exit;

EOF


# Take Backup

task_id=$(curl -s -k -X POST -H ”$oauth” --header ”Content-Type: application/vnd.simplivity.v1.7+json”
--header ”Accept: application/json” -d ”{\”backup_name\”: \”$bkp_name\”, \”destination_id\”:
\”$dest_id\”, \”app_consistent\”: false, \”consistency_type\”: \”NONE\”,\”retention\”: 0}”
”https://simplivity@$ovc_ip/api/virtual_machines/$vm_id/backup” | jq '.task.id’ | sed 's/”//g’)

echo ”Backup initiated..”

echo ”The backup task id is $task_id”

while true

do

       sleep 2

       # Check whether backup task has been moved to ”COMPLETED” state

       current_task_state=$(curl -s -k -X GET -H ”$oauth” --header ”Accept: application/json”
”https://simplivity@$ovc_ip/api/tasks/$task_id” | jq '.task.state’ | sed 's/”//g’)

       echo ”Current State of Backup Task: $current_task_state”

       if [ $current_task_state = ”COMPLETED” ]
```

```
        then
        ebchk=$(sqlplus / as sysdba <<EOF
        alter database end backup;
        exit;
EOF
)
        echo $ebchk | grep -w ”Database altered.”
        if [ $? = 0 ]
        then
        echo ”End Backup successful ”
        echo ”Checking the current status of backup mode - Should be STATUS: NOT ACTIVE”
        sqlplus / as sysdba <<EOF
        select * from v\$backup;
        exit;
EOF
        echo ”Backup Operation Completed Successfully.”
        fi
        # Backup is done and alter database end backup is successful. Exiting.
        exit
        fi
        echo ”Looping until backup is COMPLETED”
done
```

APPENDIX B: BILL OF MATERIALS

The following BOMs contain electronic license to use (E-LTU) parts. Electronic software license delivery is now available in most countries. Hewlett Packard Enterprise recommends purchasing electronic products over physical products (when available) for faster delivery and for the convenience of not tracking and managing confidential paper licenses. For more information, please contact your reseller or a Hewlett Packard Enterprise representative.

NOTE
Part numbers are at time of publication/testing and subject to change. The bill of materials does not include complete support options or other rack and power requirements. If you have questions regarding ordering, please consult with your HPE Reseller or HPE Sales Representative for more details. hpe.com/us/en/services/consulting.html

TABLE B1. Bill of Materials (per HPE SimpliVity 380 Gen10 node)

Qty	Product#	Product Description
1	Q8D81A	HPE SimpliVity 380 Gen10 Node
1	Q8D81A 001	HPE SimpliVity 380 Gen10 VMware Solution
1	826880-L21	HPE DL380 Gen10 Intel Xeon-Gold 6142 (2.6GHz/16-core/150W) FIO Processor Kit
1	826880-B21	HPE DL380 Gen10 Intel Xeon-Gold 6142 (2.6GHz/16-core/150W) FIO Processor Kit
1	826884-B21 0D1	Factory Integrated
2	Q8D88A	HPE SimpliVity 384G 6 DIMM FIO Kit
1	Q5V86A	HPE SimpliVity 380 for 6000 Series Small Storage Kit
1	873209-B21	HPE DL38X Gen10 x8/x16/x8 PCIe NEBS Riser Kit
1	873209-B21 0D1	Factory Integrated
1	P01366-B21	HPE 96W Smart Storage Battery (up to 20 Devices) with 145mm Cable Kit
1	P01366-B21 0D1	Factory Integrated
1	804331-B21	HPE Smart Array P408i-a SR Gen10 (8 Internal Lanes/2GB Cache) 12G SAS Modular Controller
1	804331-B21 0D1	Factory Integrated
1	700751-B21	HPE FlexFabric 10Gb 2-port 534FLR-SFP+ Adapter
1	700751-B21 0D1	Factory Integrated
2	830272-B21	HPE 1600W Flex Slot Platinum Hot Plug Low Halogen Power Supply Kit
2	830272-B21 0D1	Factory Integrated
1	BD505A	HPE iLO Advanced 1-server License with 3yr Support on iLO Licensed Features
1	BD505A 0D1	Factory Integrated
1	Q8A68A	HPE OmniStack 16-22c 2P Small SW
1	733664-B21	HPE 2U Cable Management Arm for Easy Install Rail Kit
1	733664-B21 0D1	Factory Integrated
1	867809-B21	HPE Gen10 2U Bezel Kit
1	867809-B21 0D1	Factory Integrated
1	826703-B21	HPE DL380 Gen10 SFF Systems Insight Display Kit
1	826703-B21 0D1	Factory Integrated
1	733660-B21	HPE 2U Small Form Factor Easy Install Rail Kit
1	733660-B21 0D1	Factory Integrated
1	H1K92A3	HPE 3Y Proactive Care 24x7 SVC
1	H1K92A3 R2M	HPE iLO Advanced Non Blade - 3yr Support
1	H1K92A3 Z9X	HPE SVT 380 Gen10 Node (1 Node) Support
1	H1K92A3 ZC0	HPE OmniStack 16-22c 2P Small Support
1	HA114A1	HPE Installation and Startup Service
1	HA114A1 5LY	HPE SimpliVity 380 HW Startup SVC
1	HA124A1	HPE Technical Installation Startup SVC
1	HA124A1 5LZ	HPE SVT 380 for VMware Remote SW St SVC

TABLE 10. Bill of Material for HPE Serviceguard for Linux

Qty	Product#	Product Description
4	R1T32AAE	HPE Serviceguard for Linux x86 Enterprise 1yr Subscription 24x7 Support PSL E-LTU (One SGLX LTU per Socket)

RESOURCES AND ADDITIONAL LINKS

HPE Reference Architectures, https://www.hpe.com/docs/reference-architecture

HPE Servers, hpe.com/servers

HPE Storage, hpe.com/storage

HPE Networking, hpe.com/networking

HPE GreenLake Advisory and Professional Services, https://www.hpe.com/us/en/services/consulting.html

HPE SimpliVity 380 Gen10 QuickSpecs, https://h20195.www2.hpe.com/v2/GetDocument.aspx?docname=a00021989enw

Oracle Data Guard Concepts and Administration,

https://docs.oracle.com/en/database/oracle/oracle-database/12.2/sbydb/introduction-to-oracle-data-guard-concepts.html#GUID-5E73667D-4A56-445E-911F-1E99092DD8D7

How To Calculate the Required Network Bandwidth Transfer Of Redo In Data Guard Environments (Doc ID 736755.1),

https://support.oracle.com/knowledge/Oracle%20Database%20Products/736755_1.html

Oracle Real Application Clusters Administration and Deployment Guide,

https://docs.oracle.com/en/database/oracle/oracle-database/12.2/racad/real-application-clusters-administration-and-deployment-guide.pdf

HPE Serviceguard Toolkit for Oracle Data Guard on Linux User Guide,

https://support.hpe.com/hpsc/doc/public/display?docId=emr_na-a00018045en_us

To help us improve our documents, please provide feedback at hpe.com/contact/feedback.

LEARN MORE AT

https://www.hpe.com/us/en/integrated-systems/simplivity.html

a00071652enw, April 2021