Virtualizing Oracle Workloads with VMware vSphere Virtual Volumes on VMware Hybrid Cloud

REFERENCE ARCHITECTURE

Table of contents

Executive Summary

Business Case

Business-critical databases are among the last workloads to be virtualized in most enterprises, primarily because of the challenges posed as workloads grow and scale. In a typical virtualization project, once a proof of concept (POC) is successfully run, development and testing and staging databases come relatively easy, leaving production databases as the final hurdle to overtake.

Customers have successfully run their business-critical Oracle workloads with high performance demands on VMware vSphere® for many years. Virtualization of mission-critical databases adds layers of complexity to the infrastructure, however, making common operations like backup and recovery, cloning and other day-to-day activities difficult. The most efficient storage operations for mission-critical databases are offloaded to the storage array.

Concerns that often delay virtualization of business-critical database workloads include:

- Difficulties of meeting strict SLAs for performance with typically slow traditional storage.
- Rapid database growth and the need to reduce backup windows to meet performance and business SLAs.
- The size of modern databases makes it harder to regularly clone and refresh data from production to QA and other environments.
- While storage-based replication speed is superior, its array LUN-level granularity causes unnecessary data to be copied over when replicating select databases over metro distance.
- Databases of different levels of criticality need different storage performance characteristics and capabilities.
- Database and systems administrators continue to debate the utility of filesystems versus raw devices and VMware vSphere® Virtual Machine File System (VMFS) versus Raw Device Mapping, primarily due to deficiencies that existed in the past with virtualization.

Solution Overview

VMware vSphere Virtual Volumes™ (vVols) addresses the business challenges discussed in the previous section regarding business-critical databases' day-to-day operations like backup and recovery, cloning, and database provisioning.

vVols is an integration and management framework that virtualizes SAN/NAS arrays, enabling a more efficient operational model that is optimized for virtualized environments and centered on the application instead of the infrastructure.

vVols exposes virtual disks as native storage objects and enables array-based operations at the virtual disk level. vVols transform the data plane of SAN/NAS devices by aligning storage consumptions and operations with the VM. In other words, vVols make SAN/NAS devices VM-aware and unlocks the ability to leverage array-based data services with a VM-centric approach at the granularity of a single virtual disk.

vVols allows customers to leverage the unique capabilities of their current storage investments and transition without disruption to a simpler and more efficient operational model optimized for virtual environments that work across all storage types. It simplifies the delivery of storage service levels to individual applications by providing finer control of hardware resources and native array-based data services that can be instantiated with VM granularity.

vVols are very useful for backup and recovery, cloning, enhanced storage policy-based management (SPBM) control and other operations for business-critical databases.

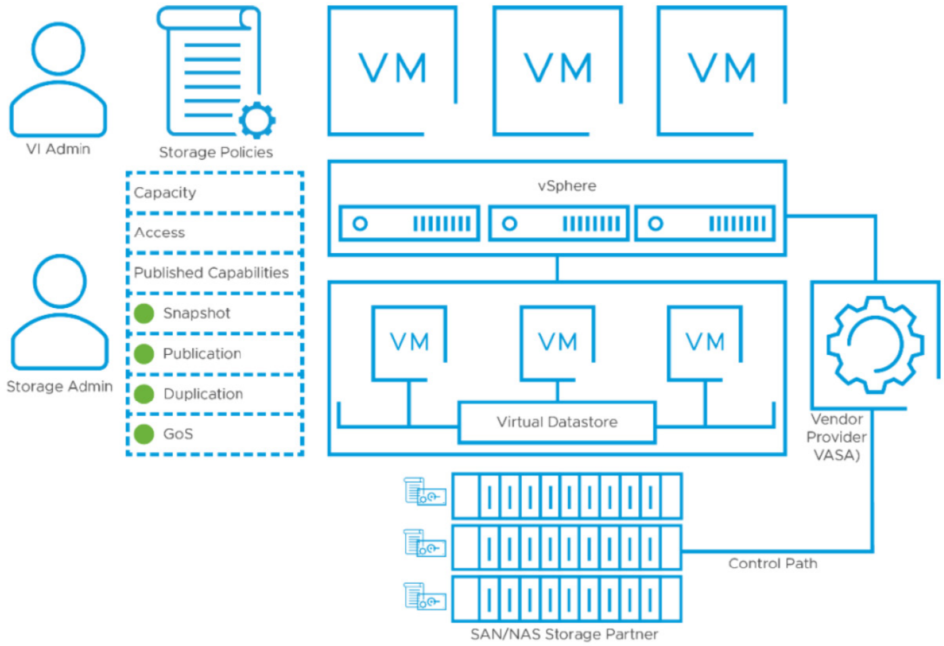


FIGURE 1. VMWARE VSPHERE VIRTUAL VOLUMES OVERVIEW

Key Results

vVols provides simplicity, speed, performance and granularity of operations to business-critical Oracle workloads in the following ways:

- **Flexible consumption at the logical level** – vVols virtualize SAN and NAS devices by abstracting physical hardware resources into logical pools of capacity (i.e., virtual datastore) that can be flexibly consumed and configured to span a portion of one to several storage arrays. The vVols virtual datastore defines capacity boundaries and access logic and exposes a set of data services accessible to VMs provisioned in the pool. vVols virtual datastores are purely logical constructs that may be configured on the fly – when needed and without disruption – and do not require file system formatting.
- **Precise Control at a VM level** – It becomes possible to execute storage operations with VM granularity and to provision native array-based data services to individual VMs. This allows admins to provide the right storage service levels to each VM. For example, it enables storage-level snapshots with VM granularity due to the fact that every VMDK is natively represented by an independent vVol.
- **Efficient Operations through Automation** – SPBM allows capturing storage service level requirements, such as capability, performance, or availability, in the form of logical templates (policies) to which VMs are associated. SPBM automates VM placement by identifying available datastores that meet policy requirements and, coupled with vVols, dynamically instantiates necessary data services. Through policy enforcement, SPBM also automates service-level monitoring and compliance throughout the lifecycle of the VM.
- **Simplified Storage Operations** – For both the virtualization admin and storage admin, vVols greatly simplifies management over the existing operational model. vVols allow the separation of provisioning and consumption of storage for VMs.
- **Simplified Delivery of Storage Service Levels** – With vVols, it is easier to deliver and enable the right storage service levels according to the specific requirements of individual applications. With granular control over storage resources and native array-based data services at the VM level, administrators can create specific policy combinations and precisely deliver storage service levels on a case-by-case basis. Additionally, policy-driven automation enwsures desired service levels are met, and enables dynamic adjustments in real time when needed, making it possible to adapt to ever-changing application and business requirements quickly.
- **Improved Resource utilization** – Precisely mapping application requirements with storage resources when they are needed fundamentally eliminates over-provisioning issues. By virtualizing the storage infrastructure, vVols enables more flexible, VM-centric consumption of storage capacity and data services. Through automation, it enables dynamic adjustments in real time. This is in contrast with the legacy operational model, in which resources had to be pre-allocated and were hard to change, contributing to inefficient upfront investments in capacity and misuse of data services that create inefficient use of infrastructure resources over time.

Introduction

Purpose

This reference architecture validates the capability of vVols to provide flexible storage consumption, granularity of control at a VM level, simplicity and efficiency of operations through automation, simplified storage operations and delivery of storage service levels, and improved resource utilization to business-critical Oracle workloads.

Audience

This reference architecture is intended for Oracle database administrators (DBAs) and virtualization and storage architects involved in planning, architecting, and administering business-critical Oracle environments on the VMware software-defined datacenter (SDDC) platform.

Terminology

The following terms are used throughout this paper:

Term	Definition
Oracle Single Instance	Oracle Single-Instance database consists of a set of memory structures, background processes, and physical database files, which serves the database users.
Oracle Clusterware	Oracle Clusterware is a portable cluster software that allows clustering of independent servers so that they cooperate as a single system.
Oracle Automatic Storage Management (Oracle ASM)	Oracle ASM is a volume manager and a file system for Oracle database files that support Single-Instance Oracle Database and Oracle Real Application Cluster (RAC) configurations.
Oracle ASMLIB and Oracle ASMFD	Oracle ASMLIB maintains permissions and disk labels that are persistent on the storage device, so that the label is available even after an operating system upgrade. Oracle ASMFD helps prevent corruption in Oracle ASM disks and files within the disk group.

TABLE 1. TERMINOLOGY

Technology Overview

Overview

This section provides an overview of the technologies used in this solution:

- VMware vSphere®
- VMware vSphere® Virtual Volumes™ (vVols)
- VMware vSphere Metro Storage Cluster (vMSC)
- VMware Virtual Disk Provisioning Policies
- VMware Multi-Writer Attribute for Shared VMDKs
- Shared Disks Using vVols
- Storage Policy-Based Management (SPBM)
- vVols Storage Policy

- Oracle Database Architecture
- Oracle ASM, ASMLIB and ASMFD
- Oracle Clusterware
- Oracle Real Application Cluster (RAC)
- Oracle RAC One Node
- Extended Oracle RAC
- Oracle RAC on vVols
- Extended Oracle RAC on vVols

VMware vSphere

VMware vSphere, the industry-leading virtualization and cloud platform, is the efficient and secure platform for hybrid clouds, accelerating digital transformation by delivering simple and efficient management at scale, comprehensive built-in security, a universal application platform, and a seamless hybrid cloud experience. The result is a scalable, secure infrastructure that provides enhanced application performance and can be the foundation of any cloud.

As the next-generation infrastructure for next-generation applications, vSphere 7.0 has been rearchitected with native Kubernetes, enabling IT admins to use VMware vCenter Server® to operate Kubernetes clusters through namespaces. VMware vSphere with Tanzu allows IT admins to leverage their existing skillset to deliver self-service infrastructure access to their DevOps teams, while providing observability and troubleshooting of Kubernetes workloads. vSphere 7 provides an enterprise platform for both traditional and modern applications, enabling customers and partners to deliver a developer-ready infrastructure, scale without compromise, and simplify operations.

Learn more about *VMware vSphere 7.0.*

VMware vSphere Virtual Volumes (vVols)

Historically, vSphere storage management used a datastore-centric approach, with the datastore providing the lowest granularity level for data management. With a single datastore containing multiple VMs, each with its own unique requirements, meeting these individual requirements is difficult.

Using vVols, an individual VM (not the datastore) becomes a unit of storage management, while storage hardware gains complete control over virtual disk content, layout, and management.

vVols helps to improve granularity. It helps to differentiate VM services on a per-application level by offering a new approach to storage management. Rather than arranging storage around features of a storage system, vVols arranges storage around the needs of individual VMs, making storage VM-centric.

vVols maps virtual disks and their derivatives, clones, snapshots, and replicas, directly to objects (i.e., virtual volumes) on a storage system. This mapping allows vSphere to offload intensive storage operations such as snapshot, cloning, and replication to the storage system.

By creating a volume for each virtual disk, policies can be set at the optimal level. One can decide in advance what the storage requirements of an application will be and communicate these requirements to the storage system. The storage system creates an appropriate virtual disk based on these requirements. For example, if the VM requires an active-active storage array, one no longer must select a datastore that supports the active-active model. Instead, one can create an individual vVol that is automatically placed to the active-active array.

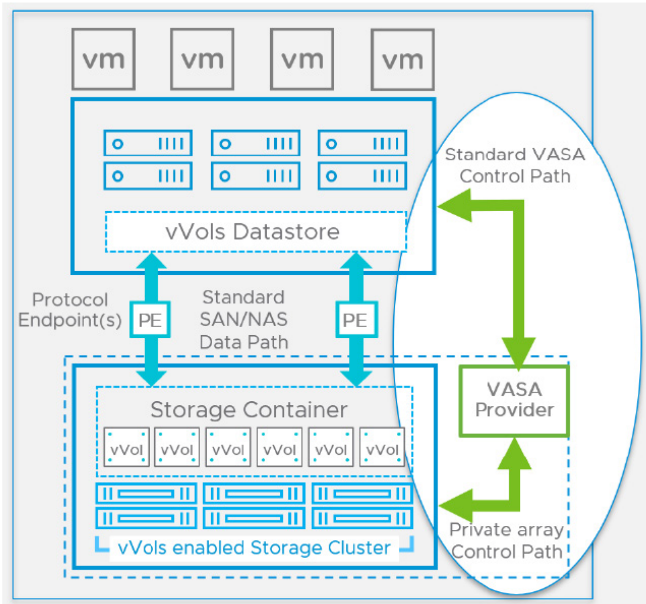


FIGURE 2. VMWARE VSPHERE CLUSTER WITH VIRTUAL VOLUMES

Learn more about *VMware Virtual Volumes.*

VMware recommends talking with your storage vendor to obtain guidance for vVols implementation, as each vendor accomplishes this differently.

A vVols datastore represents a storage container in vCenter Server and in the vSphere client.

After vCenter Server discovers storage containers exported by storage systems, they must be mounted as vVols datastores. The vVols datastores are not formatted in a traditional (e.g., VMFS datastores). These must be created as all vSphere functionalities (e.g., FT, HA, and DRS) require the datastore construct to function properly.

Review *VMware Docs* and *core.vmware.com/vVols* to learn more about VMware Virtual Volumes.

vVols Objects

vVols are stored natively inside a storage system that is connected to a VMware ESXi™ host through Ethernet or SAN. They are exported as objects by a compliant storage system and are managed entirely by hardware on the storage side.

Typically, a unique GUID identifies a vVol. vVols are not pre-provisioned but created automatically when VM management operations are performed. These operations include VM creation, cloning, and snapshotting. ESXi and vCenter Server associate one or more vVols with a VM.

The following types of vVols represent the core elements of the VM:

- **Data-vVol** – A data-vVol corresponds directly to each virtual disk VMDK file. As virtual disk files on traditional datastores, vVols are presented to VMs as SCSI disks. A data-vVol can be thick or thin-provisioned.

- **Config-vVol** – A config-vVol, or a home directory, represents a small directory that contains metadata files for a VM. The files include but are not exclusive to a .vmx file, descriptor files for virtual disks, and log files. The configuration vVol is formatted with a file system. When ESXi uses the SCSI protocol to connect to storage, configuration vVols are formatted with VMFS. With NFS protocol, configuration a vVol is presented as an NFS directory. Typically, it is thin-provisioned.
- **Swap-vVol** – The swap-vVol is created when a VM is powered on. It holds copies of VM memory pages that cannot be retained in memory. Its size is determined by the VM's memory size. A swap-vVol is thick-provisioned by default.
- **Snapshot-vVol** – A snapshot-vVol is an array-based snapshot that holds the contents of a VM and memory. It is thick-provisioned.
- **Other** – Other vVol are designed to address specific features. For example, a digest-vVol is created for content-based read cache (CBRC).

Typically, a VM creates a minimum of three vVol: a data-vVol, config-vVol, and swap-vVol. The maximum is determined by the number of virtual disks and snapshots residing on the VM.

By using different vVols for different VM components, admins can apply and manipulate storage policies at the finest level of granularity. For example, a vVol containing a log or database virtual disk can have a richer set of services than a vVol for the VM boot disk. Similarly, a snapshot-vVol can use a different storage tier compared to a current vVol.

Learn more about *VMware Virtual Volume object types.*

VMware Virtual Machine Snapshots

Snapshots preserve the state and data of a VM at the time the snapshot is taken. When a VM snapshot is captured, an image of the VM in a given state is copied and stored. Snapshots are useful when frequently reverting to a particular VM state and creating multiple VMs is undesirable.

Snapshots for Oracle databases on VMware vSphere can be performed in three ways:

- **Database** – using Oracle ACFS snapshots, for example, which is an online, read-only or read-write point-in-time copy of an Oracle ACFS file system. See *About Oracle ACFS Snapshots* for detailed information.
- **vSphere VM** – using VMware snapshots.
- **Storage** – using LUN-based snapshots available in a traditional storage array.

VMware vSphere, using VM snapshots, enables users to capture point-in-time state and data of a VM. This includes the VM's storage, memory, and other devices, such as virtual NICs.

Snapshots are useful for creating point-in-time state and data of a VM for backup or archival purposes and for creating test and rollback environments for applications.

For further information about using VM snapshots in a vSphere environment, see *Using Snapshots To Manage Virtual Machines.*

A VM snapshot can be taken through:

- **Web Client GUI** – see *Taking a Snapshot* for detailed information.
- **PowerCLI commands** – see *PowerCLI Reference: New Snapshot* for detailed information.

VMware Virtual Machine Clones

Cloning a VM creates a VM that is a copy of the original. The new VM is configured with the same virtual hardware, installed software, and other properties that were configured for the original VM.

Clones for Oracle databases on VMware vSphere can be performed in three ways:

- **Database** – using Oracle Enterprise Manager Cloud Control, for example, or classic cloning using RMAN backups. See *Cloning Oracle Databases and Pluggable Databases* for more information.

- **vSphere** – using VMware cloning technology.
- **Storage** – using traditional storage-array-based cloning.

There are two types of cloning operations performed in this guide:

- Cloning of an entire VM containing all VMDKs, including the operating system, Oracle binaries, and Oracle data VMDKs.
- Cloning the database VMDKs of a VM alone.

For further information about VM cloning in a vSphere environment, see *Clone a Virtual Machine.*

VMware Virtual Disk Provisioning Policies

When creating virtual disk, cloning a VM to a template, or migrating a VM with VMware vSphere® Storage vMotion®, it's possible to specify a provisioning policy for the virtual disk file. It's also possible to use Storage vMotion or cross-host Storage vMotion to transform virtual disks from one format to another.

Option	Description
Thick-Provision Lazy Zeroed	Creates a virtual disk in a default thick format. Space required for the virtual disk is allocated when the disk is created. Data remaining on the physical device is not erased during creation but is zeroed out on demand later on first write from the VM. VMs do not read stale data from the physical device.
Thick-Provision Eager Zeroed	A type of thick virtual disk that supports clustering features such as the multi-writer attribute for Oracle RAC. Space required for the virtual disk is allocated at creation time. In contrast to the thick-provision lazy zeroed format, the data remaining on the physical device is zeroed out when the virtual disk is created. It can take longer to create virtual disks in this format than to create other types of disks. Increasing the size of an eager zeroed thick virtual disk causes a significant stun time for the VM.
Thin-Provision	Use this format to save storage space. For the thin disk, provision as much datastore space as the disk would require based on the value entered for the virtual disk size. However, the thin disk starts small and at first, using only as much datastore space as the disk needs for its initial operations. If the thin disk needs more space later, it can grow to its maximum capacity and occupy the entire datastore space provisioned to it. Thin provisioning is the fastest method to create a virtual disk because it creates a disk with just the header information. It does not allocate or zero out storage blocks. Storage blocks are allocated and zeroed out when they are first accessed.

TABLE 2. VIRTUAL DISK PROVISIONING POLICIES

VMDK modes are reflected in the table below:

Option	Description
Dependent	Dependent disks are included in snapshots.
Independent-persistent	Disks in persistent mode behave like conventional disks on a physical computer. All data written to a disk in persistent mode is written permanently to the disk.

Independent– non-persistent	Changes to disks in non-persistent mode are discarded when a VM is turned off or reset. With non-persistent mode, a VM can be restarted with a virtual disk in the same state every time. Changes to the disk are written to and read from a redo log file that is deleted when the VM is turned off or reset.

TABLE 3. VMDK MODES

Learn more about *VMware virtual disk provisioning policies.*

VMware Multi-Writer Attribute for Shared VMDKs

VMFS is a clustered file system that disables (by default) multiple VMs from opening and writing to the same virtual disk (.vmdk file). This prevents more than one VM from inadvertently accessing the same .vmdk file. The multi-writer option allows VMFS-backed disks to be shared by multiple VMs. An Oracle Real Application Cluster (RAC) cluster using shared storage is a common use case.

VMware vSphere on VMFS, VVols (beginning with ESXi 6.5), network files system (NFS) datastores and VMware vSAN™ prevents multiple VMs from opening the same virtual disk (VMDK) in read-write mode.

Current restrictions of the multi-writer attribute documented in *KB 1034165* include:

- Storage vMotion is disallowed.
- Snapshots are not supported (snapshots of VMs with independent-persistent disks are supported, however).
- Changed block tracking (CBT) is not supported.
- Cloning, hot-extend virtual disk are not supported.

Independent-persistent mode is **NOT** required for enabling multi-writer attribute.

When working with VMware vSphere on VMFS, NFS datastores and vVols (beginning with ESXi 6.5), using multi-writer to share VMDKs for Oracle RAC requires:

- SCSI bus-sharing is set to **None**.
- VMKDs must be thick-provision eager zeroed (i.e., VMDKs cannot be thick-provision lazy zeroed or thin-provisioned).

Beginning with VMware vSphere 6.5, vVols 2.0, and vSphere APIs for Storage Awareness 3.0 (VASA), vVols are now validated to support Oracle RAC workloads delivering policy-based, VM-centric storage for Oracle RAC clusters.

For a detailed explanation, read *What's New in Virtual Volumes (vVols) 2.0.*

Learn more about *using shared disks with vVols here.*

Oracle RAC shared storage provisioning for vVols can be summarized as follows:

VMware Platform	Datastore	Version	RAC shared VMDK requirement for multi-writer attribute	Reference
VMware vSphere	vVol	ESXi 6.5 and above	Vendor specific	*KB 1034165* and *KB 2113013*

TABLE 4. ORACLE RAC SHARED STORAGE PROVISIONING

Storage Policy-Based Management (SPBM)

Storage policy-based management (SPBM) is a storage policy framework that helps administrators match VM workload requirements to storage capabilities. SPBM runs as an independent service in vCenter Server and helps to align storage with the application demands of VMs.

SPBM enables the following mechanisms:

- Advertisement of storage capabilities and data services offered by storage arrays and other entities, such as I/O filters.
- Bidirectional communication between ESXi and vCenter Server on one side and storage arrays and entities on the other.
- VM provisioning based on VM storage policies.

Administrators build policies by selecting the desired capabilities of the underlying storage array. The SPBM engine interprets the storage requirements of individual applications specified in policies associated with individual VMs and dynamically composes the storage service, placing the VM on the right storage tier, allocating capacity, and instantiating the necessary data services (e.g., snapshots, replication).

As an abstraction layer, SPBM abstracts storage services delivered by vVols, vSAN, I/O filters, or other storage entities.

Rather than integrating with each individual type of storage and data service, SPBM provides a universal framework for multiple types of storage.

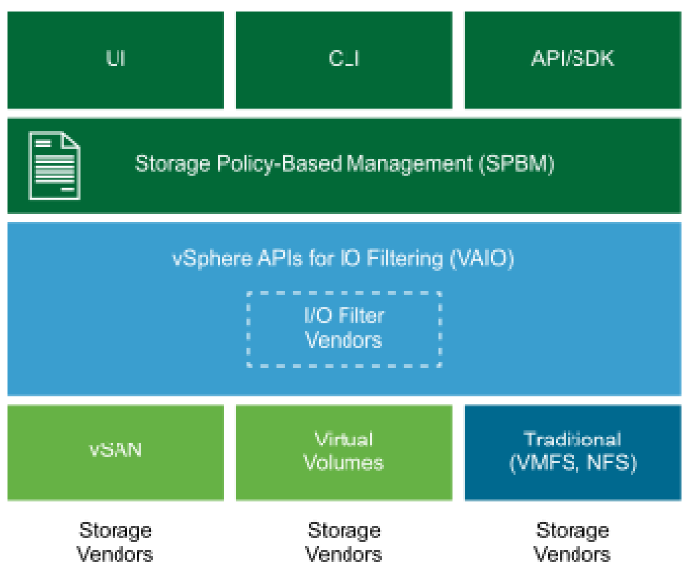


FIGURE 3. SPBM UNIVERSAL FRAMEWORK

Learn more about *VMware storage policy-based management (SPBM).*

vVols Storage Policy

For vVols, VMware provides a default storage policy that contains no rules or storage requirements (vVol No Requirements Policy). This policy is applied to VM objects when another policy for the VM on the vVols datastore is not specified.

The policy allows storage arrays to determine the optimum placement for the VM objects.

The default vVol No Requirements Policy provided by VMware has the following characteristics:

- The policy cannot be deleted, edited, or cloned.
- It is compatible only with vVols datastores.
- Users can create a VM storage policy for vVols and designate it as the default.

The default vVol No Requirements Policy is shown below:

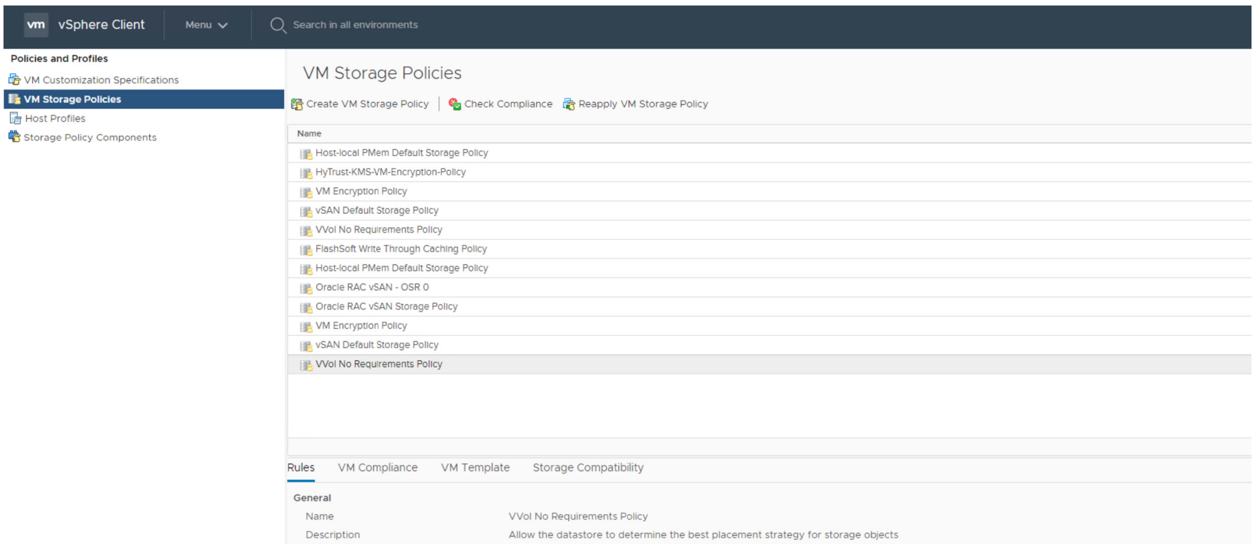


FIGURE 4. VMWARE VVOL NO REQUIREMENTS POLICY

The Pure Storage FlashArray storage policy is shown below:

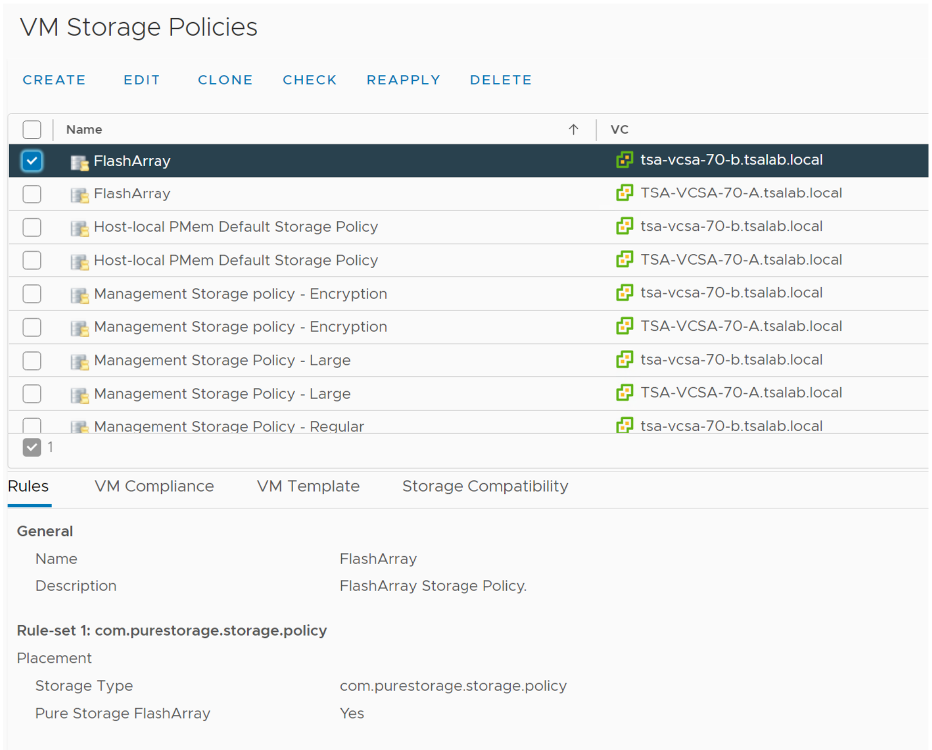


FIGURE 5. PURE STORAGE FLASHARRAY STORAGE POLICY

Learn more about *VMware vVols storage policy*.

Storage Providers

A storage provider is a software component that is offered by VMware or developed by a third party through VASA, also referred to as a VASA provider. Storage providers integrate with various storage methods or types, including both external physical storage and storage abstractions (e.g., vSAN and vVols). Storage providers can also support software solutions (e.g., I/O filters).

Generally, vCenter Server and ESXi use storage providers to obtain information about storage configuration, status, and the storage data services offered in an environment. This information appears in the vSphere client and aids decision-making with regard to VM placement, setting of storage requirements, and monitoring of a storage environment.

A VASA for a Pure Storage x50 FlashArray is shown below:

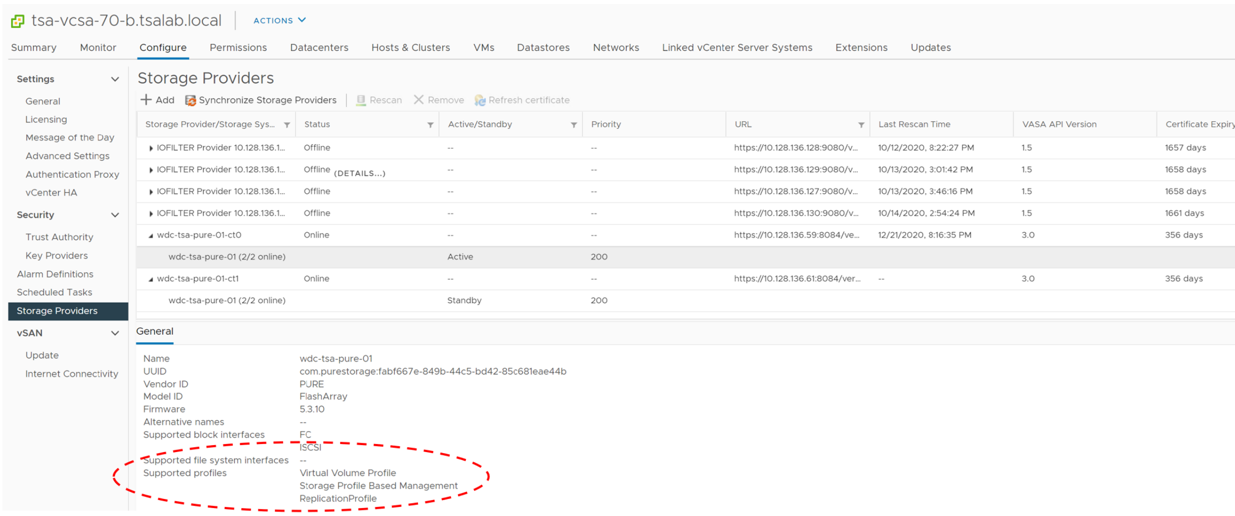


FIGURE 6. VASA FOR PURE STORAGE X50 FLASHARRAY

Learn more about *using storage providers.*

VMware vSphere Metro Storage Cluster (vMSC)

A VMware vSphere Metro Storage Cluster (vMSC) is a specific storage configuration commonly referred to as stretched storage clusters or metro storage clusters. These configurations are usually implemented in environments where disaster and downtime avoidance are a key requirement.

A vMSC configuration is a specific storage configuration that combines replication with array-based clustering. These solutions are typically deployed in environments where the distance between datacenters is limited, often in metropolitan or campus environments.

vMSC infrastructures are implemented with a goal of reaping the same benefits that vSphere HA clusters provide to a local site, in a geographically dispersed model with two datacenters in different locations. A vMSC infrastructure is essentially a stretched cluster. The architecture is built on the premise of extending what is defined as “local” in terms of network, storage and compute to enable these subsystems to span geographies, presenting a single and common base infrastructure set of resources to the vSphere cluster at both sites. It in essence stretches storage, network and compute between sites.

The primary benefit of a stretched cluster model is that it enables fully active and workload-balanced data centers to be used to their full potential and it allows for an extremely fast recovery in the event of a host or even full site failure. The capability of a stretched cluster to provide this active balancing of resources should always be the primary design and implementation goal. Although often associated with disaster recovery, vMSC infrastructures are not recommended as primary solutions for pure disaster recovery.

vMSC solutions are classified into two distinct categories. These categories are based on a fundamental difference in how hosts access storage. It is important to understand the different types of stretched storage solutions because this influences design considerations.

The following two main categories are as described on the VMware hardware compatibility list:

- **Uniform host access configuration** – ESXi hosts from both sites are all connected to a storage node in the storage cluster across all sites. Paths presented to ESXi hosts are stretched across a distance.
- **Non-uniform host access configuration** – ESXi hosts at each site are connected only to storage node(s) at the same site. Paths presented to ESXi hosts from storage nodes are limited to the local site.

Learn more about *VMware vSphere Metro Storage Cluster.*

Pure ActiveCluster VMware vSphere Metro Storage Cluster

Pure Storage Purity//FA ActiveCluster is a fully symmetric active-active bidirectional replication capability for FlashArrays that provides zero RPO zero and automatic transparent failover for zero RTO.

ActiveCluster spans multiple sites enabling clustered arrays and clustered ESXi hosts to be used to deploy flexible active/active datacenter configurations for applications like Extended Oracle RAC.

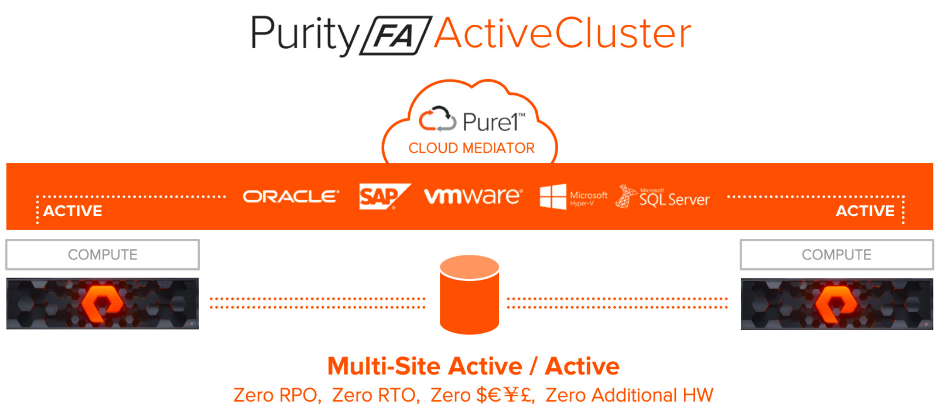


FIGURE 7. PURE STORAGE PURITY//FA ACTIVECLUSTER

Key features of ActiveCluster include symmetric active-active replication, synchronous operation, transparent failover, transparent mediation, and simple management model.

Core Components

Purity ActiveCluster is composed of three core components: the Pure1 Cloud Mediator, active/active clustered array pairs, and stretched storage containers.

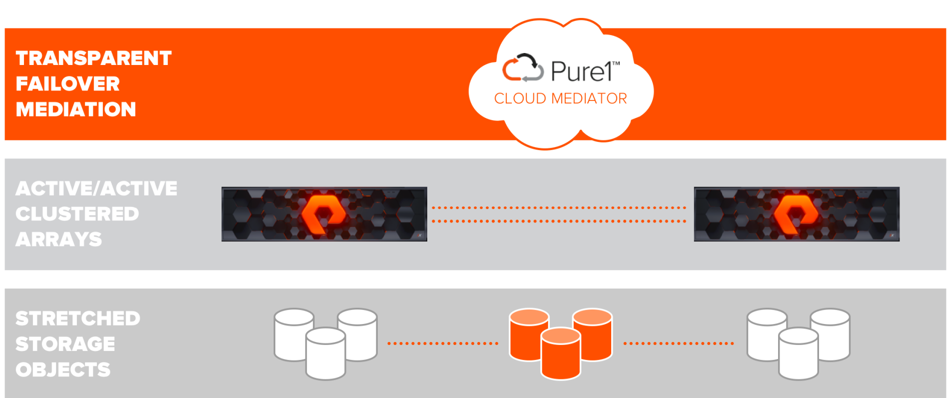


FIGURE 8. PURE1 CLOUD MEDIATOR

- **The Pure1 Cloud Mediator** – A required component of the solution that is used to determine which array will continue data services should an outage occur in the environment. An on-premises mediator VM is also available.
- **Active/Active clustered FlashArrays** – Utilize synchronous replication to maintain a copy of data on each array and present those as one consistent copy to hosts that are attached to either, or both, arrays.
- **Stretched storage objects** – Management containers, called pods, that collect storage objects such as volumes into groups that are stretched between two arrays.

Learn more about *the Pure Storage ActiveCluster VMware vMSC.*

Hybrid and Multi-Cloud as the VMware Cloud

The term hybrid cloud describes the use of both private and public cloud platforms, working in conjunction. It can refer to any combination of cloud solutions that work together on-premises and off-site to provide cloud computing services to a company. A hybrid cloud environment allows organizations to benefit from the advantages of both types of cloud platforms and choose which cloud to use based on specific data needs.

A multi-cloud environment is as its name suggests, reflecting multiple and disparate cloud offerings and forms, all of which are part of the ubiquitous VMware Cloud.

VMware's *hybrid cloud* portfolio offers a combination of solutions that enable organizations to easily extend, protect, or replace on-premises infrastructure. These hybrid cloud offerings are built on an SDDC architecture, leveraging VMware's industry-leading compute, networking, and storage virtualization technologies.

Any combination of clouds powered by VMware creates a common operating environment across VMware-based on-premises private clouds and VMware-based public clouds. Cloud solutions from VMware Cloud Provider Partners (VCPP) including IBM, Oracle, Microsoft, Google, Amazon Web Services (AWS) and others. Native public clouds such as AWS, Azure, Oracle and Google Cloud Platform using VMware technologies including VMware Cloud Foundation™, VMware vRealize® and VMware Cloud™ Services, along with on-premises managed cloud services such as VMware Cloud on DellEMC, form the core of VMware Cloud offerings.

This approach enables a diverse set of use cases, including regional capacity expansion, disaster recovery, application migration, data center consolidation, new application development and burst capacity.

Learn more about *VMware Hybrid Cloud.*

VMware Cloud on AWS

VMware Cloud on AWS is an on-demand service that enables customers to run applications across vSphere-based cloud environments with access to a broad range of AWS services. Powered by VMware Cloud Foundation, this service integrates vSphere, vSAN and VMware NSX® along with VMware vCenter management, and is optimized to run on dedicated, elastic, bare-metal AWS infrastructure.

With VMware Hybrid Cloud Extension™ (HCX), customers can easily and rapidly perform large-scale bi-directional migrations between on-premises and VMware Cloud on AWS environments.

With the same architecture and operational experience on-premises and in the cloud, IT teams can now quickly derive instant business value from use of the AWS and VMware hybrid cloud experience. VMware Cloud on AWS is ideal for enterprise IT infrastructure and operations organizations looking to migrate on-premises vSphere-based workloads to the public cloud, consolidate and extend data center capacities, and optimize, simplify and modernize their disaster recovery solutions.

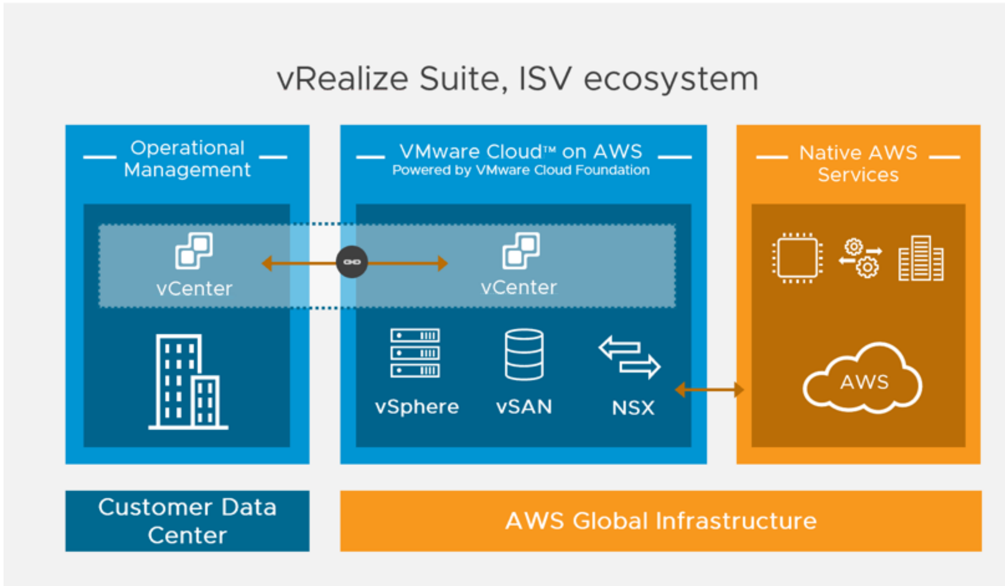


FIGURE 9. VMWARE CLOUD ON AWS

Learn more about *VMware Cloud on AWS.*

Pure Storage Cloud Block Store (CBS)

Pure Storage Cloud Block Store (CBS) for AWS is block storage delivered natively in the cloud. CBS is a software-defined storage solution powered by the Purity Operating Environment (POE), which uses the native resources of AWS to provide block storage capabilities to cloud workloads.

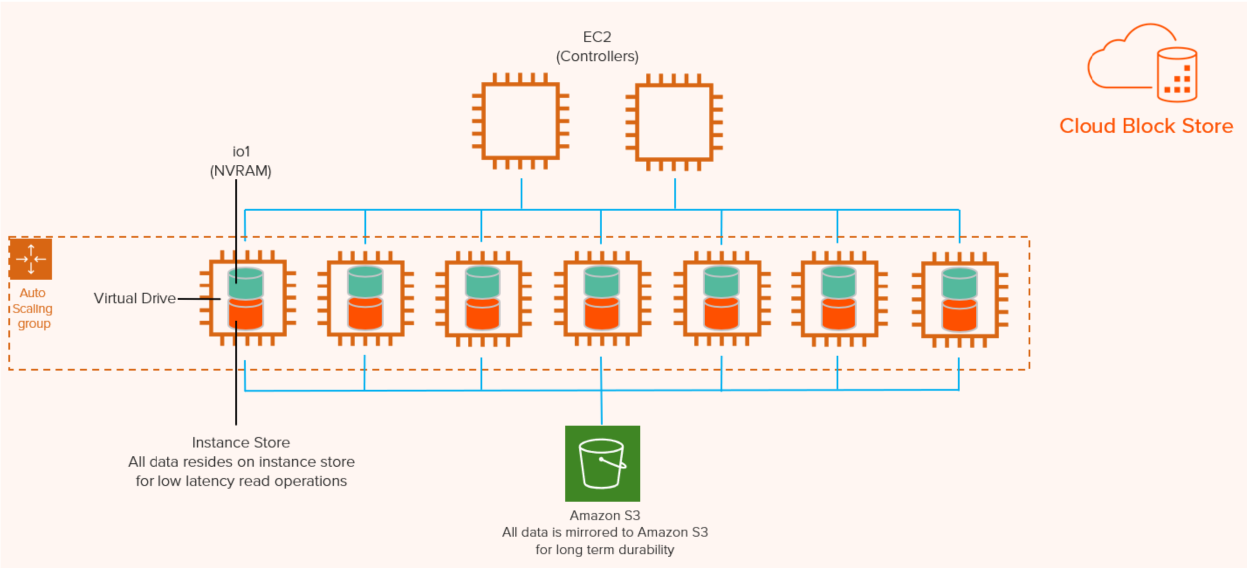


FIGURE 10. PURE STORAGE CLOUD BLOCK STORE (*CBS*)

Learn more about *Pure Storage Cloud Block Store.*

Pure Storage array-based replication can be used for migrating data vVols via async replication from an on-premises FlashArray to CBS on AWS. The replica data vVol(s) can then be presented to an existing VM on VMware Cloud on AWS via in-guest iSCSI using the Pure CBS service over the AWS ENI.

Learn more about *vVol migration and presentation using Pure CBS.*

Oracle Database Architecture

Oracle Database 19c, the latest generation of the world's most popular database, provides businesses of all sizes with access to the world's fastest, most scalable and reliable database technology for secure and cost-effective deployment of transactional and analytical workloads in the cloud, on-premises and hybrid cloud configurations.

An Oracle database server consists of a database and at least one database instance. In Oracle RAC, an Oracle database will have more than one instance accessing the database.

- A database is a set of files, located on disk, that store data. These files can exist independently of a database instance.
- An instance is a set of memory structures that manage database files. The instance consists of a shared memory area, called the system global area (SGA), and a set of background processes. An instance can exist independently of database files.

The physical database structures that comprise a database are:

- **Data files** – Every Oracle database has one or more physical data files, which contain all database data. The data of logical database structures, such as tables and indexes, is physically stored in the data files.
- **Control files** – Every Oracle database has a control file. A control file contains metadata specifying the physical structure of the database, including the database name, along with the names and locations of the database files.
- **Online redo log files** – Every Oracle database has an online redo log, representing a set of two or more online redo log files. An online redo log is made up of redo entries (also called redo log records), which record all changes made to data.
- Many other files, including parameter files, archived redo files, backup files and networking files, are important to any oracle database operation.

Learn more about *Oracle database architecture.*

Oracle ASM, ASMLIB and ASMFD

ASM

Oracle Automatic Storage Management (ASM) is a volume manager and a file system for Oracle database files that supports single-instance Oracle Database and Oracle RAC configurations.

Oracle ASM is Oracle's recommended storage-management solution that can be used for both Oracle RAC and single-instance Oracle databases and provides an alternative to conventional volume managers, file systems, and raw devices.

Oracle ASM uses disk groups to store data files. An Oracle ASM disk group is a collection of disks that Oracle ASM manages as a unit. Users can add or remove disks from a disk group while a database continues to access files from the disk group.

Learn more about *Oracle Automatic Storage management (ASM).*

ASMLIB

Oracle ASMLIB maintains permissions and disk labels that are persistent on the storage device, so that the label is available even after an operating system upgrade.

The Oracle ASMLIB driver simplifies the configuration and management of block disk devices by eliminating the need to rebind block disk devices used with Oracle Automatic Storage Management (ASM) each time the system is restarted.

Learn more about *Oracle ASMLIB.*

ASMFD

Oracle ASMFD helps prevent corruption in Oracle ASM disks and files within the disk group. Oracle ASMFD simplifies the configuration and management of disk devices by eliminating the need to rebind disk devices used with Oracle ASM each time the system is restarted.

Learn more about *Oracle ASMFD.*

Oracle Backup and Recovery

The purpose of backup and recovery is to protect the database against data loss and reconstruct the database after data loss. Oracle provides different options for database backup and recovery.

Oracle Recovery Manager (RMAN) is the most popular and preferred backup solution for Oracle database.

Common Oracle backup and recovery options include:

- User-managed database backup (hot and cold backup)
- Crash-consistent backup using storage-based snapshots
- Oracle RMAN
- Oracle Data Pump export/import

Learn more about *Oracle Backup and Recovery Solutions.*

Oracle User Managed Database Backup

The user-managed backup and recovery mechanism includes performing backup and recovery with a mixture of host operating system commands and SQL*Plus recovery commands. This strategy does not depend on using the Oracle RMAN.

A database-consistent backup is a whole database backup that can be opened with the **RESETLOGS** option without performing media recovery. It's not necessary to apply redo to this backup to make it consistent. Unless the redo generated is applied after the consistent backup is created, however, all transactions since the time of the consistent backup will be lost.

All datafiles in a consistent backup must:

- Have the same checkpoint system change number (SCN) in their headers, unless they are datafiles in tablespaces that are read-only or offline normal (in which case they will have a clean SCN that is earlier than the checkpoint SCN).
- Contain no changes past the checkpoint SCN (i.e., are not fuzzy).
- Match the data file checkpoint information stored in the control file.

See *Oracle Backup and Recovery User Guide* for more information.

Consistent backups can only be taken after a clean shutdown has been made or by turning on hot backup mode of the database. This is the most trusted backup by DBAs but is also complex, as the admin will need to run scripts to put the database in hot-backup mode, take a snapshot, and then take the database out of the hot-backup mode.

Oracle data pump backups are "logical" database backups in that they extract logical definitions and data from the database to a file.

With a cold backup, it's possible to make a consistent whole database backup of all files in a database after the database is shut down with the **NORMAL**, **IMMEDIATE**, or **TRANSACTIONAL** options.

See *Making User-Managed Backups of the Whole Database* for more information.

With a hot backup, this would require:

- Putting the tablespace/database (depending on whether it is a tablespace level or database level backup) in a **BEGIN** backup mode by the **ALTER TABLESPACE/DATABASE BEGIN BACKUP** command.
- Taking an operating system-level backup of the tablespace/database data files.
- Taking the tablespace/database out of the backup mode with the **ALTER TABLESPACE/ DATABASE END BACKUP** command.

There is overhead involved in transitioning a database in and out of backup mode:

- Additional redo data is logged
- Complete database checkpoint is required
- More operational steps and complexity during the backup operation

Oracle Crash-Consistent Backup

A crash-consistent backup is the backup of a point-in-time image of an Oracle database that is equivalent to a database crash induced by a power outage, other failures, or a shutdown abort.

When the database is started up, instance recovery (i.e., the process of applying records in the online redo log to data files to reconstruct changes) is performed automatically to bring the database to a consistent state.

This is one of the most common backup methods used for storage-based backups and is fully supported by Oracle as long as the following conditions are met.

As noted in *Supported Backup, Restore and Recovery Operations using Third Party Snapshot Technologies* (Oracle Doc ID 604683.1), third-party vendor snapshots must conform to the following requirements:

- Integrated with Oracle's recommended restore and recovery operations above
- Database crash-consistent at the point of the snapshot
- Write-ordering is preserved for each file within a snapshot

See *Making Backups with Third-Party Snapshot Technologies* for more information.

Oracle RMAN

Oracle RMAN is an Oracle Database client that performs backup and recovery tasks on databases and automates administration of backup strategies. It greatly simplifies backing up, restoring, and recovering database files.

The RMAN environment consists of the utilities and databases that play a role in backing up data. Minimally, the environment for RMAN must include the following components:

- **A target database** – An Oracle database to which RMAN is connected with the **TARGET** keyword. A target database is a database on which RMAN is performing backup and recovery operations. RMAN always maintains metadata about its operations on a database in the control file of the database. The RMAN metadata is known as the RMAN repository.
- **The RMAN client** – An Oracle database executable that interprets commands, directs server sessions to execute those commands, and records its activity in the target database control file. The RMAN executable is automatically installed with the database and is typically located in the same directory as the other database executables.

Advantages of Oracle RMAN-based backups include:

- Only used space in the database is backed up
- RMAN does not put tablespaces in backup mode, saving on redo generation overhead. RMAN will re-read database blocks until it gets a consistent image of it.

Learn more about *Oracle RMAN*.

Oracle Database Cloning

Cloning of an Oracle database is the process of making an exact copy of another database for various reasons. The cloned database is both fully functional and separate in its own right.

Use cases for cloning include making copies of the production database to use it:

- As a development database for developing new applications or adding new features to existing applications.
- As a QA database for testing existing software for bugs or testing new software features or versions.
- As a test database for backup and recovery scenarios.
- To provision a copy of a database for different business units.
- To test database patching, upgrade, and migration strategies.
- To benchmark for performance.

After cloning, the DBA may choose to mask sensitive data in the cloned database before releasing it for general consumption.

For example, a production database for a credit card company will have real customer data that cannot be revealed for security purposes, so Oracle data masking is used to mask customer names and social security number.

Examples of database cloning include using Oracle Enterprise Manager Cloud Control or classic cloning using RMAN backups. See *Cloning Oracle Databases and Pluggable Databases* for more information.

The database cloning process may also occasionally include making copies of Oracle database home directories, along with a copy of the Oracle database, for those instances when testing database patching, upgrade, or migration strategies is needed.

Oracle Database Refresh

Database refresh is typically the process of refreshing the contents of a development or testing database from a copy of the production database, ensuring the latest data is available to develop or test against.

Common use cases for database refresh include:

- Development and test environments may not have latest data to test the current application against.
- A production bug needs to be investigated in the test environment, but the data is not the latest.

Oracle provides tools for database refresh, including:

- Oracle RMAN
- Data Pump

Oracle Database Patching

Oracle DBAs are responsible for patching databases to apply system patches, including patch set update (PSU), security patch update (SPU) and critical patch update (CPU). The process for applying these patches can be tedious and time-consuming.

It's important that patches are applied carefully. Multiple factors must be taken into consideration when performing a patch, including communicating the expected outage window to application owners and ensuring a solid backup and restore mechanism is in place in the event of a roll back following a failed patching exercise. Ignoring these steps risks a full system restore being required, resulting in a lengthy outage and potential revenue loss.

Beginning with Oracle Database 11g Release 2 (11.2), Oracle ASM is installed as part of an Oracle Grid Infrastructure installation.

Patching single-instance databases using Grid Infrastructure requires patching the Grid Infrastructure binaries, the RDBMS binaries, or both, depending on the patch type. See *Introduction to OPatch and Patching* for more information.

When patching an Oracle RAC, Oracle offers flexibility, allowing a rolling patch upgrade in which RAC nodes are patched one by one. This avoids taking the entire cluster down for patching, ensuring SLAs are met and uptime is guaranteed.

See *Oracle Database Upgrade Guide* for more information.

See *My Oracle Support* document for more information on Rolling Patch—OPatch Support for RAC (Doc ID 244241.1).

Considerations for Oracle Database Day 2 Operations

The duration of a database backup is determined by the following:

- **Level of backup** – The duration of the backup varies based on the backup level.

  A full backup has to back up the entire database, while incremental or cumulative backups leverage and backup only things that have changed since the previous backup.
- **Data churn** – The amount of data that changes between backups impacts the duration of incremental and cumulative backups. The more churn, the longer the backups.
- **Backup mechanism** – The mechanism used for backups impacts the duration. Backups can be tape or disk-based and leverage compression and parallel streams. Some backups track block-level changes and back up only the blocks that have changed.

  The backup mechanism can critically affect the backup duration and potentially affect the databases being backed up.
- **Underlying infrastructure** – The network and storage infrastructure along with backup medium can affect the duration. If the backups happen over the network and the network is shared with applications, there can be an adverse effect on backup and database performance.

  SAN-based backups traditionally have minimal impact on the infrastructure.

Production database backup and restore operations must meet strict SLAs and requirements, including:

- Small backup windows to minimize the impact on the database.
- The need to be recoverable and repeatable.

While database storage continues to grow exponentially, there is a tremendous pressure to reduce and optimize backup windows.

Due to restricted backup windows and data churn, it is not feasible to make full backups of multi-terabyte databases in the allotted backup windows.

For example, it's possible to increase the number of RMAN channels if the production system has enough bandwidth for data transfer but adding more channels can introduce performance overhead into the production system.

Backups for large databases have traditionally been a challenge in virtual environments due to the additional layer of abstraction they introduce. Special provisions, such as dedicated RDMs and storage-level snapshots, must be leveraged for efficient backups.

Cloning multi-terabyte databases and refreshing multi-terabyte databases present the same challenges outlined above.

Oracle Database Day 2 Operations and vVols

Generally speaking, regular database operations (e.g., database backup, cloning, or data refresh) can be triggered in three ways:

- At the application (Oracle Database) level
- At the VMware VM level
- At the storage level

Each of these approaches offers advantages and drawbacks:

- **Application (Oracle Database) level** – Leveraging database backup applications such as Oracle RMAN or Oracle Data Pump to backup or clone databases.

  It provides a finer level of granularity (i.e., tablespace, datafile, or table level granularity) but it is not always the fastest.
- **VMware VM level** – Leverages VMware snapshots, the way by which most VMs are backed up, including third-party backup applications.

  VM-level granularity is ideal, however a VM-level snapshot may stun a VM for some time during snapshot coalescing and deletion. See VMware *KB 1002836* for more information.

  This may not be acceptable for tier-one business-critical databases. Removal of snapshots may impact the performance of the database as the VMDKs are consolidated.
- **Storage Array level** – Operations at the storage level offer better performance but lack VM-level granularity, as operations are executed at storage LUN level.

  If the database to be backed up contains dedicated LUNs in the SAN, it can potentially be backed up at the SAN level with a storage-level snapshot.

  In the event the database is sharing the datastore/LUN with other databases, the database can still be backed up using storage-level backup, however it will include VMDKs of the other databases as well.

For large databases, DBAs typically prefer triggering the above database operations at either the application level or storage level for the reasons outlined above.

Combining VMDK-level granularity with the speed of storage array database operations like backup and recovery, cloning, or provisioning a database VM provides an ideal solution to the challenges outlined above. This approach enables:

- Triggering of backups and clones with VMDK granularity simultaneously.
- Performing a storage-level snapshot or clone at the VM level (the fastest of the three options available).

This approach should also allow:

- Operations like backup and restore and cloning to flexibly operate at VMDK-level granularity.
- Rapid storage-level snapshotting or cloning, with operations triggered at the VM level.
- Alignment of different database components with different storage data services needed for SPBM.

vVols storage management and automation framework addresses the following requirements:

- vVols eliminates the need for a native file system. vVols objects are natively represented on the storage array, mitigating any performance concerns and potentially eliminating debate between DBAs and vSphere administrators regarding the appropriate approach.
- Backups can be triggered at a VM level and at VMDK-level granularity, with the actual operation run natively by the array.
- Regular DBA tasks such as database backup and restore, cloning, and database refresh of non-production databases from production takes less time.
- Because vVols snapshots are array-based, taking a snapshot of the machine is much faster and less disruptive to the VM. Snapshots do not incur a performance penalty, regardless of the number of snapshots taken.

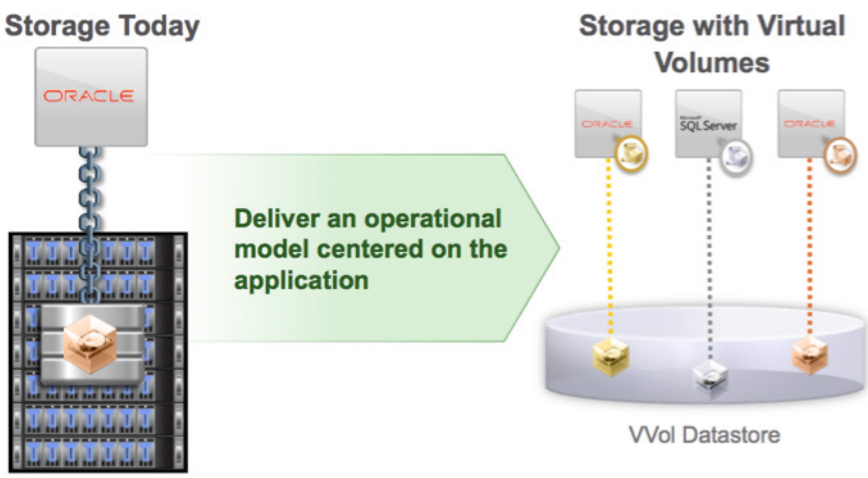


FIGURE 11. APPLICATION STORAGE WITH VVOLS

Oracle Real Application Cluster (RAC)

Oracle Clusterware is portable cluster software that provides comprehensive multi-tiered high availability and resource management for consolidated environments. It supports clustering of independent servers so that they cooperate as a single system.

Oracle Clusterware is the integrated foundation for Oracle Real Application Clusters (Oracle RAC), and the high-availability and resource management framework for all applications on any major platform.

More information on *Oracle Clusterware 19c* can be found here.

Non-cluster Oracle Database instances have a one-to-one relationship between Oracle Database and the instance. Oracle RAC environments, however, have a one-to-many relationship between the database and instances. An Oracle RAC database can have several instances, all of which access one Oracle database. All database instances must use the same interconnect, which can also be used by Oracle Clusterware.

The combined processing power of the multiple servers can provide greater throughput and Oracle RAC scalability than is available from a single server.

A cluster comprises multiple interconnected computers or servers that appear as if they are one server to end users and applications. The Oracle RAC option with Oracle Database enables users to cluster Oracle Database instances. Oracle RAC uses Oracle Clusterware for the infrastructure to bind multiple servers, so they operate as a single system.

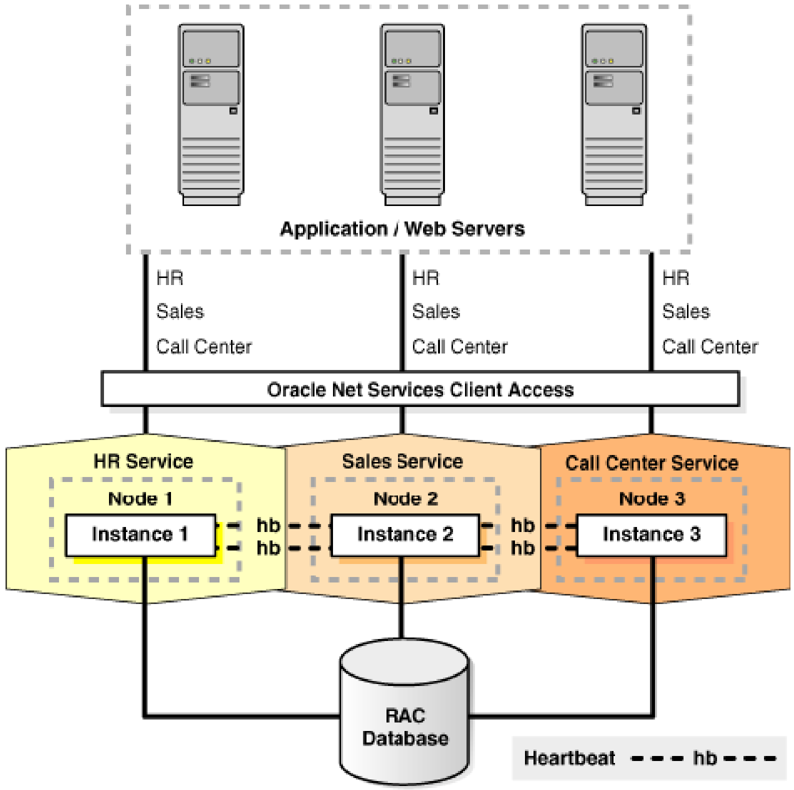


FIGURE 12. *ORACLE RAC*

Learn more about *Oracle RAC 19c.*

Extended Oracle RAC

An Oracle Extended Cluster consists of nodes that are in multiple locations called sites.

When deploying an Oracle Standalone Cluster, administrators can also choose to configure the cluster as an Oracle Extended Cluster. An Oracle RAC cluster can be extended across two, or more, geographically separate sites, each equipped with its own storage. In the event that one of the sites fails, the other site acts as an active standby.

Oracle RAC on extended distance (stretched) clusters provides extremely fast recovery from a site failure and allows for all nodes, in all sites, to actively process transactions as part of a single database cluster

Both Oracle ASM and the Oracle Database stack, in general, are designed to use enterprise-class shared storage in a datacenter. Fibre Channel technology, however, enables distribution of compute and storage resources across two or more datacenters, connecting them through Ethernet cables and Fibre Channel, for compute and storage needs, respectively.

The high impact of latency, and therefore distance, creates some practical limitations as to where this architecture can be deployed. An active/active Oracle RAC architecture fits best where the two datacenters are located relatively close (<100km) and where the costs of setting up a low latency and dedicated direct connectivity between the sites for Oracle RAC has already taken place, which is why it cannot be used as a replacement for a disaster recovery solution such as Oracle Data Guard or Oracle GoldenGate.

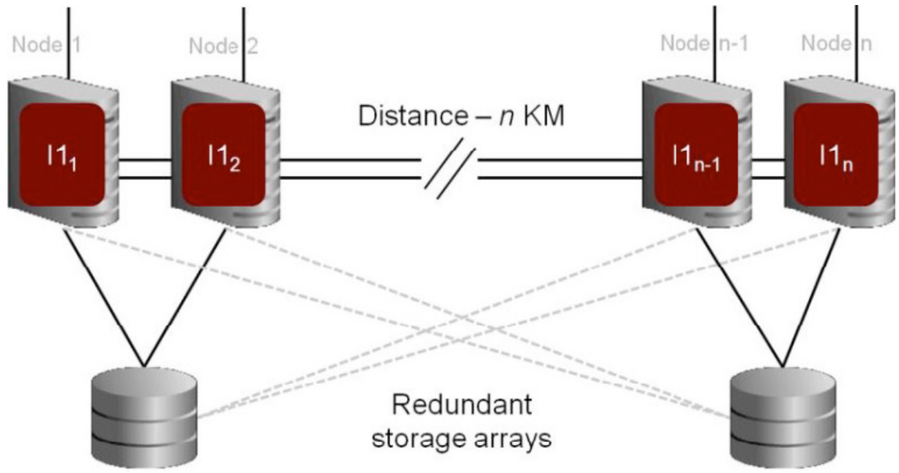


FIGURE 13. *ORACLE RAC ON EXTENDED DISTANCE CLUSTERS CHARACTERISTICS*

More information about Extended Oracle RAC 19c can be found *here* and *here*.

Oracle RAC on VMware vSphere

There are two key requirements for Oracle RAC:

- Shared storage
- Multicast Layer 2 networking

These requirements are fully addressed when running Oracle RAC on VMware vSphere, as both shared storage and Layer 2 networking are natively supported by vSphere.

vSphere high availability (HA) clusters enable a collection of ESXi hosts to work together so that, as a group, they provide higher levels of infrastructure level availability for VMs than each ESXi host can provide individually.

vSphere HA provides high availability for VMs by pooling the VMs and the hosts they reside on into a cluster. Hosts in the cluster are monitored and in the event of a failure, the VMs on a failed host are restarted on alternate hosts.

When creating a vSphere HA cluster, a single host is automatically elected as the master host. The master host communicates with vCenter Server and monitors the state of all protected VMs and of the slave hosts.

Learn more about *VMware vSphere HA.*

Oracle RAC and VMware HA solutions are completely complementary to each other. Running Oracle RAC on a VMware platform provides the application-level HA enabled by Oracle RAC, in addition to the infrastructure-level HA enabled by VMware vSphere.

Learn more about *Oracle RAC on VMware vSphere.*

Oracle RAC on VMware Virtual Volumes (vVols)

The steps for deploying an Oracle RAC, whether traditional or extended, on VMware vSphere are essentially the same across all VMware vSphere platforms.

Beginning with VMware vSphere 6.5, vVols 2.0 and VASA 3.0, vVols are now validated to support Oracle RAC workloads delivering policy-based, VM-centric storage for Oracle RAC clusters.

Learn more by reading *What's New in Virtual Volumes (vVols) 2.0.*

The RAC shared storage provisioning for vVols can be summarized in the table below:

VMware Platform	Datastore	Version	RAC shared VMDK requirement for multi-writer attribute	Reference
VMware vSphere	vVol	Beginning with ESXi 6.5	Vendor specific	*KB 1034165* and *KB 2113013*

TABLE 5. ORACLE RAC SHARED STORAGE FOR VVOLS

The RAC private interconnect networking setup for VMware on-premises is shown as below:

VMware Platform	Distributed Switch	Distributed Port Group	Version	Reference
VMware vSphere, vSAN, vVols	Regular Distributed Switch	Dedicated Distributed Port Group for RAC	Beginning with 5.5	*https://docs.vmware.com/en/VMware-vSphere/7.0/com.vmware.vsphere.networking.doc/GUID-D21B3241-0AC9-437C-80B1-0C8043CC1D7D.html*

TABLE 6. ORACLE RAC PRIVATE INTERCONNECT NETWORKING

The previous details provide guidance for Oracle RAC storage deployment on vVols using Pure Storage, Network deployment details, along with the basic steps for RAC deployment on VMware vVols using Pure Storage, can be found in *Oracle VMware Hybrid Cloud High Availability Guide REFERENCE ARCHITECTURE.*

Extended Oracle RAC on Pure ActiveCluster vMSC

Many business-critical applications require five 9s of availability, or 99.999% availability (less than 5 minutes of downtime per year). This is where the marriage of vSphere HA and Oracle RAC really shines. This combination has been used to great effect by several very large organizations globally.

As mentioned above, Oracle RAC and VMware HA solutions are completely complementary to each other. Running Oracle RAC on the VMware platform provides application-level HA enabled by Oracle RAC, in addition to the infrastructure-level HA enabled by vSphere.

Extended Oracle RAC provides greater availability than local Oracle RAC. It provides extremely fast recovery from a site failure and enables all servers, in all sites, to actively process transactions as part of a single database cluster.

Running Extended Oracle RAC on a stretched cluster architecture provides the same advantages as traditional Oracle RAC across datacenters and sites, in addition to the site-level protection enabled by stretched cluster architecture.

Both traditional Oracle RAC and extended Oracle RAC require:

- Layer 2 network adjacency
- Shared storage

In addition to these requirements, it's important to keep the following in mind when deploying an extended Oracle RAC:

- Latency requirements of the workload, for which considerations include:
  - Site distance (0, 25, 50, 100, > 100KM)
  - Maintaining network latency at 5ms (milliseconds) or less, as latency between RAC nodes and across sites increases
  - Network connection and bandwidth between sites. Dark Fiber over Dense Wavelength Division Multiplexing (DWDM)
- Storage that is synchronously, bi-directionally replicated in 5ms or less is required at each site
- A witness or quorum site is required for any clustered app or storage to avoid a "split brain"

There are two general implementations of VMware stretched cluster architectures:

- VMware vSphere Metro Storage Cluster (vMSC)
- VMware vSAN Stretched Cluster

Learn more about *vMSC.*

Learn more about *Extended Oracle RAC on vMSC.*

More information about Pure ActiveCluster VMware vMSC can be found *here* **and** *here*.

Running Extended Oracle RAC on Pure ActiveCluster VMware vMSC is beyond the scope of this document.

Solution Configuration

This section introduces the resources and configurations for the solution, including:

- Architecture diagram
- Hardware resources
- Software resources
- Network configuration
- Storage configuration
- Pure Storage Plugin for VMware vSphere client
- VM and Oracle configuration

Architecture Diagram

This solution architecture relies on a three-site scenario:

- On-premises vSphere cluster on Site A (Santa Clara)
- On-premises vSphere cluster on Site B (Wenatchee)
- VMware Cloud on AWS

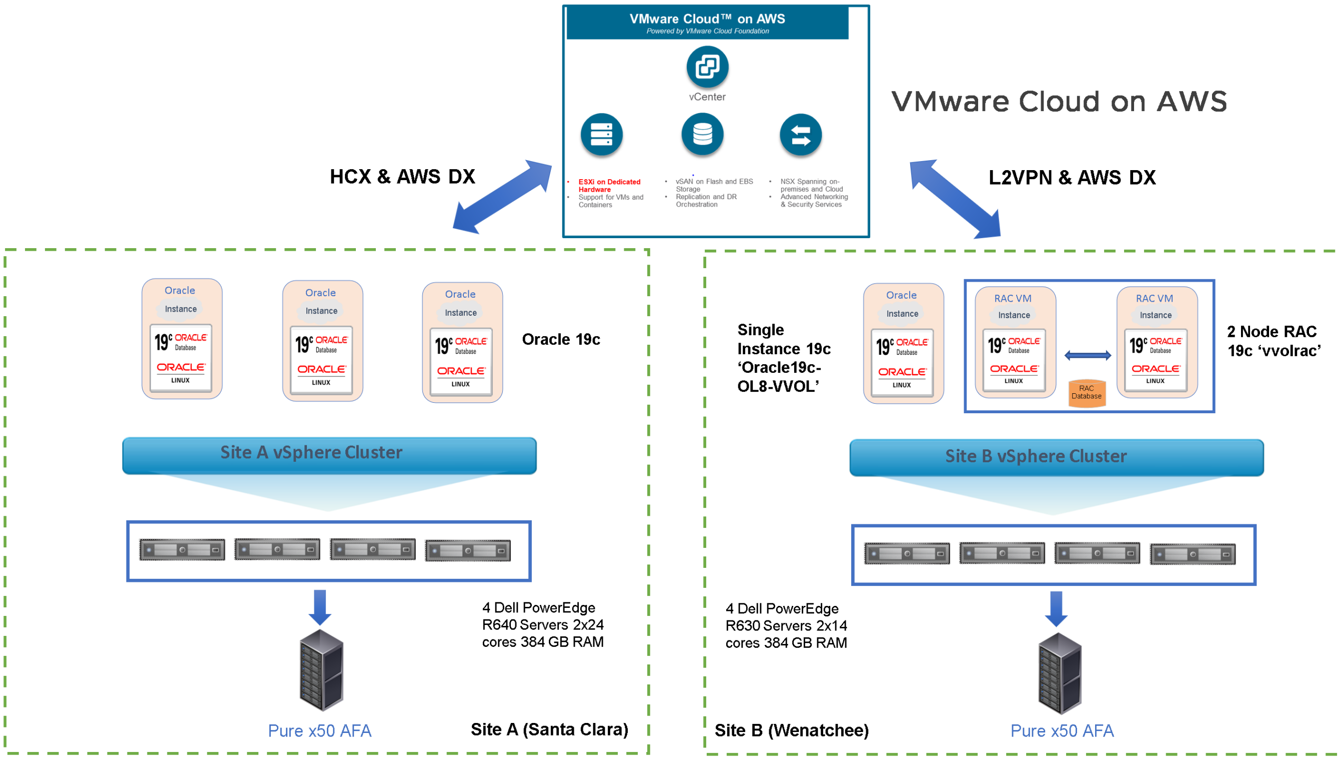


FIGURE 14. SITE ARCHITECTURE DIAGRAM

The on-premises setup features two separate and dedicated vSphere cluster configurations: Site A and Site B.

- Site A is hosting non-production RAC and non-RAC workloads, including disaster recover (DR)
- Site B is hosting production RAC and production single-instance workloads
- Both sites are connected to VMware Cloud on AWS

This solution focuses on vVols and will examine the on-premises Site B configurations for traditional Oracle single-instance and Oracle RAC setup.

Site B infrastructure details are as follows:

- A 4-node vSphere VMware ESXi, 7.0.1, 16850804
- Each ESXi server is a Dell PowerEdge R630 Server with Intel Xeon CPU E5-2680 v4 at 2.40GHz, 2x14 cores, and 384GB RAM with hyperthreading
- Each ESXi server has access to a Pure Storage FlashArray//x50 (Purity/FA 5.3.10) for both block FC storage and vVols
- Each ESXi server features:
  - 2 x QLogic 8Gb FC host bus adapters for FC storage
  - 2 x 10Gb connections

Hardware Resources

Below are the hardware resources for the vSphere cluster on Site B:

Description	Specification
Server	4 x ESXi Server
Server Model	Dell Inc. PowerEdge R630
CPU	2 sockets with 14 cores each, Intel(R) Xeon(R) CPU E5-2680 v4 @ 2.40GHz, 2x14 cores, with Hyperthreading enabled
RAM	384GB RAM
Storage controller	2 x 8Gb ISP2532-based Fibre Channel to PCI Express HBA
Storage Array	Pure x50 AFA (Purity/FA 5.3.10)
vSAN Storage Controller	Dell HBA330 Mini
Storage Network	2 x QLogic 8Gb FC Host Bus Adapters for FC Storage
Network	2 x 10Gbit connections

TABLE 7. SITE B HARDWARE RESOURCES

The following summarizes one of the ESXi servers in the vSphere cluster on site B:

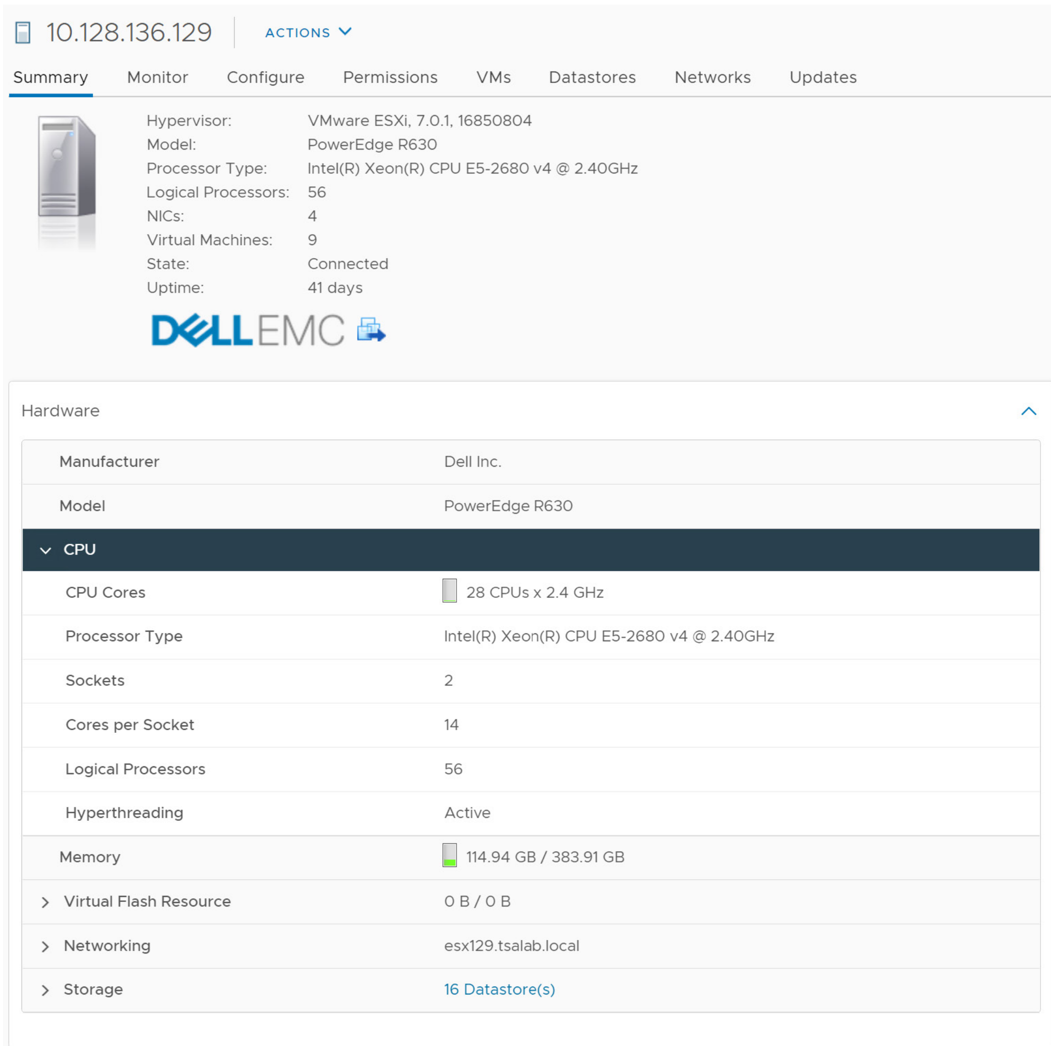


FIGURE 15. SITE B VMWARE ESXI SERVER SUMMARY

Software Resources

The following is a summary of the software resources used:

Software	Version	Purpose
VMware vCenter Server	7.0.1 Build 16858589	VMware vCenter Server provides a centralized platform for managing VMware vSphere environments
VMware ESXi Server	7.0.1 Build 16850804	ESXi servers to host VMs
ESXi Datastores	Purity//FA 5.3.10	Pure AFA provides both VMFS and vVol datastores
Oracle Linux	8.3 UEK	Oracle database server nodes
Oracle Database 19c	19.8.0.0.0	Oracle database

TABLE 8. SOFTWARE RESOURCES

Network Configuration

A VMware vSphere Distributed Switch (VDS) acts as a single virtual switch across all associated hosts in the datacenter. This setup enables VMs to maintain a consistent network configuration as they migrate across multiple hosts.

A port group defines properties regarding security, traffic-shaping, and network adapter-teaming. Jumbo frames (MTU=9000 bytes) are enabled on the VMware vSphere® vMotion® interface and the default port group setting is used.

For Site B, VDS **TSA-WDC-70B-vDS01** uses 2x 10GbE adapter and 2 x 1GbE adapter per host:

- 2 x 10GbE uplinks for VM traffic
- 2 x 1GbE uplinks for VMkernel non-VM traffic

The following distributed switch-port groups were created for Oracle RAC and Oracle VM traffic to balance traffic across the available uplinks:

- Port group **TSA-WDC-70B-PG1403** with VLAN ID 1403 is for VM user traffic
- Port group **TSA-WDC-70B-PG70-NFS** with VLAN ID 70 for NFS traffic
- Port group **TSA-WDC-70B-PG71-OraPrivate** with VLAN ID 71 is for Oracle RAC interconnect traffic with two active/active uplinks set to **Route based on originating virtual port**.

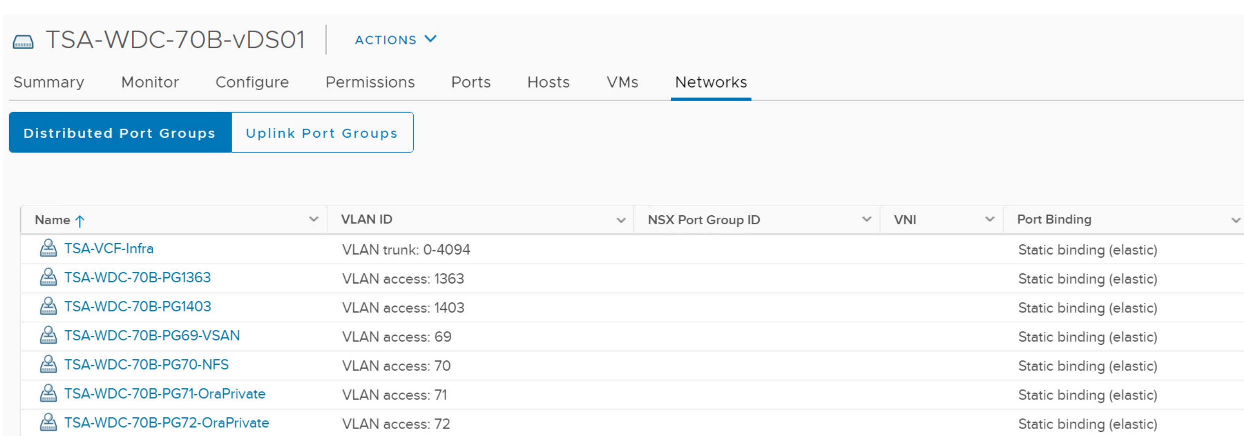
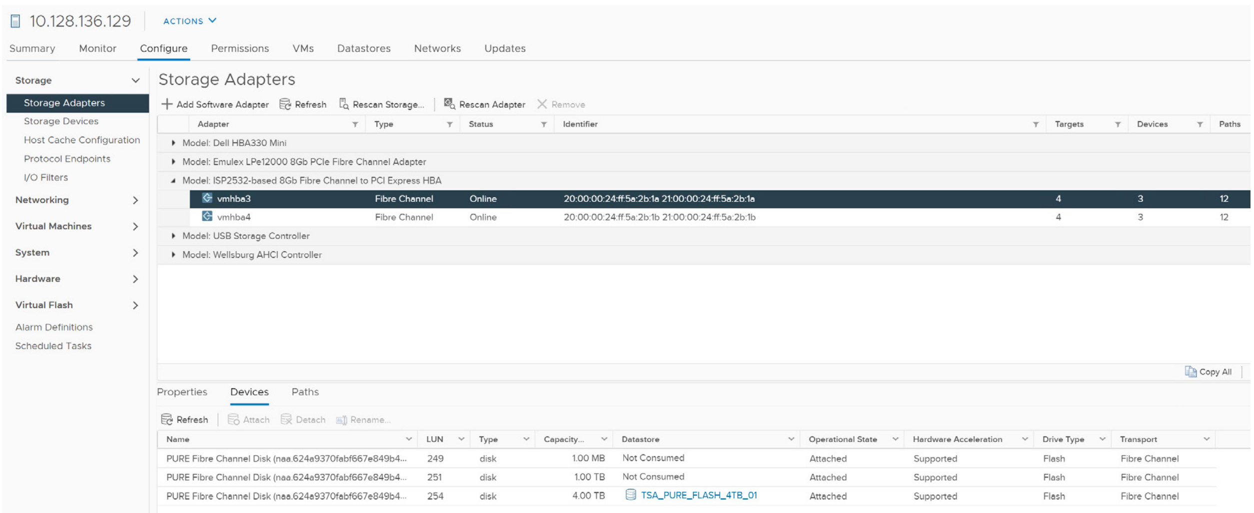


FIGURE 16. SITE B VSPHERE DISTRIBUTED SWITCH PORT GROUP CONFIGURATION

Storage Configuration

Site B has access to a Pure Storage FlashArray//x50 all-flash storage (Purity/FA 5.3.2) for vVols.

On Site B, each of the 4 ESXi servers contains 2 x QLogic 8Gb FC host bus adapters for vVols.



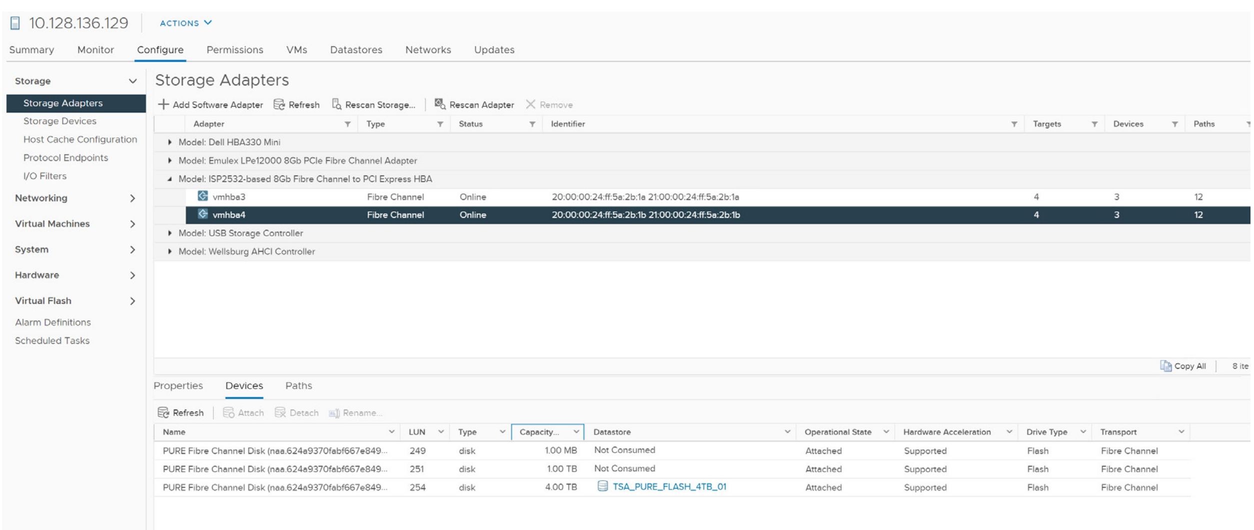


FIGURE 17. ESXI SERVER FC STORAGE CONNECTIONS

On Site B, on the 4-node vSphere cluster, a Pure x50 storage-backed vVols datastore (TSA_PURE_FLASH_VVOL) was created.

The graphic below outlines the vVols datastores using Pure Storage in Site B:

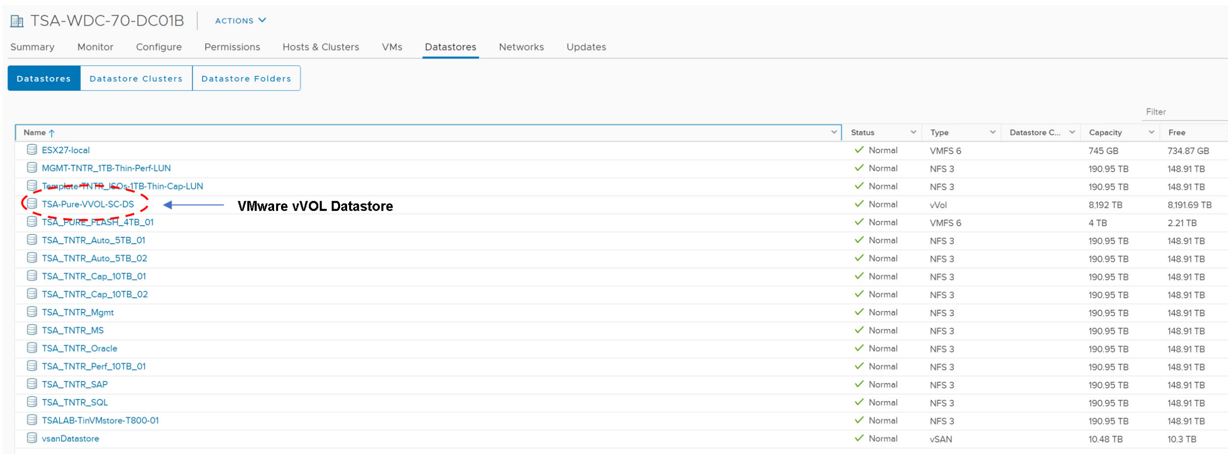


FIGURE 18. SITE B VVOLS DATASTORES

Pure Storage Plugin for VMware vSphere Client

The Pure Storage Plugin for the vSphere client enables VMware users to have insight into, and control of, their Pure Storage FlashArray environment while directly logged into the vSphere client.

The Pure Storage Plugin extends the vSphere client interface to include environmental statistics and objects that underpin the VMware objects in use and to provision new resources as needed.

Learn more about *installing the Pure Storage Plugin for the vSphere client.*

Pure Storage Plugin details are shown below:

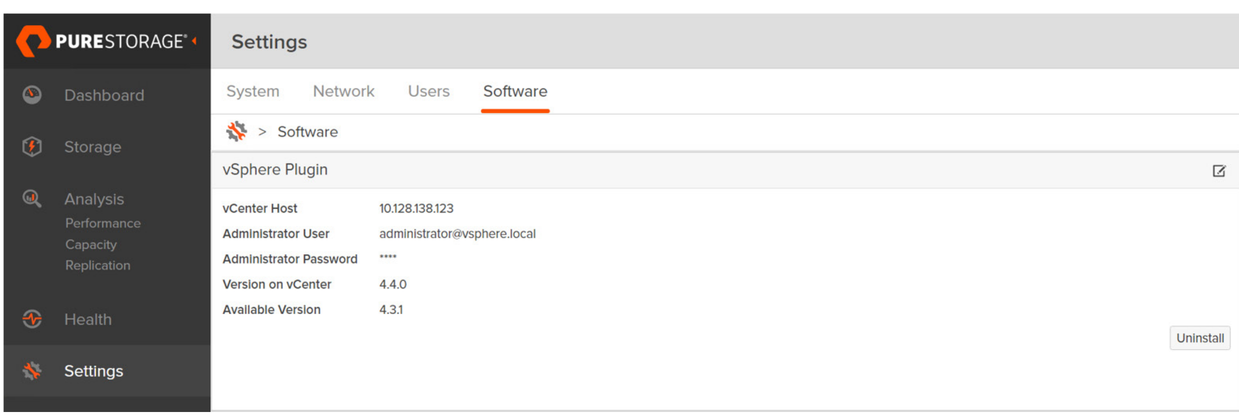


FIGURE 19. PURE STORAGE PLUGIN DETAILS

VMware vCenter and Pure Storage Plugin:

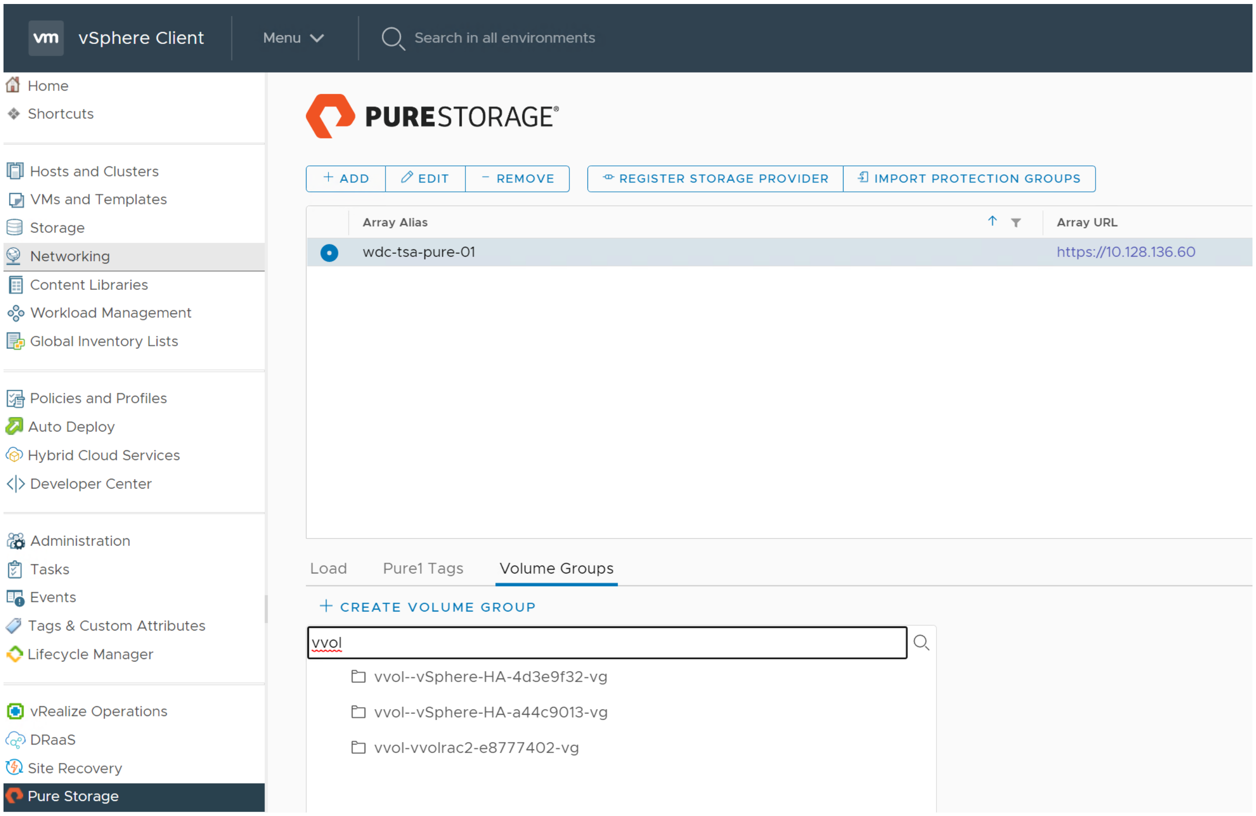


FIGURE 20. VMWARE VCENTER AND PURE STORAGE PLUGIN

Once the Plugin is installed, from the VM **Oracle19c-OL8-VVOL** view and summary tab, there is a FlashArray widget box indicating whether or not the VM has undelete protection. Undelete protection means that there is currently a FlashArray snapshot of the VM's config-vVol.

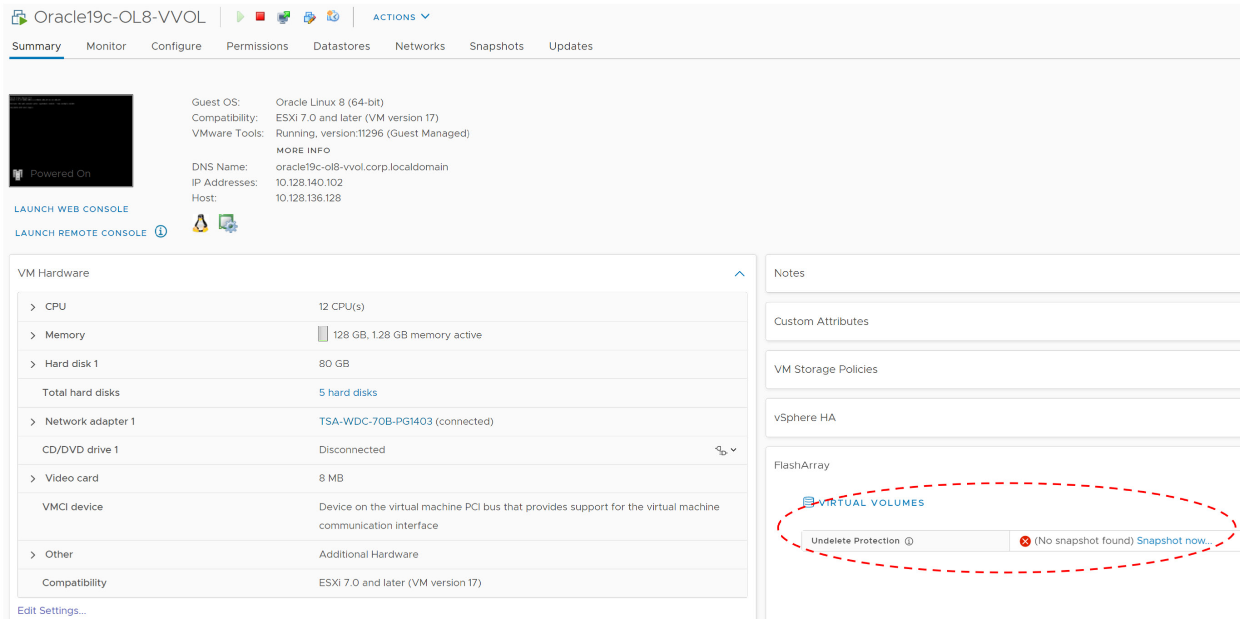


FIGURE 21. UNDELETE PROTECTION WIDGET

Navigate to VM **Oracle19c-OL8-VVOL**'s Configure tab to see vVols on Pure Storage.

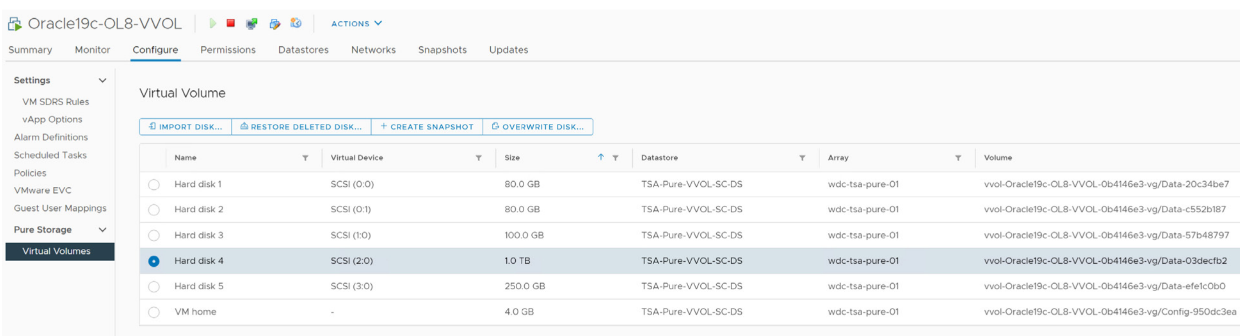


FIGURE 22. VVOLS ON PURE STORAGE

The Pure Storage Plugin enables the following operations:

- **Import Disk** – to import a virtual disk (vVol)
- **Restore Deleted Disk** – to restore a destroyed vVol
- **Create Snapshot** – to take a snapshot
- **Overwrite Disk** – to overwrite an existing vVol

Learn more about *Pure Storage Plugin operations.*

Virtual Machine and Oracle Configuration

Three VMs were created as follows:

- VM Oracle19c-OL8-VVOL
- VM Oracle19c-OL8-VVOL-QAVM
- VM Oracle19c-OL8-VVOL-RMAN

Each VM was created with the following tools or characteristics:

- VM version 17 on ESXi 7.0
- Guest operating system Oracle Enterprise Linux 8.3 UEK
- Oracle Grid and RDBMS binaries version 19.8
- ASM disk group for Oracle Grid Infrastructure Management Repository (GIMR) named **MGMT_DATA**
- Different names for DATA and FRA ASM disks on VMs **Oracle19c-OL8-VVOL** and **Oracle19c-OL8-RMAN**
  - On VM **Oracle19c-OL8-VVOL**, ASM diskgroup DATA_DG contains a ASM disk **DATA_01** and ASM diskgroup FRA_DG has a ASM disk **FRA_01**
  - On VM Oracle19c-OL8-RMAN, ASM diskgroup RMAN_DATA_DG contains a ASM disk **RMAN_DATA_01**
- Storage for all three VMs was provisioned on the vVols datastore **TSA-Pure-VVOL-SC-DS**

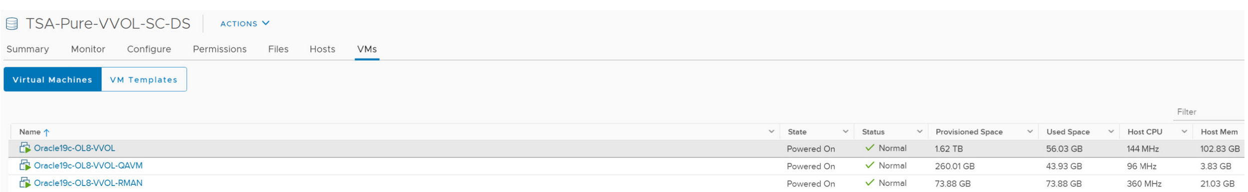


FIGURE 23. ORACLE CONFIGURED VIRTUAL MACHINES

Details for VM **Oracle19c-OL8-VVOL** are as follows:

- 12 vCPUs with 128GB RAM
- Oracle SGA set to 96GB with traditional HugePages and PGA set to 6GB
- VM hosts both Oracle Grid and RDBMS 19.8 multi-tenant production database **vvol19c** with a pluggable database **pdb1**
- 3 ASM disks groups:
  - MGMT_DATA for Oracle Grid Infrastructure Management Repository (GIMR) with ASM disk **MGMT_DATA01**
  - DATA_DG for data and redo log files with ASM disk **DATA_01**
  - FRA_DG for archive logs files with ASM disk **FRA_01**

All Oracle on VMware platform best practices were followed as per the *VMware Hybrid Cloud Best Practices Guide for Oracle Workloads.*

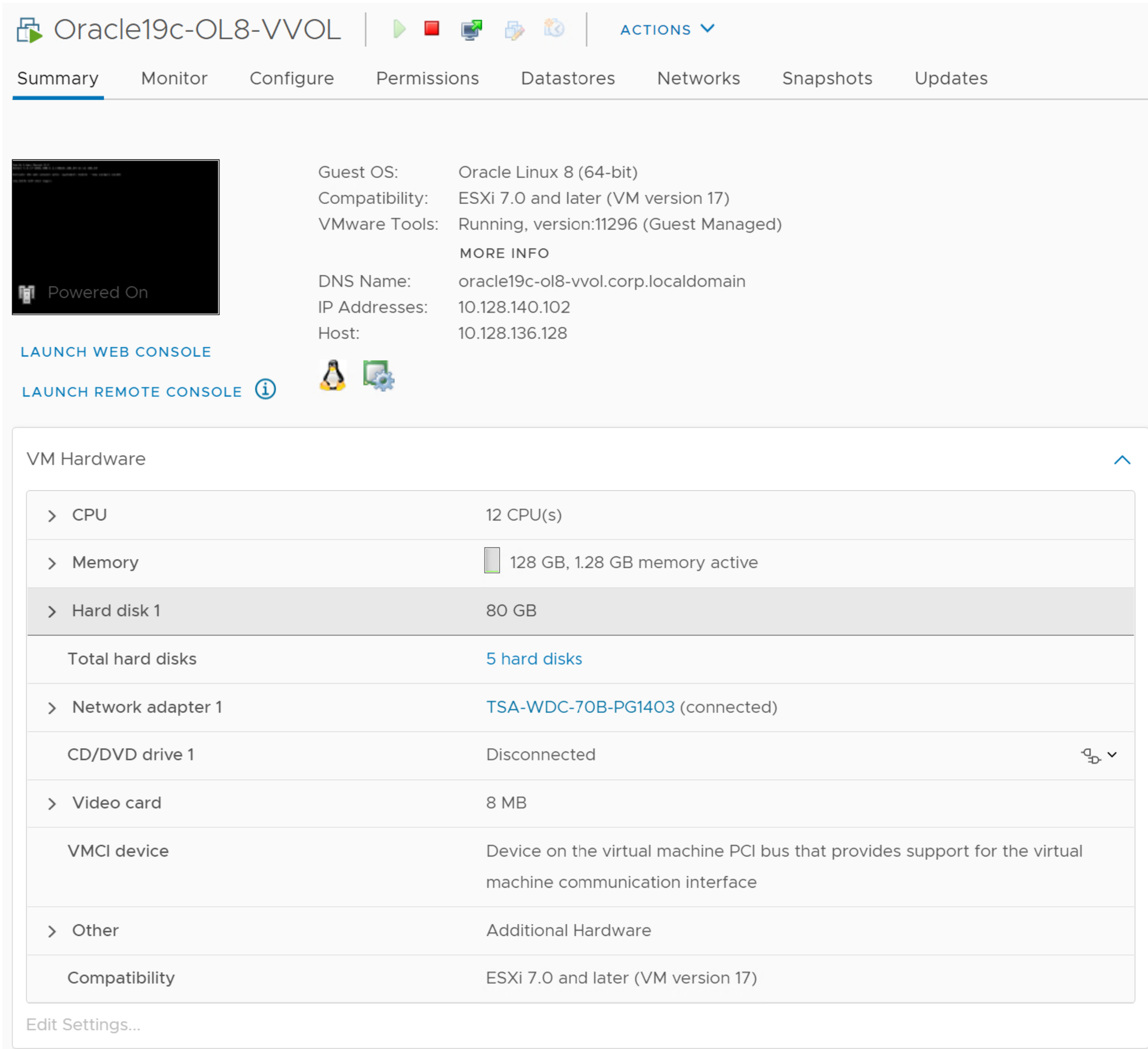


FIGURE 24. VM ORACLE19C-OL8-VVOL SUMMARY

VM **Oracle19c-OL8-VVOL** virtual disks (VMDKs) are shown below. All SCSI controllers are set to **VMware Paravirtual SCSI Controller** type.

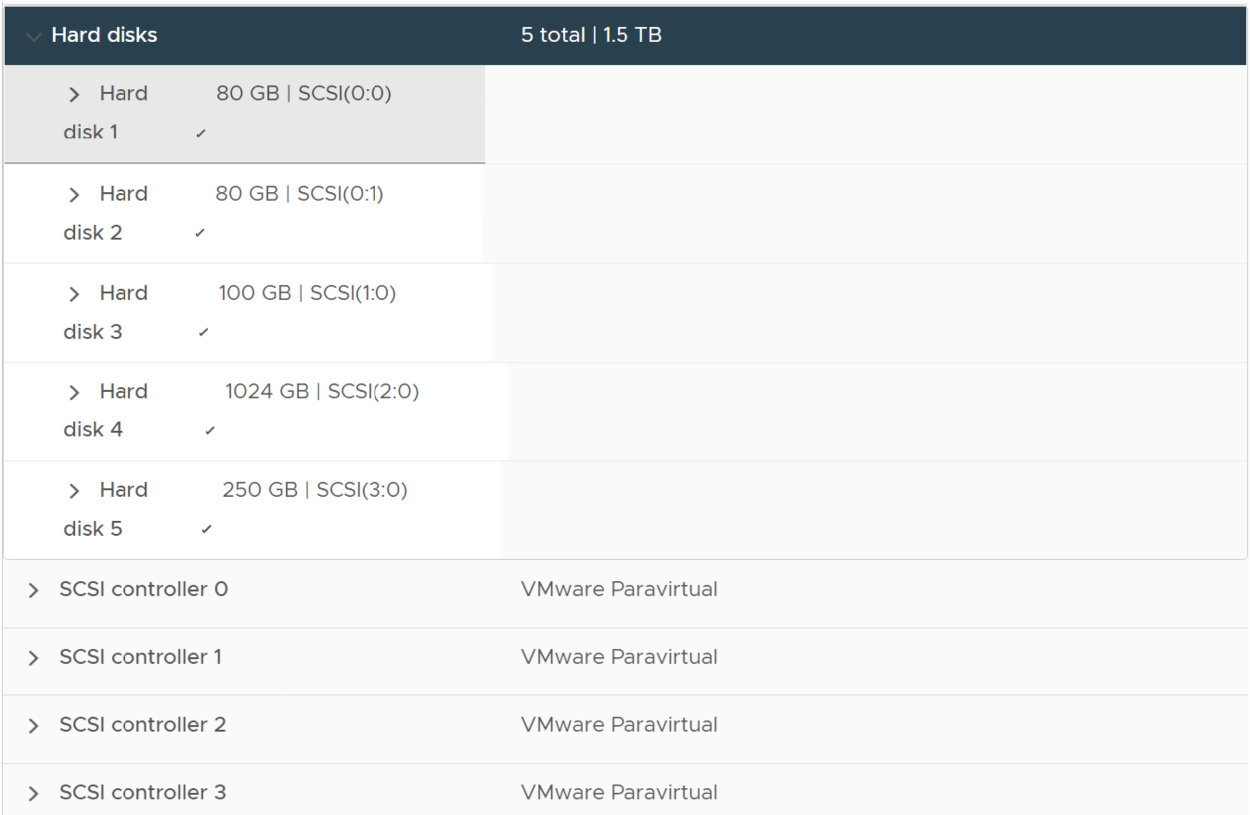
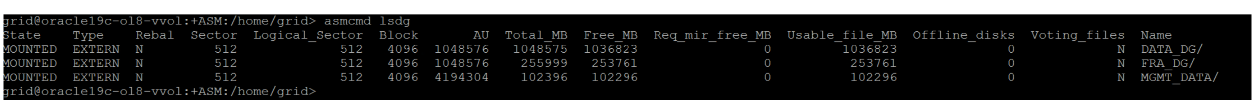


FIGURE 25. VM ORACLE19C-OL8-VVOL VMDKS

Virtual disk (VMDK) details:

- Hard Disk 1 – 80GB for operating system
- Hard Disk 2 – 80GB for Oracle Grid and RDBMS binaries
- Hard Disk 3 – 100GB for Oracle Grid Infrastructure Management Repository (GIMR) (Management Database (MGMTDB)) (ASM Disk Group MGMT_DATA)
- Hard Disk 4 – 1TB for database **vvol19c** data and redo log files (ASM Disk Group DATA_DG)
- Hard Disk 5 – 250GB for database **vvol19c** archive logs files (ASM Disk Group FRA_DG)

Oracle ASM disk group details:



```
grid@oracle19c-ol8-vvol:+ASM:/home/grid> asmcmd lsdg
State    Type    Rebal  Sector  Logical_Sector  Block       AU  Total_MB  Free_MB  Req_mir_free_MB  Usable_file_MB  Offline_disks  Voting_files  Name
MOUNTED  EXTERN  N         512             512   4096  1048576   1048575  1036823                0         1036823              0             N  DATA_DG/
MOUNTED  EXTERN  N         512             512   4096  1048576    255999   253761                0          253761              0             N  FRA_DG/
MOUNTED  EXTERN  N         512             512   4096  4194304    102396   102296                0          102296              0             N  MGMT_DATA/
grid@oracle19c-ol8-vvol:+ASM:/home/grid>
```

FIGURE 26. ORACLE ASM DISK GROUP

Hard Disk 4 (1TB) details are shown below:

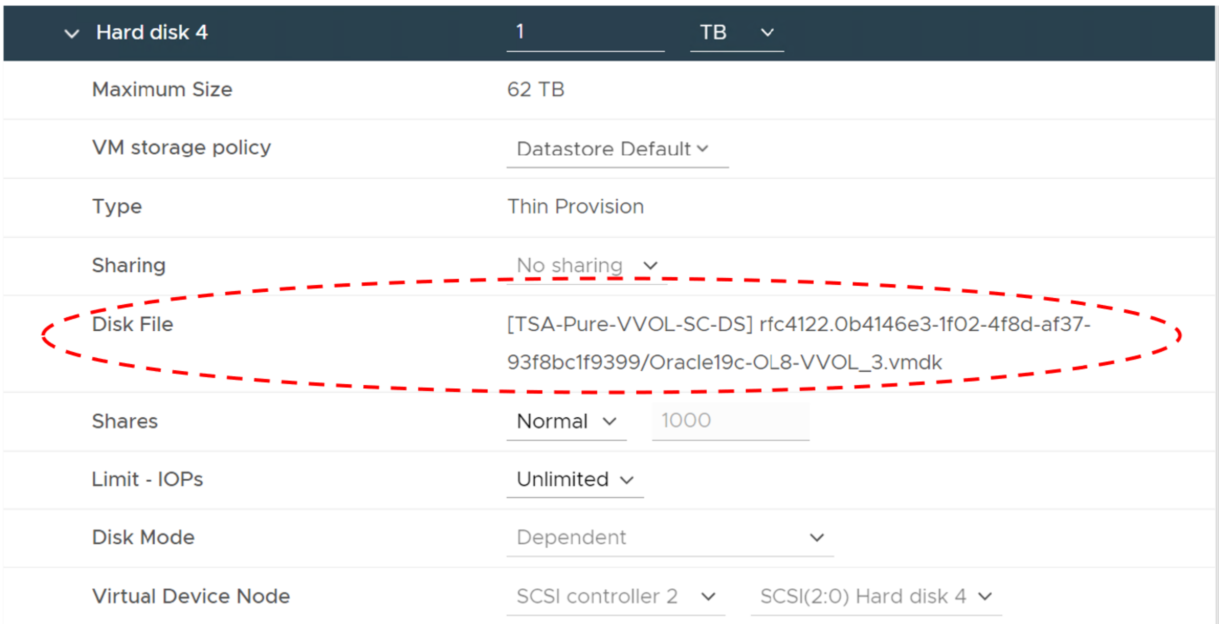


FIGURE 27. HARD DISK 4 (1TB)

Additional vVols-backed VMDKs details are shown below:

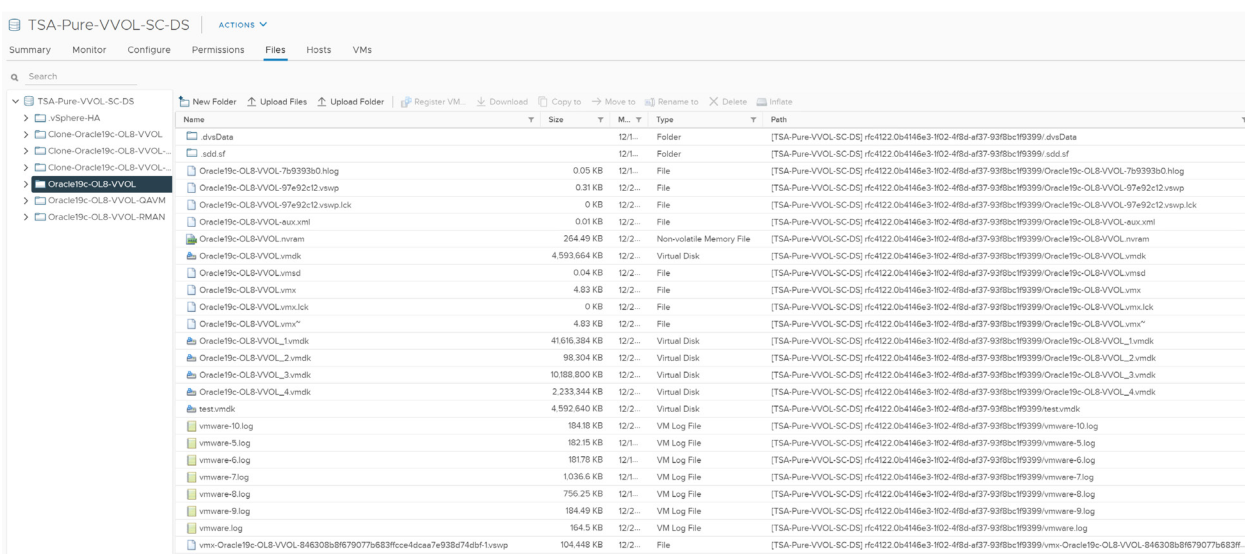


FIGURE 28. VVOLS-BACKED VMDKS

The subtlety lies in the way a vVols datastore type is implemented compared to a regular VMFS datastore.

A **df** command on the ESXi server will reveal that vVols datastores are implemented as a datastore of type **vvol**.

```
[root@esx129:~] df
Filesystem            Bytes           Used         Available Use% Mounted on
NFS         209949611261952 46224916664320  163724694597632  22% /vmfs/volumes/Template-TNTR_ISOs-1TB-Thin-Cap-LUN
NFS         209949611261952 46224916664320  163724694597632  22% /vmfs/volumes/TSA_TNTR_SQL
NFS         209949611261952 46224916664320  163724694597632  22% /vmfs/volumes/TSA_TNTR_SAP
NFS         209949611261952 46224916664320  163724694597632  22% /vmfs/volumes/TSA_TNTR_Auto_5TB_01
NFS         209949611261952 46224916664320  163724694597632  22% /vmfs/volumes/TSALAB-TinVMstore-T800-01
NFS         209949611261952 46224916664320  163724694597632  22% /vmfs/volumes/TSA_TNTR_MS
NFS         209949611261952 46224916664320  163724694597632  22% /vmfs/volumes/TSA_TNTR_Auto_5TB_02
NFS         209949611261952 46224916664320  163724694597632  22% /vmfs/volumes/TSA_TNTR_Cap_10TB_02
NFS         209949611261952 46224916664320  163724694597632  22% /vmfs/volumes/MGMT-TNTR_1TB-Thin-Perf-LUN
NFS         209949611261952 46224916664320  163724694597632  22% /vmfs/volumes/TSA_TNTR_Cap_10TB_01
NFS         209949611261952 46224916664320  163724694597632  22% /vmfs/volumes/TSA_TNTR_Mgmt
NFS         209949611261952 46224916664320  163724694597632  22% /vmfs/volumes/TSA_TNTR_Oracle
NFS         209949611261952 46224916664320  163724694597632  22% /vmfs/volumes/TSA_TNTR_Perf_10TB_01
VMFS-6        4397778075648  1968705634304    2429072441344  45% /vmfs/volumes/TSA_PURE_FLASH_4TB_01
VMFS-L          13690208256     1719664640      11970543616  13% /vmfs/volumes/LOCKER-5f862770-9cd1a2c9-119c-246e965377bc
vfat             1073577984      183894016        889683968  17% /vmfs/volumes/BOOTBANK2
vfat             1073577984      174227456        899350528  16% /vmfs/volumes/BOOTBANK1
vvol       9007199254740992   341226553344 9006858028187648   0% /vmfs/volumes/TSA-Pure-VVOL-SC-DS
vsan         11522248015872   197623776542   11324624239330   2% /vmfs/volumes/vsanDatastore
[root@esx129:~]
```

FIGURE 29. DF COMMAND RESULT

Unlike any regular VM files in the VM folder on a datastore where we can see the different –flat.vmdk files, with VMs on vVols, the folder contents appear as follows:

```
[root@esx129:/vmfs/volumes/vvol:52cc3c9a587e393e-a20e901faf033899/rfc4122.0b4146e3-1f02-4f8d-af37-93f8bc1f9399] pwd
/vmfs/volumes/TSA-Pure-VVOL-SC-DS/Oracle19c-OL8-VVOL
[root@esx129:/vmfs/volumes/vvol:52cc3c9a587e393e-a20e901faf033899/rfc4122.0b4146e3-1f02-4f8d-af37-93f8bc1f9399] ls -l
total 113680
-rw-r--r--    1 root     root            53 Dec 15 16:20 Oracle19c-OL8-VVOL-7b9393b0.hlog
-rw-------    1 root     root           315 Dec 29 01:25 Oracle19c-OL8-VVOL-97e92c12.vswp
-rw-------    1 root     root             0 Dec 29 01:25 Oracle19c-OL8-VVOL-97e92c12.vswp.lck
-rw-r--r--    1 root     root            13 Dec 28 22:25 Oracle19c-OL8-VVOL-aux.xml
-rw-------    1 root     root        270840 Dec 29 01:25 Oracle19c-OL8-VVOL.nvram
-rw-------    1 root     root           672 Dec 29 01:25 Oracle19c-OL8-VVOL.vmdk
-rw-r--r--    1 root     root            43 Dec 28 22:25 Oracle19c-OL8-VVOL.vmsd
-rwxr-xr-x    1 root     root          4948 Dec 29 01:25 Oracle19c-OL8-VVOL.vmx
-rw-------    1 root     root             0 Dec 29 01:25 Oracle19c-OL8-VVOL.vmx.lck
-rwxr-xr-x    1 root     root          4947 Dec 29 01:25 Oracle19c-OL8-VVOL.vmx~
-rw-------    1 root     root           673 Dec 29 01:25 Oracle19c-OL8-VVOL_1.vmdk
-rw-------    1 root     root           617 Dec 29 01:25 Oracle19c-OL8-VVOL_2.vmdk
-rw-------    1 root     root           621 Dec 29 01:25 Oracle19c-OL8-VVOL_3.vmdk
-rw-------    1 root     root           618 Dec 29 01:25 Oracle19c-OL8-VVOL_4.vmdk
-rw-------    1 root     root           633 Dec 28 19:57 test.vmdk
-rw-r--r--    1 root     root        188605 Dec 28 19:57 vmware-10.log
-rw-r--r--    1 root     root        186517 Dec 16 00:51 vmware-5.log
-rw-r--r--    1 root     root        186143 Dec 16 03:03 vmware-6.log
-rw-r--r--    1 root     root       1061478 Dec 18 18:18 vmware-7.log
-rw-r--r--    1 root     root        774398 Dec 18 20:20 vmware-8.log
-rw-r--r--    1 root     root        188913 Dec 21 16:22 vmware-9.log
-rw-r--r--    1 root     root        168449 Dec 29 20:27 vmware.log
-rw-------    1 root     root     106954752 Dec 29 01:25 vmx-Oracle19c-OL8-VVOL-846308b8f679077b683ffcce4dcaa7e938d74dbf-1.vswp
[root@esx129:/vmfs/volumes/vvol:52cc3c9a587e393e-a20e901faf033899/rfc4122.0b4146e3-1f02-4f8d-af37-93f8bc1f9399]
```

FIGURE 30. VM FOLDER ON VVOLS

Notice the lack of –flat.vmdk files as they exist on the vVols storage.

Logging into the Pure x50 storage GUI shows the various vVols which represent the different virtual disks of VM **Oracle19c-OL8-VVOL.**

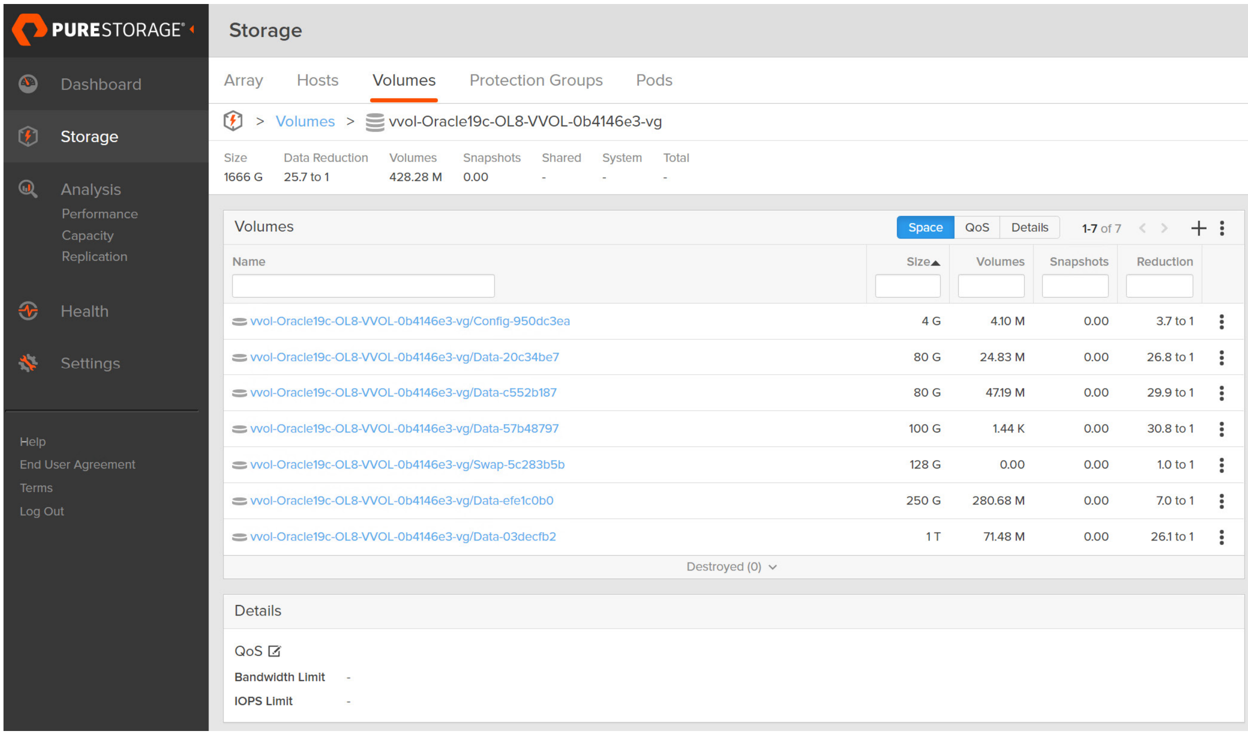


FIGURE 31. PURE STORAGE VOLUMES OF VM ORACLE19C-OL8-VVOL

Logging into Pure Storage via Putty and running the following command also produces the information provided above:

```
pureuser@wdc-tsa-pure-01> purevgroup list
Name                                        Volumes
vvol-Oracle19c-OL8-VVOL-0b4146e3-vg         vvol-Oracle19c-OL8-VVOL-0b4146e3-vg/Config-950dc3ea
                                            vvol-Oracle19c-OL8-VVOL-0b4146e3-vg/Data-03decfb2
                                            vvol-Oracle19c-OL8-VVOL-0b4146e3-vg/Data-20c34be7
                                            vvol-Oracle19c-OL8-VVOL-0b4146e3-vg/Data-57b48797
                                            vvol-Oracle19c-OL8-VVOL-0b4146e3-vg/Data-c552b187
                                            vvol-Oracle19c-OL8-VVOL-0b4146e3-vg/Data-efe1c0b0
                                            vvol-Oracle19c-OL8-VVOL-0b4146e3-vg/Swap-5c283b5b
pureuser@wdc-tsa-pure-01>

pureuser@wdc-tsa-pure-01> purevol list
Name                                                Size  Source                                                   Created                  Serial
vvol-Oracle19c-OL8-VVOL-0b4146e3-vg/Config-950dc3ea 4G    -                                                        2020-12-14 20:13:01 PST  FABF667E849B44C500013526
vvol-Oracle19c-OL8-VVOL-0b4146e3-vg/Data-03decfb2   1T    -                                                        2020-12-15 14:37:54 PST  FABF667E849B44C50001353C
vvol-Oracle19c-OL8-VVOL-0b4146e3-vg/Data-57b48797   100G  -                                                        2020-12-15 08:32:16 PST  FABF667E849B44C50001352E
vvol-Oracle19c-OL8-VVOL-0b4146e3-vg/Data-20c34be7   80G   -                                                        2020-12-14 20:13:02 PST  FABF667E849B44C500013527
vvol-Oracle19c-OL8-VVOL-0b4146e3-vg/Data-c552b187   80G   -                                                        2020-12-14 20:14:18 PST  FABF667E849B44C500013528
vvol-Oracle19c-OL8-VVOL-0b4146e3-vg/Data-efe1c0b0   250G  -                                                        2020-12-18 11:56:35 PST  FABF667E849B44C5000135A7
vvol-Oracle19c-OL8-VVOL-0b4146e3-vg/Swap-5c283b5b   128G  -                                                        2020-12-29 13:29:22 PST  FABF667E849B44C5000139E9
pureuser@wdc-tsa-pure-01>
```

FIGURE 32. PURE STORAGE VOLUMES OF VM ORACLE19C-OL8-VVOL

It's important to note that when following an SSH to the ESXi server and navigating to the vVol datastore mount to check the VM **Oracle19c-OL8-VVOL** folder, there are no **XXXX-flat.vmdk**. We only see the .vmdk files.

The VMDK file contents point to the actual vVol, which backs the **XXXX-flat.vmdk.**

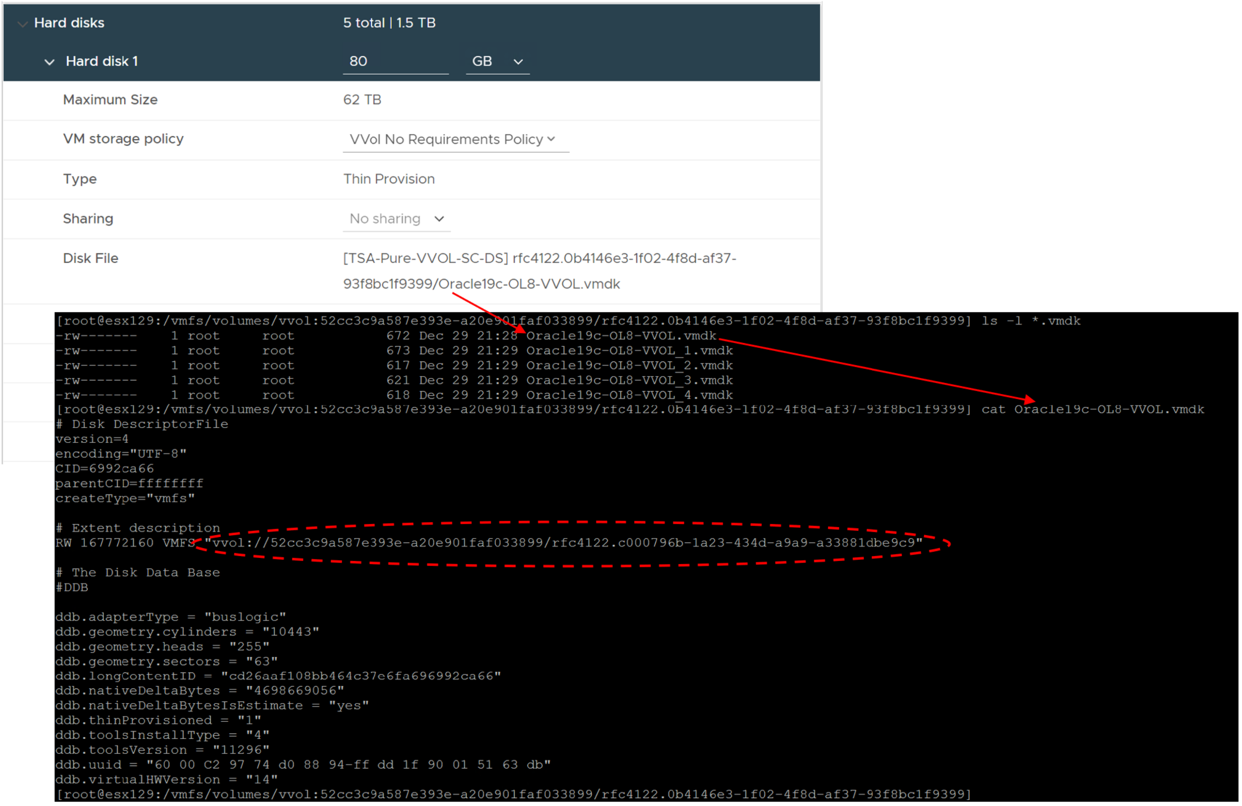


FIGURE 33. ORACLE VM VMDK FILE

Details of VM **Oracle19c-OL8-QAVM** are as follows:

- 8 vCPUs with 96GB RAM
- VM hosts both Oracle Grid and RDBMS 19.8 binaries only for staging and mount purposes
- 1 ASM disk group
  - MGMT_DATA for Oracle Grid Infrastructure Management Repository (GIMR) with ASM disk **MGMT_DATA01**

All Oracle on VMware platform best practices were followed as outlined in *VMware Hybrid Cloud Best Practices Guide for Oracle Workloads.*

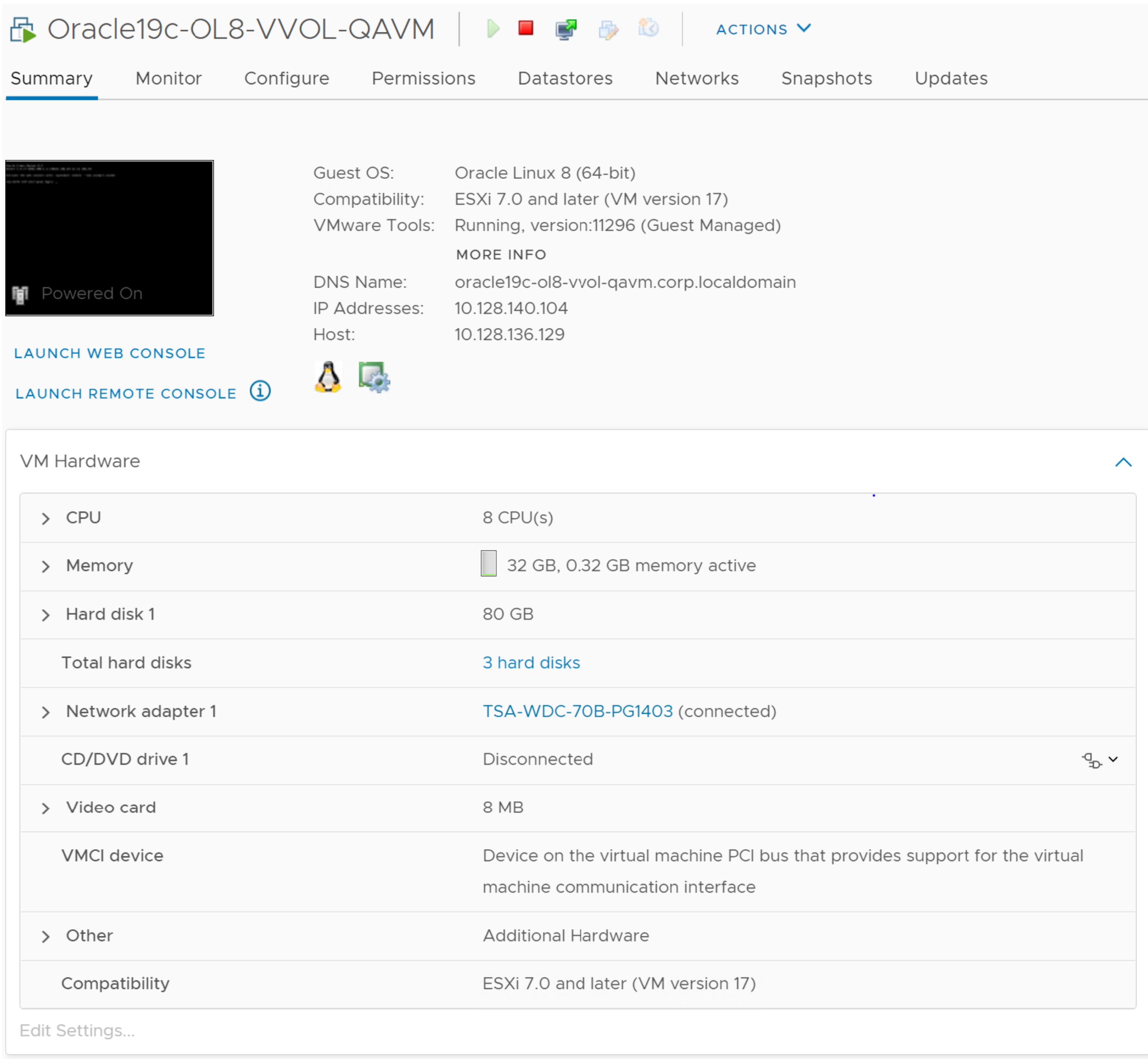


FIGURE 34. VM ORACLE19C-OL8-VVOL-QAVM SUMMARY

VM **Oracle19c-OL8-VVOL-QAVM** virtual disks (VMDKs) are shown below. All SCSI controllers are set to **VMware Paravirtual SCSI Controller type.**

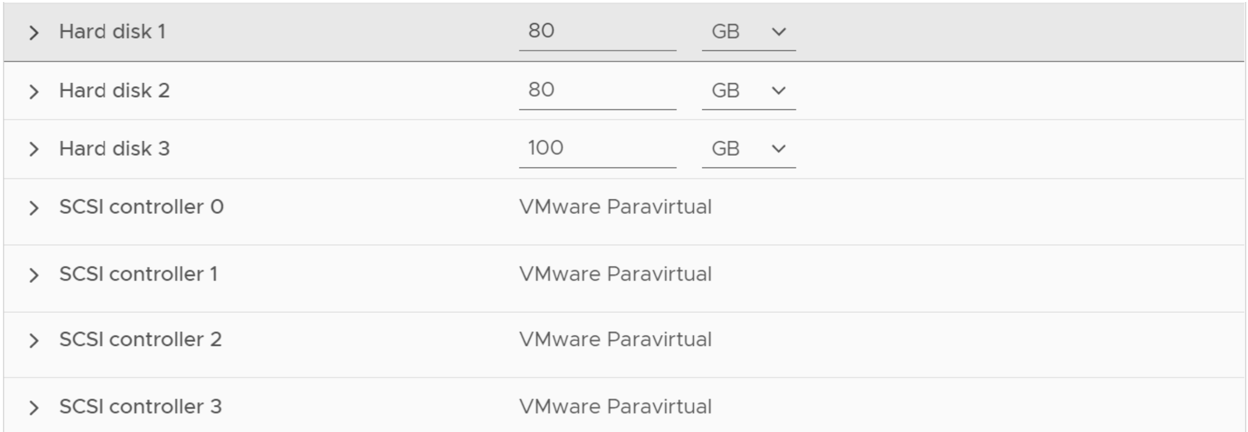


FIGURE 35. VM ORACLE19C-OL8-VVOL-QAVM VMDKS

Virtual disk (VMDKs) details:

- Hard Disk 1 – 80GB for operating system
- Hard Disk 2 – 80GB for Oracle Grid and RDBMS binaries
- Hard Disk 3 – 100GB for Oracle Grid Infrastructure Management Repository (GIMR) (Management Database (MGMTDB)) (ASM Disk Group MGMT_DATA)

Additional vVols-backed VMDK details are shown below:

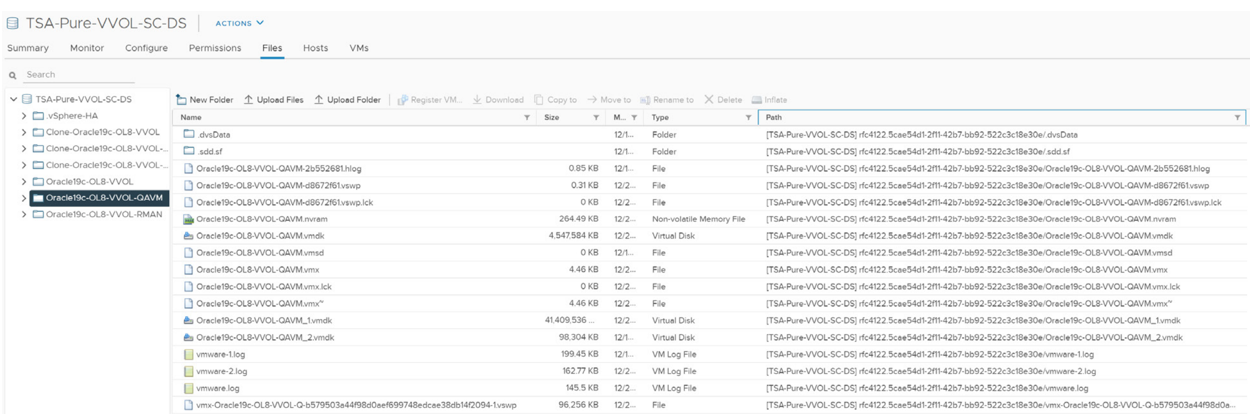


FIGURE 36. VVOLS-BACKED VMDKS

Logging into the Pure x50 storage GUI shows the various vVols representing the different virtual disks of VM **Oracle19c-OL8-VVOL-QAVM.**

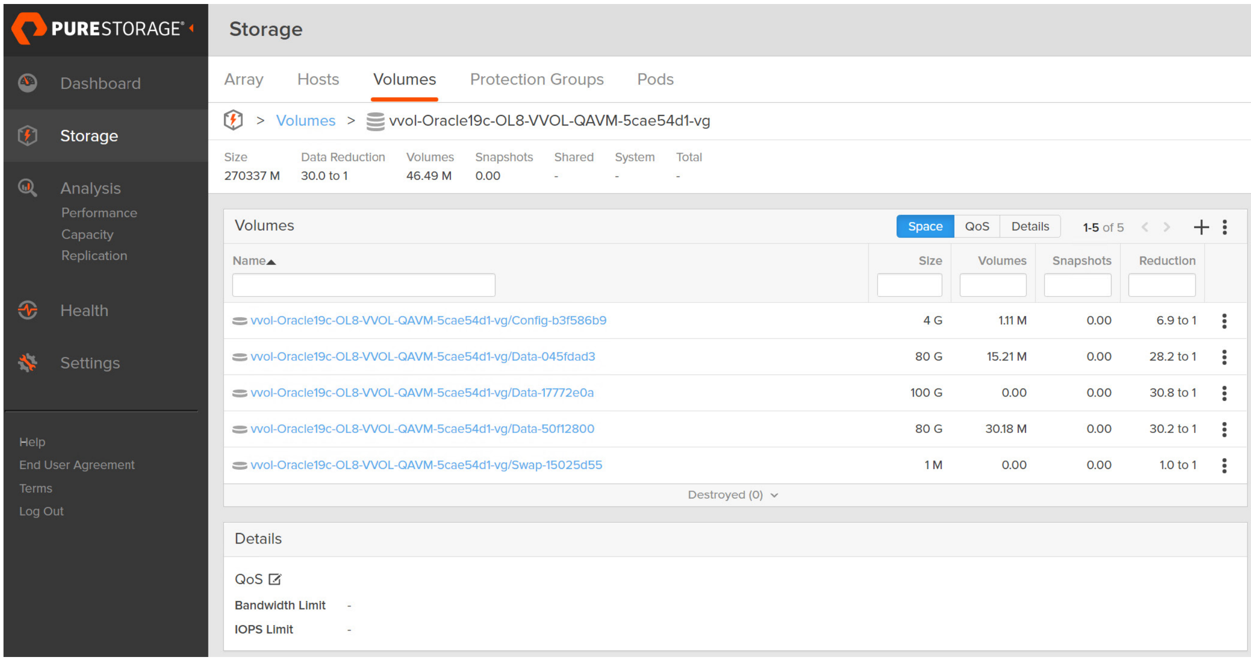


FIGURE 37. PURE STORAGE VOLUMES OF VM ORACLE19C-OL8-VVOL-QAVM

VM **Oracle19c-OL8-RMAN** details are as follows:

- 8 vCPUs with 96GB RAM
- Oracle SGA set to 16GB with traditional HugePages and PGA set to 6GB
- VM hosts both Oracle Grid and RDBMS 19.8 multi-tenant production database **rmandb** with a pluggable database **pdb1** for Oracle RMAN catalog purpose and a xfs file system **/rman** for holding Oracle RMAN backups
- 2 ASM disks groups
  - MGMT_DATA for Oracle Grid Infrastructure Management Repository (GIMR) with ASM disk **MGMT_DATA01**
  - RMAN_DATA_DG for data, redo log files and archive log files with ASM disk **RMAN_DATA_01**

All Oracle on VMware platform best practices were followed as outlined in *VMware Hybrid Cloud Best Practices Guide for Oracle Workloads.*

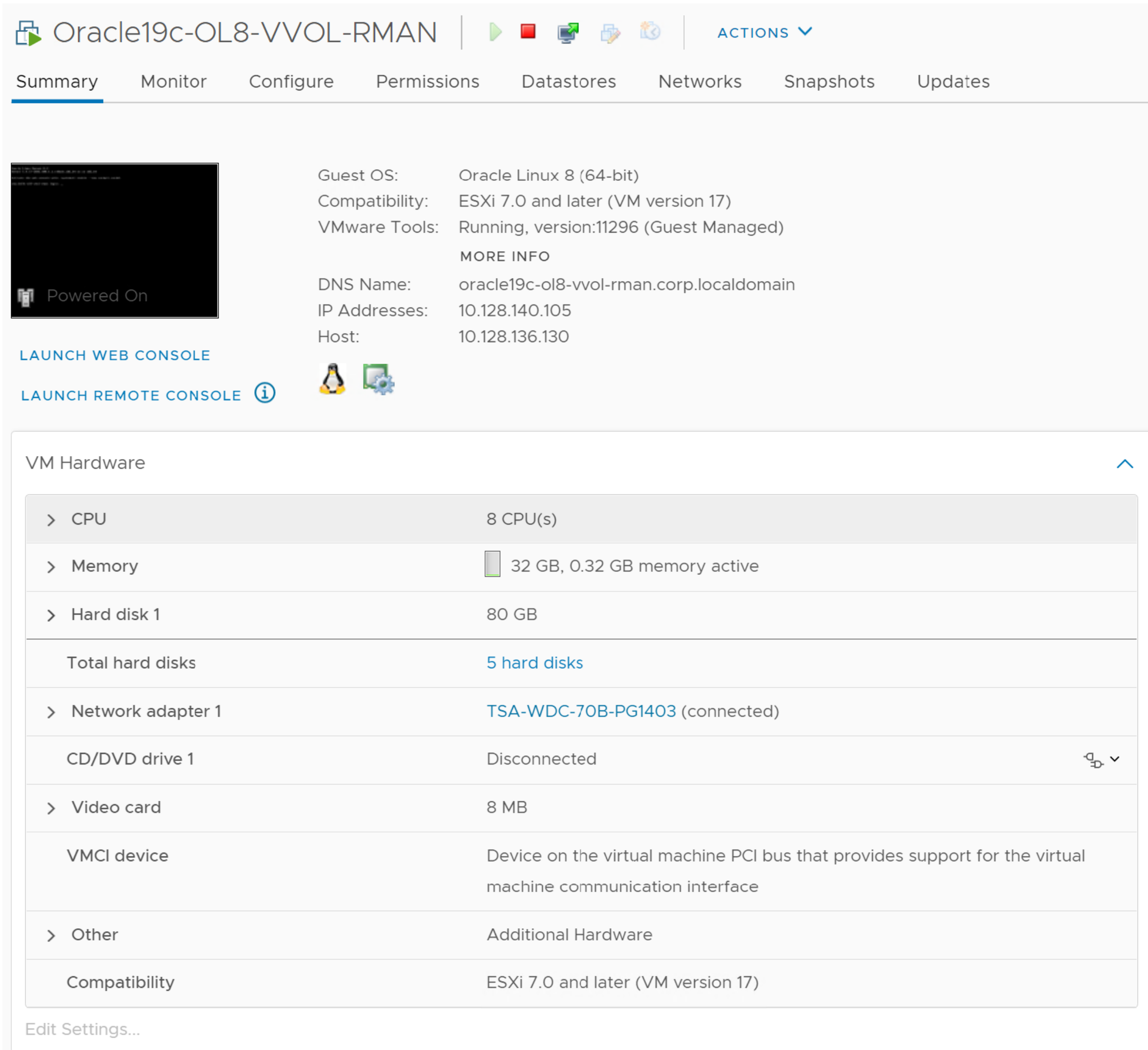


FIGURE 38. VM ORACLE19C-OL8-VVOL-RMAN SUMMARY

VM **Oracle19c-OL8-VVOL-RMAN** virtual disks (VMDKs) are shown below. All SCSI controllers are set to **VMware Paravirtual SCSI Controller** type.

Hard disks	5 total | 1.5 TB
Hard disk 1	80 GB | SCSI(0:0)
Hard disk 2	80 GB | SCSI(0:1)
Hard disk 3	100 GB | SCSI(1:0)
Hard disk 4	250 GB | SCSI(2:0)
Hard disk 5	1024 GB | SCSI(3:0)
SCSI controller 0	VMware Paravirtual
SCSI controller 1	VMware Paravirtual
SCSI controller 2	VMware Paravirtual
SCSI controller 3	VMware Paravirtual

FIGURE 39. VM ORACLE19C-OL8-VVOL-RMAN VMDKS

Virtual disk (VMDK) details:

- Hard Disk 1 – 80GB for operating system
- Hard Disk 2 – 80GB for Oracle Grid and RDBMS binaries
- Hard Disk 3 – 100GB for Oracle Grid Infrastructure Management Repository (GIMR) (Management Database (MGMTDB)) (ASM Disk Group MGMT_DATA)
- Hard Disk 4 – 250GB for Oracle Database **rmandb** database, redo log and archive log files (ASM Disk Group DATA_DG)
- Hard Disk 5 – 1TB for XFS filesystem **/rman** mount point for storing physical RMAN backups

Oracle ASM disk group details:

```
grid@oracle19c-ol8-vvol-rman:+ASM:/home/grid> asmcmd lsdg
State    Type    Rebal  Sector  Logical_Sector  Block       AU  Total_MB  Free_MB  Req_mir_free_MB  Usable_file_MB  Offline_disks  Voting_files  Name
MOUNTED  EXTERN  N         512             512   4096  1048576    255999   223816                0          223816              0             N  DATA_DG/
MOUNTED  EXTERN  N         512             512   4096  4194304    102396   102296                0          102296              0             N  MGMT_DATA/
grid@oracle19c-ol8-vvol-rman:+ASM:/home/grid>
```

FIGURE 40. ORACLE ASM DISK GROUP

Additional vVols-backed VMDKs are shown below:

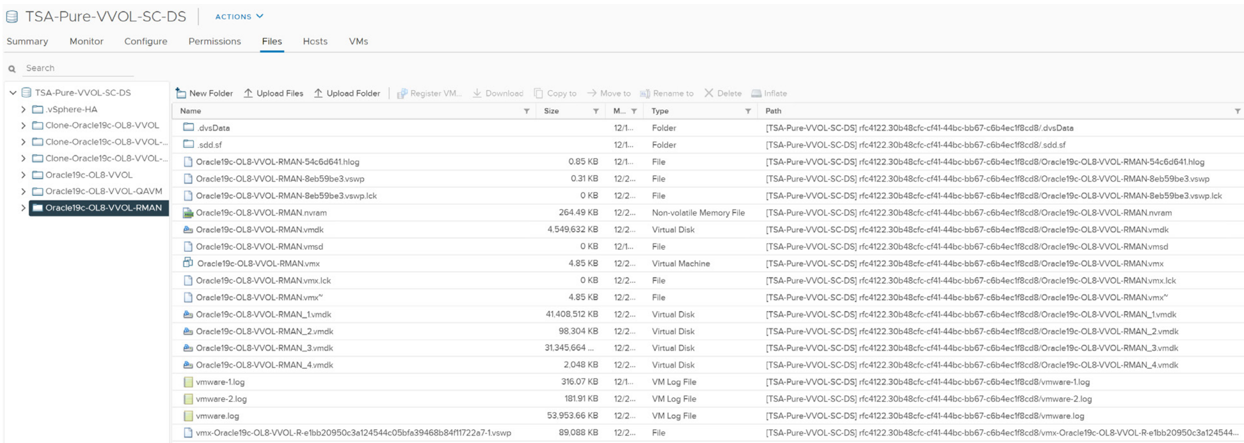


FIGURE 41. VVOLS-BACKED VMDKS

Logging into the Pure x50 storage GUI shows the various vVols representing the different virtual disks of VM **Oracle19c-OL8-VVOL-RMAN.**

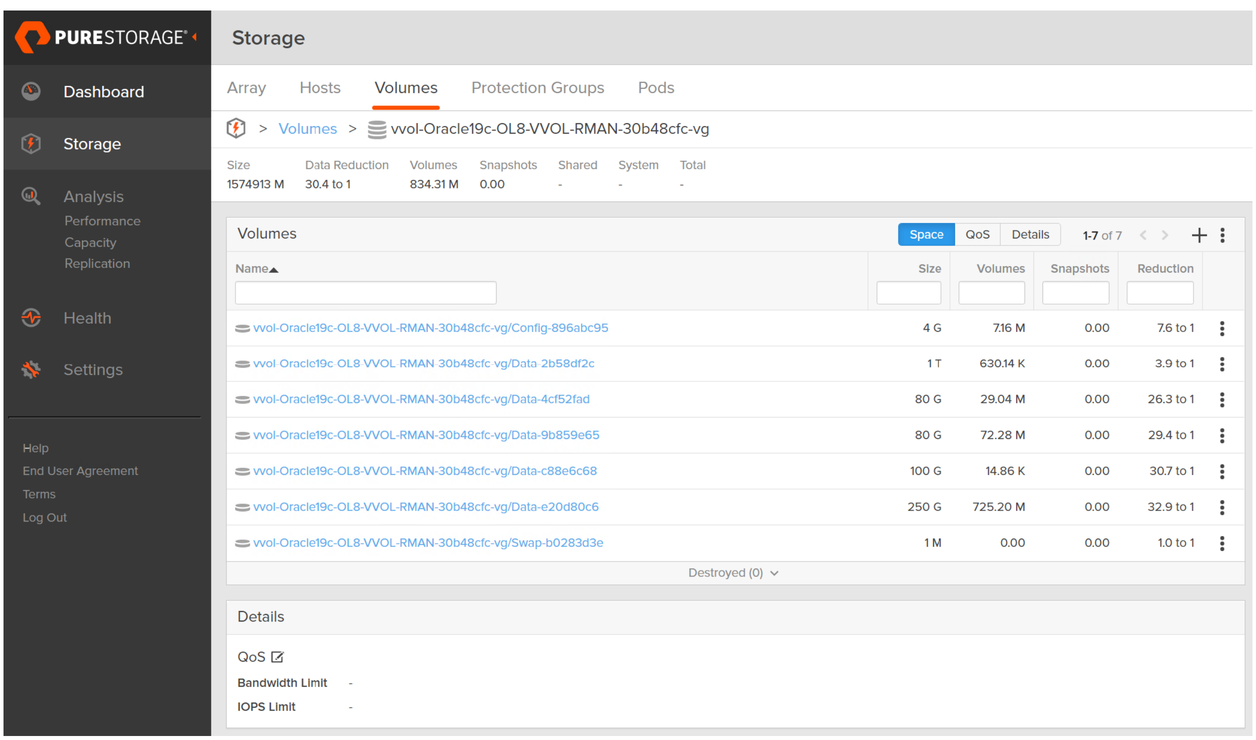


FIGURE 42. PURE STORAGE VOLUMES OF VM ORACLE19C-OL8-VVOL-RMAN

The previous details provide guidance for Oracle RAC storage deployment on vVols using Pure Storage. Network deployment details, along with the basic steps for RAC deployment on VMware vVols using Pure Storage, can be found in *Oracle VMware Hybrid Cloud High Availability Guide REFERENCE ARCHITECTURE.*

A simple 2-node Oracle 19c RAC cluster with VMs **vvolrac1** and **vvolrac2** was created to easily illustrate various use cases.

Both Oracle RAC VMs contain three VMDKs:

- 2 non-shared VMDKs
  - Hard Disk 1 80GB for Operating System with disk mode **Dependent**
  - Hard Disk 1 80GB for Oracle Grid Infrastructure and RDBMS binaries with disk mode **Dependent**
- 1 shared VMDK (1TB) with multi-writer attribute and disk mode **Independent-Persistent**

Details of the shared VMDK with multi-writer flag and disk mode **Independent-Persistent** are shown below:

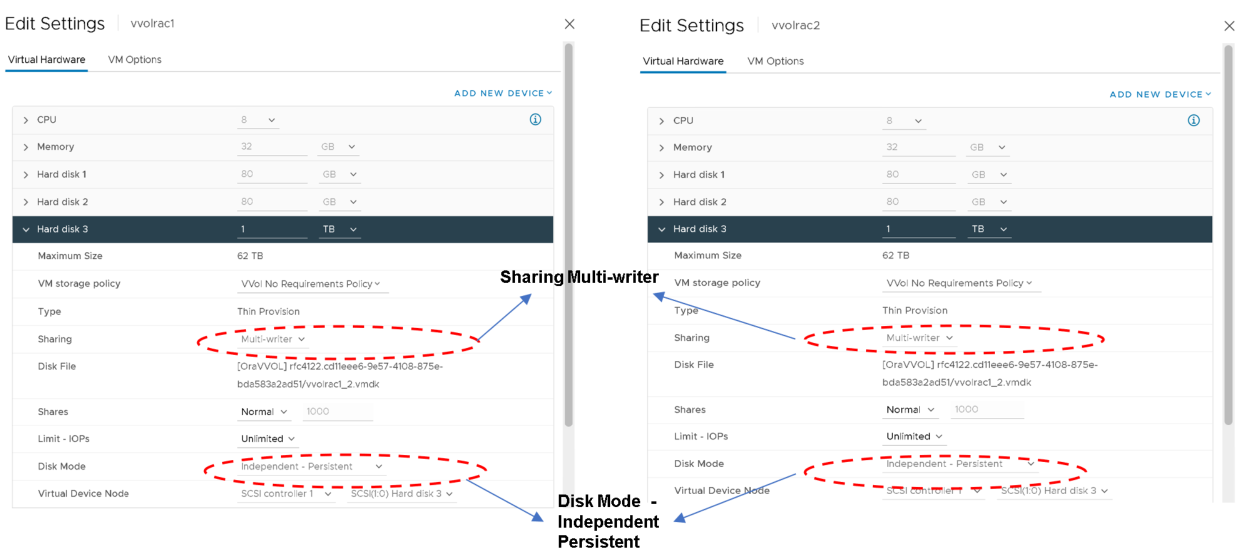


FIGURE 43. ORACLE RAC VVOLRAC SHARED VMDK DETAILS

Solution Validation

Site B was chosen as the site for on-premises Oracle RAC and single-instance deployments.

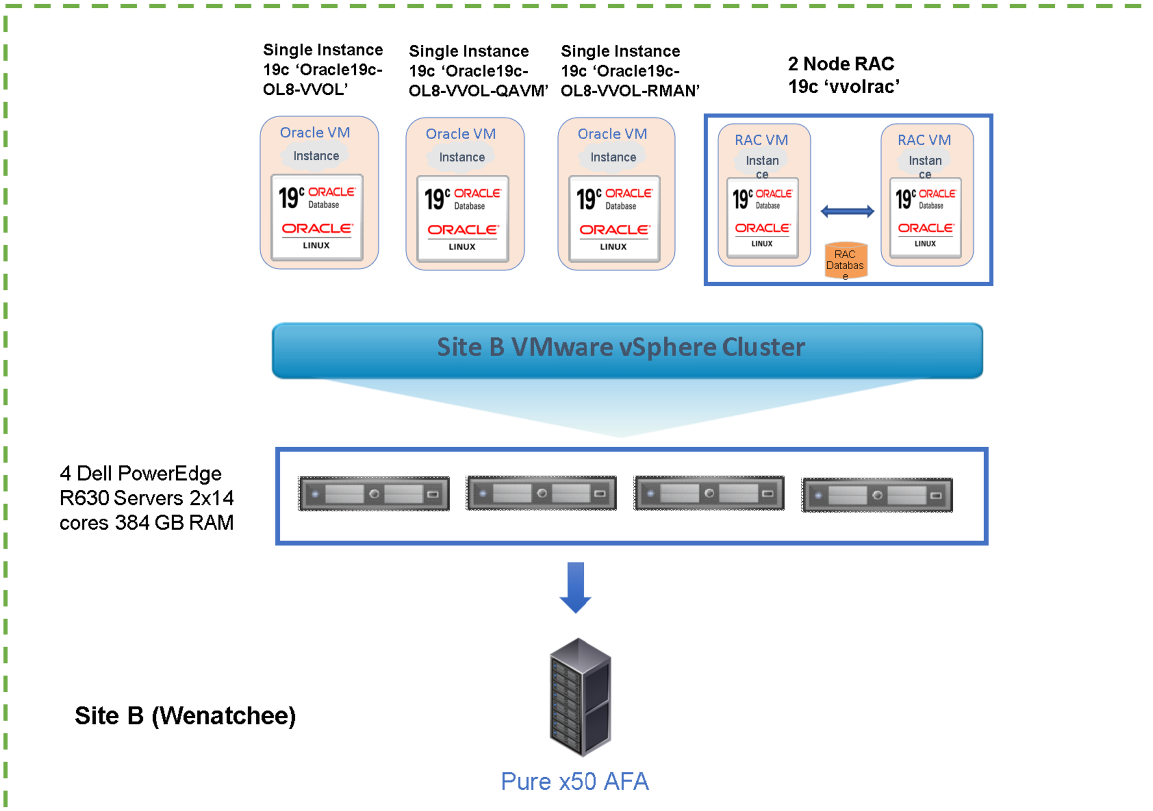


FIGURE 44. 4-NODE VSPHERE CLUSTER WITH CONNECTIVITY TO AFF PURE X50 ARRAY

This solution primarily validated the functional design of Oracle single-instance deployments on vVols storage backed by Pure x50 Storage.

The functional design of Oracle RAC deployments on vVols storage using Pure x50 Storage can be found in *Oracle VMware Hybrid Cloud High Availability Guide REFERENCE ARCHITECTURE*.

Solution Test Overview

This solution validated the following Day 2 operations of both Oracle Single Instance and RAC database on vVols using Pure Storage FlashArray//x50.

- Database backup using vVol-based snapshots and Oracle RMAN
- Database restore and recovery using vVol-based snapshots and Oracle mechanism
- Database cloning using vVol-based snapshots
- Database refresh using vVol-based snapshots
- Database patching using vVol-based snapshots
- VM provisioning and cloning using vVols SPBM

Use Case	Database	Pure Plugin Option	Snapshot	Level of Operation
Database Backup	Single Instance/RAC	Import / Overwrite	VM Level	*VM*
Database Cloning	Single Instance/RAC	Import	VMDK Level	*VMDK*
Database Refresh	Single Instance/RAC	Overwrite	VMDK Level	*VMDK*
Database Patching	Single Instance/RAC	Import / Overwrite	VMDK Level	*VMDK*

TABLE 9. SOLUTION TEST OVERVIEW

The functional design of Oracle RAC deployments on vVols using Pure x50 Storage can be found in *Oracle VMware Hybrid Cloud High Availability Guide REFERENCE ARCHITECTURE.*

Performance testing was not included as part of this reference architecture. Any performance data is a result of the combination of hardware configuration, software configuration, test methodology, test tool, and workload profile used in the testing.

Performance testing can be conducted by using the SLOB tool against the Oracle RAC cluster and generating a load on the database. Oracle AWR and Linux SAR reports can be captured to compare the performance and validate the testing use cases.

Oracle Database Backup

This section validates the database backup using VMware vVol-based snapshots and Oracle RMAN of an Oracle single-instance and Oracle RAC using Pure x50 Storage.

This use case focuses on the use of the Oracle RMAN utility with RMAN catalog database **rmandb** to back up the Oracle production database **vvol19c** and Oracle RAC database **vvolrac**.

Oracle Backup of Single Instance

Two VMs are utilized for this use case:

- Production VM **Oracle19c-OL8-VVOL**
- RMAN VM **Oracle19c-OL8-VVOL-RMAN**

The following steps were used for this testing:

1. Ensure that we can use the Oracle **tnsping** command from production VM **Oracle19c-OL8-VVOL** to check for Oracle service on RMAN VM **Oracle19c-OL8-RMAN**

```
oracle@oracle19c-ol8-vvol:vvol19c:/home/oracle> tnsping rmandb
TNS Ping Utility for Linux: Version 19.0.0.0.0 - Production on 29-DEC-2020 17:06:49
Copyright (c) 1997, 2020, Oracle. All rights reserved.

Used parameter files:

Used TNSNAMES adapter to resolve the alias
Attempting to contact (DESCRIPTION = (ADDRESS_LIST = (ADDRESS = (PROTOCOL = TCP)(HOST = 10.128.140.105)
(PORT = 1521))) (CONNECT_DATA = (SERVER = DEDICATED)(SERVICE_NAME = pdb1)))
OK (0 msec)
oracle@oracle19c-ol8-vvol:vvol19c:/home/oracle>
```

2. Ensure that the production database **vvol19c** is registered with RMAN catalog database **rmandb**. If not, register the database **vvol19c** first with RMAN catalog database **rmandb**.

```
oracle@oracle19c-ol8-vvol:vvol19c:/home/oracle> rman
Recovery Manager: Release 19.0.0.0.0 - Production on Sat Dec 19 20:53:18 2020
Version 19.8.0.0.0
Copyright (c) 1982, 2019, Oracle and/or its affiliates. All rights reserved.

RMAN> connect catalog rman/rman@rmandb
connected to recovery catalog database

RMAN> create catalog;
recovery catalog created

RMAN> connect target /
connected to target database: VVOL19C (DBID=2713363709)

RMAN> register database;
database registered in recovery catalog
starting full resync of recovery catalog
full resync complete

RMAN> report schema;

Report of database schema for database with db_unique_name VVOL19C

List of Permanent Datafiles
===================
File   Size(MB)   Tablespace           RB segs    Datafile Name
----   ----------   -------------        ----------   ----------------
1      512        SYSTEM               YES        +DATA_DG/vvol19c/system_01.dbf
2      185        PDB$SEED:SYSTEM      NO         +DATA_DG/pdbseed/system_01.dbf
3      512        SYSAUX               NO         +DATA_DG/vvol19c/sysaux_01.dbf
4      130        PDB$SEED:SYSAUX      NO         +DATA_DG/pdbseed/sysaux_01.dbf
5      512        UNDOTBS01            YES        +DATA_DG/vvol19c/undotbs01_01.dbf
6      512        USERS                NO         +DATA_DG/vvol19c/users_01.dbf
7      512        PDB$SEED:USERS       NO         +DATA_DG/pdbseed/users_01.dbf
8      512        PDB$SEED:USERTBS     NO         +DATA_DG/pdbseed/usertbs_01.dbf
9      195        PDB1:SYSTEM          NO         +DATA_DG/pdb1/system_01.dbf
10     170        PDB1:SYSAUX          NO         +DATA_DG/pdb1/sysaux_01.dbf
11     512        PDB1:USERS           NO         +DATA_DG/pdb1/users_01.dbf
12     512        PDB1:USERTBS         NO         +DATA_DG/pdb1/usertbs_01.dbf
13     250        PDB1:PDB_USER        NO         +DATA_DG/pdb1/pdb_user_01.dbf


List of Temporary Files
=================
File   Size(MB)   Tablespace           Maxsize(MB)     Tempfile Name
----   ----------   -------------        --------------      -----------------
1      512        TEMP                 32767           +DATA_DG/vvol19c/temp_01.dbf
2      512        PDB$SEED:TEMP        32767           +DATA_DG/pdbseed/temp_01.dbf
3      512        PDB1:TEMP            32767           +DATA_DG/pdb1/temp_01.dbf

RMAN>
```

3. Take two RMAN backups of the database current controlfile; the destination of the backups is the ASM diskgroup FRA_DG
   - controlfile control_db_start will be used to mount the clone of the database on RMAN VM **Oracle19c-OL-RMAN**. The CONTROL_FILES parameter in the mount host init.ora will point to the control_db_start control file
   - controlfile **control_rman_backup** will be part of the RMAN backupset

As part of the RMAN backup process, force the database **vvol19c** current log to be archived as well.

```
rman target / <<EOF
run
 {
 allocate channel t1 type disk;
 alter system archive log current;
 copy current controlfile to +FRA_DG/vvol19c/CONTROLFILE/control_db_start ;
 copy current controlfile to +FRA_DG/vvol19c/CONTROLFILE/control_rman_backup ;
 }
EOF

RMAN>
Recovery Manager: Release 19.0.0.0.0 - Production on Tue Dec 29 19:25:02 2020
Version 19.8.0.0.0
Copyright (c) 1982, 2019, Oracle and/or its affiliates. All rights reserved.
connected to target database: VVOL19C (DBID=2713363709)

RMAN>
using target database control file instead of recovery catalog

allocated channel: t1
channel t1: SID=1015 device type=DISK

Statement processed

Starting backup at 29-DEC-20
channel t1: starting datafile copy
copying current control file
output file name=+FRA_DG/vvol19c/CONTROLFILE/control_db_start tag=TAG20201229T192503 RECID=3 STAMP=1060457104
channel t1: datafile copy complete, elapsed time: 00:00:01
Finished backup at 29-DEC-20

Starting backup at 29-DEC-20
channel t1: starting datafile copy
copying current control file
output file name=+FRA_DG/vvol19c/CONTROLFILE/control_rman_backup tag=TAG20201229T192505 RECID=4
STAMP=1060457105
channel t1: datafile copy complete, elapsed time: 00:00:01
Finished backup at 29-DEC-20

Starting Control File Autobackup at 29-DEC-20
piece handle=/u01/app/oracle/product/19.0.0/dbhome_1/dbs/c-2713363709-20201229-00 comment=NONE
Finished Control File Autobackup at 29-DEC-20
released channel: t1
RMAN>
Recovery Manager complete.
oracle@oracle19c-ol8-vvol:vvol19c:/home/oracle>
```

```
grid@oracle19c-ol8-vvol:+ASM:/home/grid> asmcmd ls -l +FRA_DG/vvol19c/CONTROLFILE
Type         Redund  Striped  Time             Sys  Name
CONTROLFILE  UNPROT  FINE     DEC 29 19:00:00  Y    Backup.361.1060456915
CONTROLFILE  UNPROT  FINE     DEC 29 19:00:00  Y    Backup.362.1060456917
CONTROLFILE  UNPROT  FINE     DEC 29 19:00:00  N    control_db_start => +FRA_DG/vvol19c/CONTROLFILE/Backup.361.1060456915
CONTROLFILE  UNPROT  FINE     DEC 29 19:00:00  N    control_rman_backup => +FRA_DG/vvol19c/CONTROLFILE/Backup.362.1060456917
grid@oracle19c-ol8-vvol:+ASM:/home/grid>
```

FIGURE 45. FRA_DG BACKUP CONTROFILE

4. On production VM **Oracle19c-OL8-VVOL**, resynchronize the RMAN catalog with the production database **vvol19c**. This will add the most recent archive log info into the RMAN catalog.

   rman target=/ catalog=rman/rman@rmandb <<EOF
   resync catalog;
   exit;
   EOF

   RMAN>
   Recovery Manager: Release 19.0.0.0.0 - Production on Tue Dec 29 19:30:10 2020
   Version 19.8.0.0.0
   Copyright (c) 1982, 2019, Oracle and/or its affiliates. All rights reserved.

   connected to target database: VVOL19C (DBID=2713363709)
   connected to recovery catalog database

   RMAN> starting full resync of recovery catalog
   full resync complete
   RMAN>
   Recovery Manager complete.
   oracle@oracle19c-ol8-vvol:vvol19c:/home/oracle>

5. Take a snapshot of the database.

In the case of production VM **Oracle19c-OL8-VVOL**, the database data and redo log files were on ASM disk group DATA_DG, the database archive log files were on ASM disk group FRA_DG.

We can take a snapshot of the database in two ways:

– A traditional VM-level snapshot using a web client
– A VMDK-level snapshot using the Pure Storage Plugin

A VM-level snapshot can be taken as shown below:

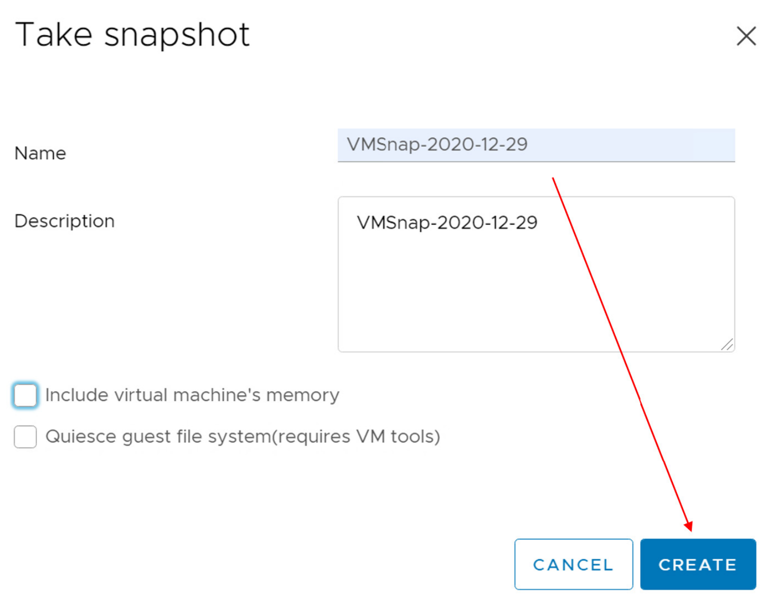


FIGURE 46. TAKING A VM-LEVEL SNAPSHOT

The VM-level snapshot has been successfully taken for production VM **Oracle19c-OL8-VVOL**.

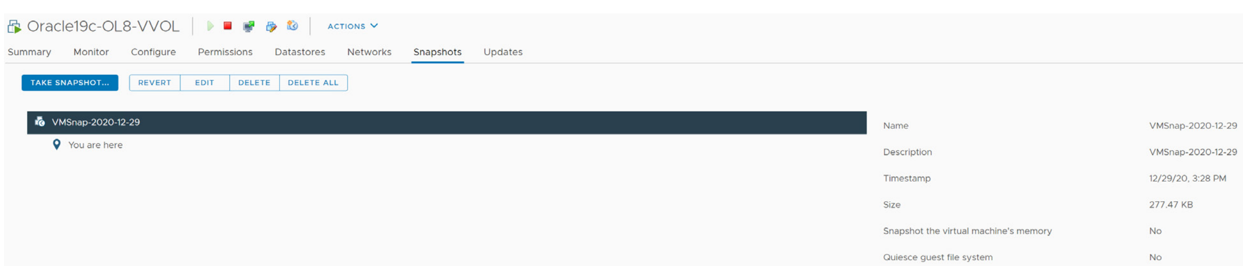


FIGURE 47. VM-LEVEL SNAPSHOT

Logging into the Pure Storage GUI indicates snapshots are created for all VM VMDKs.

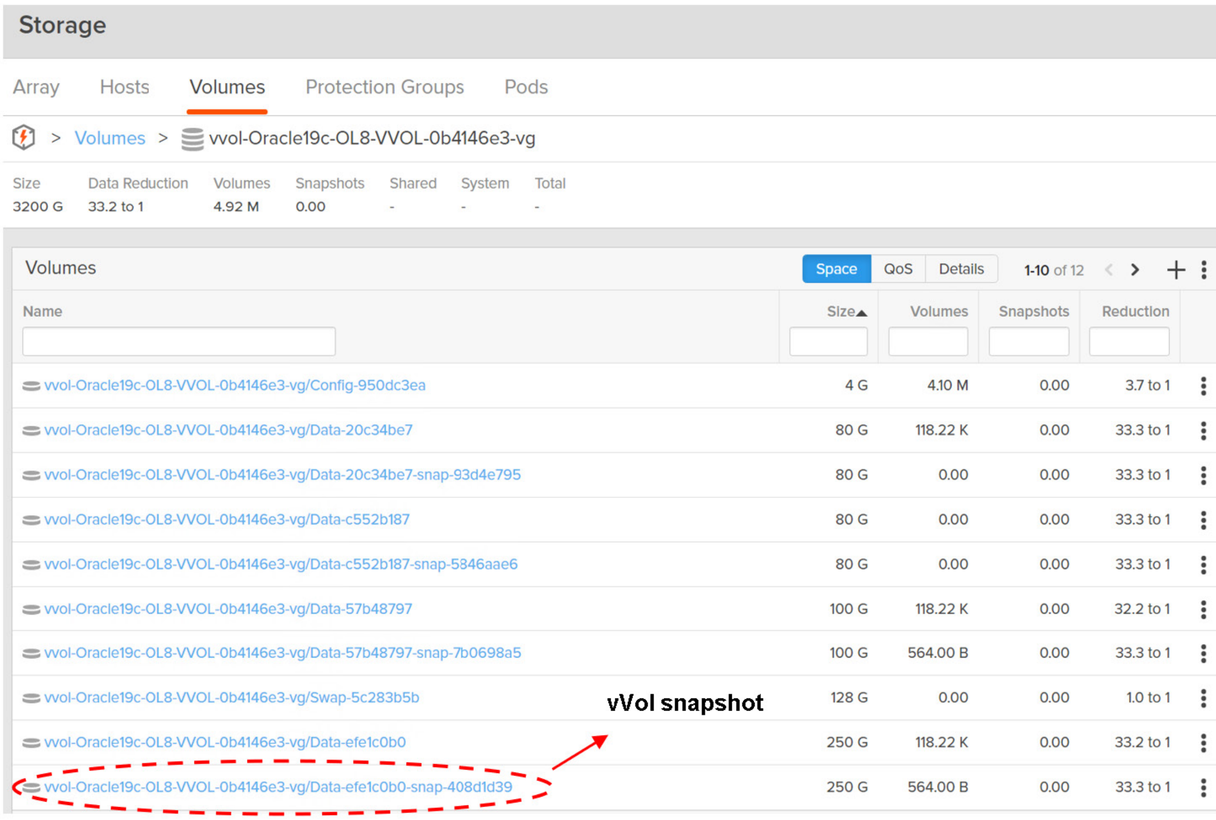


FIGURE 48. PURE STORAGE GUI – VOLUMES

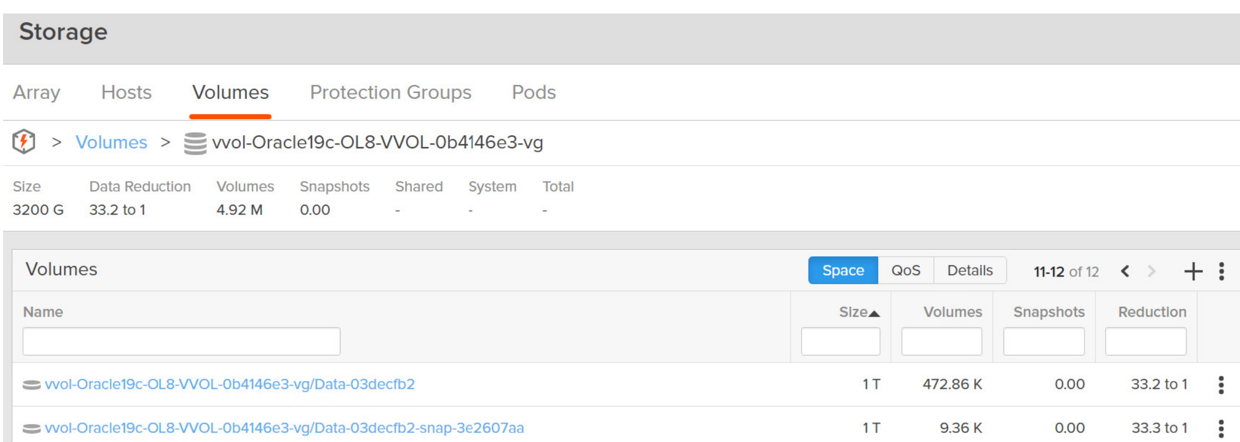


FIGURE 49. PURE STORAGE GUI – VOLUMES

A VMDK-level snapshot using the Pure Storage Plugin can be taken as shown below:

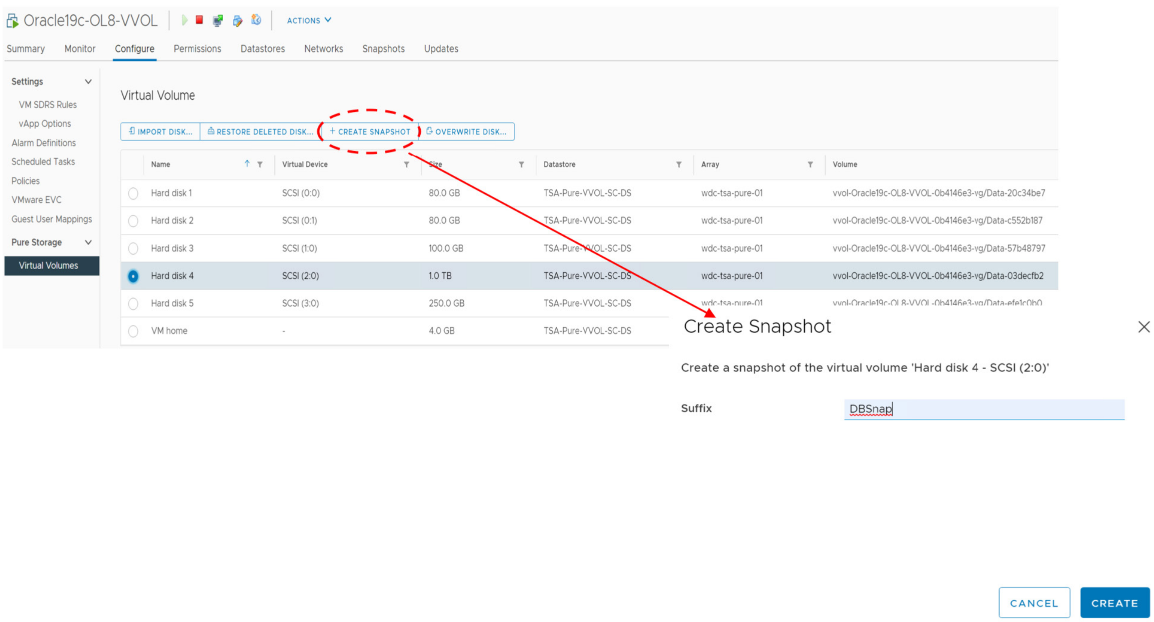


FIGURE 50. VMDK-LEVEL SNAPSHOT USING PURE STORAGE PLUGIN

In this use case, a VM-level snapshot is preferred over a VMDK-level snapshot as a VM-level snapshot will include a snapshot for both ASM disk groups DATA_DG and FRA_DG taken at the same time. A VMDK-level snapshot using the Pure Storage Plugin, would require two separate VMDK-level snapshots, one for the DATA VMDK and one for the FRA VMDK.

Remember, the VMDKs containing redo log files, controlfiles and database files must be snapshotted as a part of a single consistency group to avoid any timestamp mismatch.

In this use case, all redo log files, controlfiles and database files are part of the DATA_DG ASM disk group. A VMDK-level snapshot for both DATA_DG and FRA_DG VMDKs would have resulted in two different snapshot operations with different timestamps.

While technically workable for this use case, we chose to adopt Oracle and storage best practice recommendations to snapshot the entire database in one operation and avoid any timestamp-related issues.

A VMDK-level snapshot is can be employed for use cases like database cloning, database refresh, or database patching which are explained in the sections below.

6. On the RMAN VM **Oracle19c-OL8-RMAN**, use the Pure Plugin option **Import Disk** to import or create a new DATA VMDK (1TB) and a new FRA VMDK (250GB) from the production VM **Oracle19c-OL8-VVOL** from the snapshot **VMSnap-2020-12-29.**

To import a new DATA VMDK (1TB) from the production VM **Oracle19c-OL8-VVOL** snapshot **VMSnap-2020-12-29**, do the following:

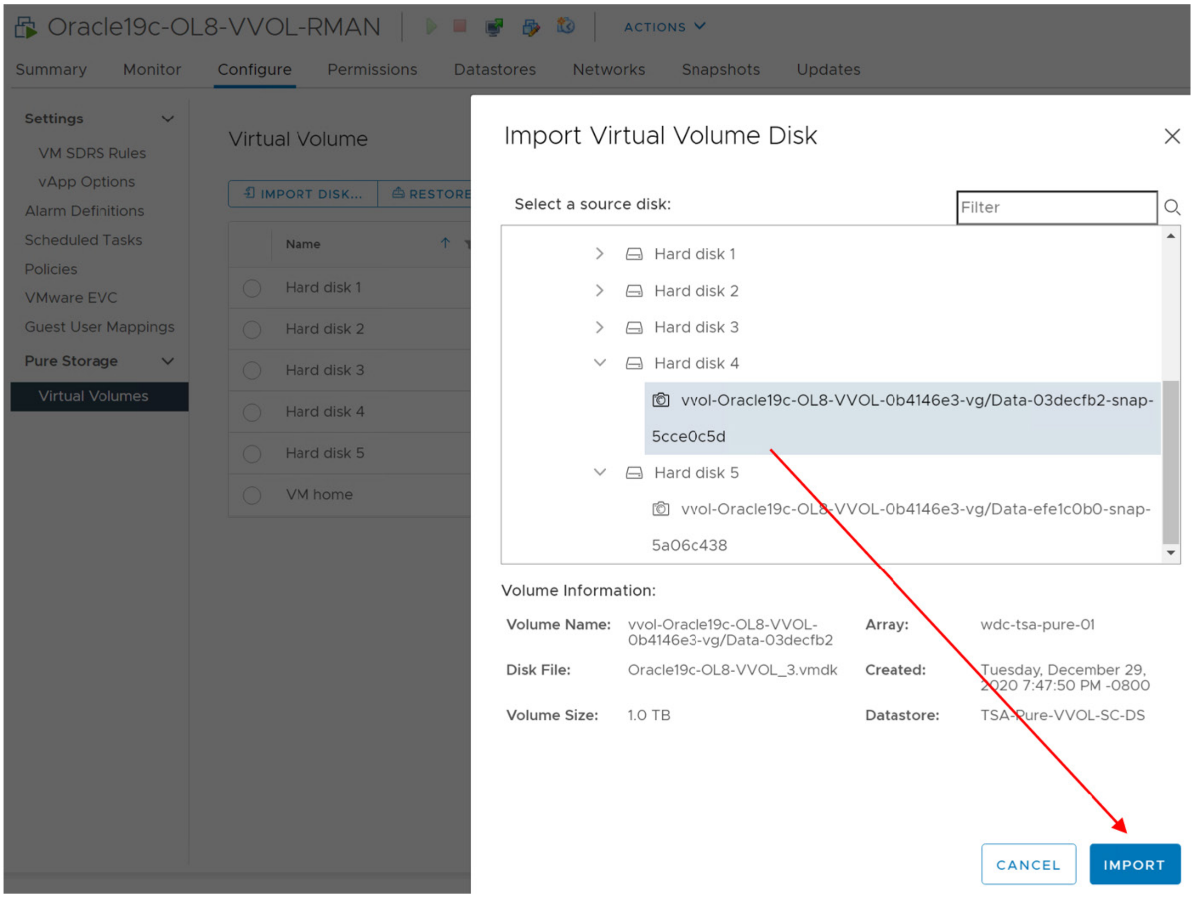


FIGURE 51. DATA VMDK IMPORT

A new DATA VMDK (1TB) is created on RMAN VM **Oracle19c-OL8-RMAN** at SCSI 0:2.

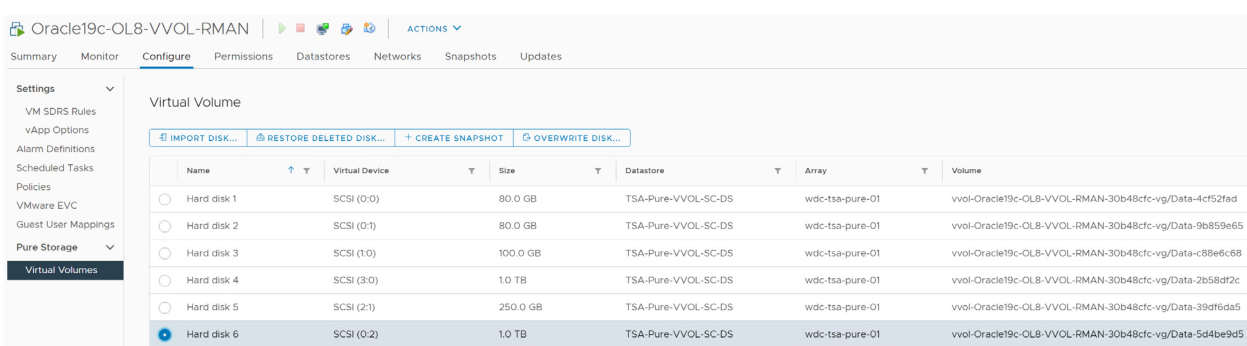


FIGURE 52. NEW DATA VMDK

To import a new FRA VMDK (250GB) from the production VM **Oracle19c-OL8-VVOL** snapshot **VMSnap-2020-12-29**, do the following:

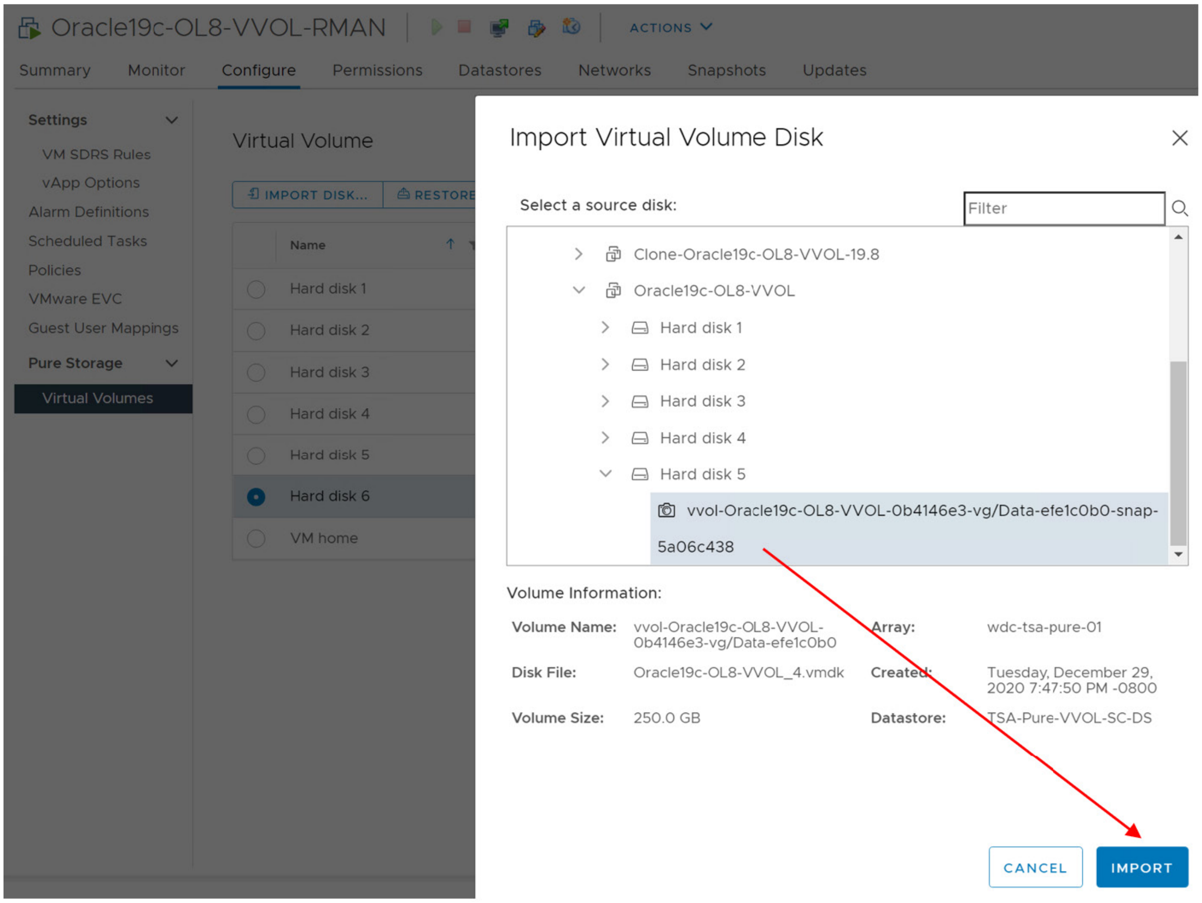


FIGURE 53. FRA VMDK IMPORT

A new FRA VMDK (250GB) is created on RMAN VM **Oracle19c-OL8-RMAN** at SCSI 0:3.

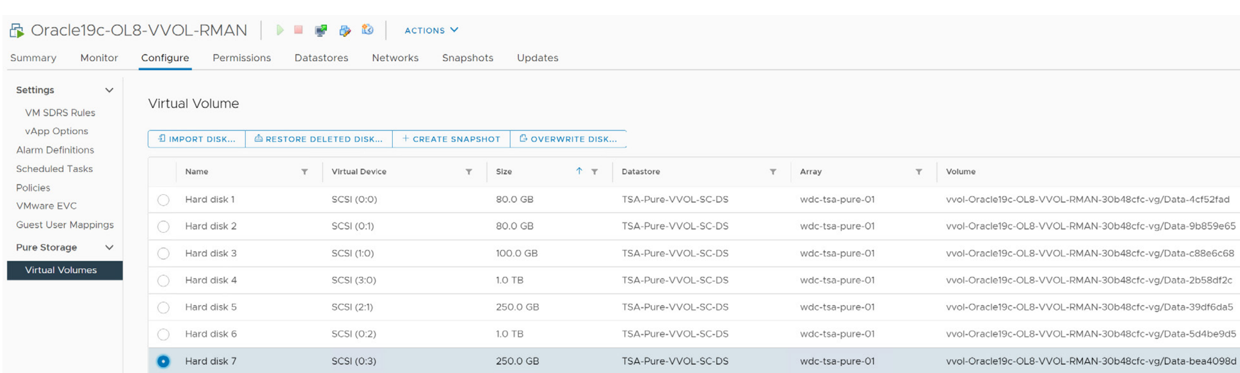


FIGURE 54. NEW FRA VMDK

RMAN VM **Oracle19c-OL8-RMAN** VMDK details are shown below:

FIGURE 55. RMAN VM ORACLE19C-OL8-RMAN VMDKS

7. Delete the snapshot taken on production VM **Oracle19c-OL8-VVOL.**
8. On RMAN VM **Oracle19c-OL8-RMAN**, scan for new ASM disks using the Oracle **oracleasm** command to discover new VMDKs **DATA_01** and **FRA_01.**

```
[root@oracle19c-ol8-vvol-rman ~]# oracleasm scandisks ; oracleasm listdisks
Reloading disk partitions: done
Cleaning any stale ASM disks...
Scanning system for ASM disks...
Instantiating disk "DATA_01"
Instantiating disk "FRA_01"
DATA_01
FRA_01
MGMT_DATA01
RMAN_DATA_01
[root@oracle19c-ol8-vvol-rman ~]#
```

FIGURE 56. ASM DISK SCAN (ORACLEASM)

Mount the new disk groups **+DATA_DG** and **+FRA_DG.**

```
grid@oracle19c-ol8-vvol-rman:+ASM:/home/grid> asmcmd mount DATA_DG
grid@oracle19c-ol8-vvol-rman:+ASM:/home/grid> asmcmd mount FRA_DG
grid@oracle19c-ol8-vvol-rman:+ASM:/home/grid> asmcmd lsdg
State    Type    Rebal  Sector  Logical_Sector  Block       AU  Total_MB  Free_MB  Req_mir_free_MB  Usable_file_MB  Offline_disks  Voting_files  Name
MOUNTED  EXTERN  N         512             512   4096  1048576   1048575  1036823                0         1036823              0             N  DATA_DG/
MOUNTED  EXTERN  N         512             512   4096  1048576    255999   253665                0          253665              0             N  FRA_DG/
MOUNTED  EXTERN  N         512             512   4096  4194304    102396   102296                0          102296              0             N  MGMT_DATA/
MOUNTED  EXTERN  N         512             512   4096  1048576    255999   223714                0          223714              0             N  RMAN_DATA_DG/
grid@oracle19c-ol8-vvol-rman:+ASM:/home/grid>
```

FIGURE 57. ASMCMD MOUNT

9. On RMAN VM **Oracle19c-OL8-RMAN**, start the database **vvol19c** in mount mode.

Make sure the init.ora file for the database **vvol19c** points to **+FRA_DG/vvol19c/CONTROLFILE/control_db_start.**

Make appropriate changes to the database **vvol19c** init.ora file for memory and file directory settings.

Copy the **orapwvvol19c** file from production VM **Oracle19c-OL8 VVOL** to RMAN VM **Oracle19c-OL8-RMAN.**

```
oracle@oracle19c-ol8-vvol-rman:rmandb:/home/oracle> export ORACLE_SID=vvol19c

oracle@oracle19c-ol8-vvol-rman:vvol19c:/home/oracle> sqlplus / as sysdba
SQL*Plus: Release 19.0.0.0.0 - Production on Tue Dec 29 21:08:18 2020
Version 19.8.0.0.0
Copyright (c) 1982, 2020, Oracle. All rights reserved.

Connected to an idle instance.

SQL> startup mount pfile=/u01/app/oracle/product/19.0.0/dbhome_1/dbs/initvvol19c.ora
ORACLE instance started.
Total System Global Area     1.2885E+10 bytes
Fixed Size                   12691536 bytes
Variable Size                1811939328 bytes
Database Buffers             1.1039E+10 bytes
Redo Buffers                 20860928 bytes
Database mounted.
SQL>
```

10. On the RMAN VM **Oracle19c-OL8-RMAN**, backup the database **vvol19c** using RMAN including the archived logs and backup control file.

```
export ORACLE_SID=vvol19c

rman target=/ catalog=rman/rman@rmandb <<EOF
run
{
allocate channel t1 type disk;
backup controlfilecopy +FRA_DG/vvol19c/CONTROLFILE/control_rman_backup format /rman/vvol19c/ctl_%d_%s_%p_%t ;
backup as backupset database format /rman/vvol19c/db_%d_%s_%p_%t ;
backup archivelog all format /rman/vvol19c/al_%d_%s_%p_%t ;
release channel t1;
}
exit
EOF
```

```
Recovery Manager: Release 19.0.0.0.0 - Production on Tue Dec 29 22:09:08 2020
Version 19.8.0.0.0
Copyright (c) 1982, 2019, Oracle and/or its affiliates. All rights reserved.

connected to target database: VVOL19C (DBID=2713363709, not open)
connected to recovery catalog database

RMAN>
allocated channel: t1
channel t1: SID=384 device type=DISK

Starting backup at 29-DEC-20
channel t1: starting full datafile backup set
channel t1: specifying datafile(s) in backup set
input control file copy name=+FRA_DG/vvol19c/CONTROLFILE/control_rman_backup
channel t1: starting piece 1 at 29-DEC-20
channel t1: finished piece 1 at 29-DEC-20
piece handle=/rman/vvol19c/ctl_VVOL19C_9_1_1060466951 tag=TAG20201229T220910 comment=NONE
channel t1: backup set complete, elapsed time: 00:00:01
Finished backup at 29-DEC-20

Starting backup at 29-DEC-20
channel t1: starting full datafile backup set
channel t1: specifying datafile(s) in backup set
input datafile file number=00003 name=+DATA_DG/vvol19c/sysaux_01.dbf
input datafile file number=00001 name=+DATA_DG/vvol19c/system_01.dbf
input datafile file number=00005 name=+DATA_DG/vvol19c/undotbs01_01.dbf
input datafile file number=00006 name=+DATA_DG/vvol19c/users_01.dbf
channel t1: starting piece 1 at 29-DEC-20
channel t1: finished piece 1 at 29-DEC-20
piece handle=/rman/vvol19c/db_VVOL19C_10_1_1060466953 tag=TAG20201229T220913 comment=NONE
channel t1: backup set complete, elapsed time: 00:00:03
channel t1: starting full datafile backup set
channel t1: specifying datafile(s) in backup set
input datafile file number=00011 name=+DATA_DG/pdb1/users_01.dbf
input datafile file number=00012 name=+DATA_DG/pdb1/usertbs_01.dbf
input datafile file number=00013 name=+DATA_DG/pdb1/pdb_user_01.dbf
input datafile file number=00009 name=+DATA_DG/pdb1/system_01.dbf
input datafile file number=00010 name=+DATA_DG/pdb1/sysaux_01.dbf
channel t1: starting piece 1 at 29-DEC-20
channel t1: finished piece 1 at 29-DEC-20
piece handle=/rman/vvol19c/db_VVOL19C_11_1_1060466956 tag=TAG20201229T220913 comment=NONE
channel t1: backup set complete, elapsed time: 00:00:01
channel t1: starting full datafile backup set
channel t1: specifying datafile(s) in backup set
input datafile file number=00007 name=+DATA_DG/pdbseed/users_01.dbf
input datafile file number=00008 name=+DATA_DG/pdbseed/usertbs_01.dbf
input datafile file number=00002 name=+DATA_DG/pdbseed/system_01.dbf
input datafile file number=00004 name=+DATA_DG/pdbseed/sysaux_01.dbf
channel t1: starting piece 1 at 29-DEC-20
channel t1: finished piece 1 at 29-DEC-20
piece handle=/rman/vvol19c/db_VVOL19C_12_1_1060466957 tag=TAG20201229T220913 comment=NONE
channel t1: backup set complete, elapsed time: 00:00:01
Finished backup at 29-DEC-20
```

```
Starting backup at 29-DEC-20
channel t1: starting archived log backup set
channel t1: specifying archived log(s) in backup set
input archived log thread=1 sequence=64 RECID=1 STAMP=1059237675
……
………. 
input archived log thread=1 sequence=197 RECID=134 STAMP=1060457103
channel t1: starting piece 1 at 29-DEC-20
channel t1: finished piece 1 at 29-DEC-20
piece handle=/rman/vvol19c/al_VVOL19C_15_1_1060466966 tag=TAG20201229T220919 comment=NONE
channel t1: backup set complete, elapsed time: 00:00:03
Finished backup at 29-DEC-20
released channel: t1
RMAN>
Recovery Manager complete.
oracle@oracle19c-ol8-vvol-rman:vvol19c:/rman>
```

RMAN Backup is successful

```
oracle@oracle19c-ol8-vvol-rman:vvol19c:/rman/vvol19c> ll
total 4242212
-rw-r----- 1 oracle asmadmin 963555328 Dec 29 22:09 al_VVOL19C_13_1_1060466960
-rw-r----- 1 oracle asmadmin 962817536 Dec 29 22:09 al_VVOL19C_14_1_1060466963
-rw-r----- 1 oracle asmadmin 906104320 Dec 29 22:09 al_VVOL19C_15_1_1060466966
-rw-r----- 1 oracle asmadmin 23740416 Dec 29 22:09 ctl_VVOL19C_9_1_1060466951
-rw-r----- 1 oracle asmadmin 958324736 Dec 29 22:09 db_VVOL19C_10_1_1060466953
-rw-r----- 1 oracle asmadmin 288104448 Dec 29 22:09 db_VVOL19C_11_1_1060466956
-rw-r----- 1 oracle asmadmin 241377280 Dec 29 22:09 db_VVOL19C_12_1_1060466957
oracle@oracle19c-ol8-vvol-rman:vvol19c:/rman/vvol19c>
```

11. On successful completion of RMAN backup, On RMAN VM **Oracle19c-OL8-RMAN**

– shutdown database **vvol19c**

– dismount the **vvol19c** ASM disk groups **DATA_DG** and **FRA_DG**

– delete the **vvol19c** ASM disks **DATA_01** and **FRA_01** using the **oracleasm deletedisk** command

– delete the added VMDKs from the mount VM **Oracle19c-OL8-RMAN**

```
oracle@oracle19c-ol8-vvol-rman:vvol19c:/home/oracle> sqlplus / as sysdba
SQL*Plus: Release 19.0.0.0.0 - Production on Wed Dec 30 12:33:04 2020
Version 19.8.0.0.0
Copyright (c) 1982, 2020, Oracle. All rights reserved.

Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.8.0.0.0

SQL> shutdown immediate;
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> exit
Disconnected from Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.8.0.0.0
oracle@oracle19c-ol8-vvol-rman:vvol19c:/home/oracle>
```

```
grid@oracle19c-ol8-vvol-rman:+ASM:/home/grid> asmcmd umount DATA_DG
grid@oracle19c-ol8-vvol-rman:+ASM:/home/grid> asmcmd umount FRA_DG

grid@oracle19c-ol8-vvol-rman:+ASM:/home/grid> asmcmd lsdg
```

State	Type	Rebal	Sector	Logical_Sector	Block	AU	Total_MB	Free_MB	Req_mir_free_MB	Usable_file_MB	Offline_disks	Voting_files	Name
MOUNTED	EXTERN	N	512	512	4096	4194304	102396	102296	0	102296	0	N	MGMT_DATA/
MOUNTED	EXTERN	N	512	512	4096	1048576	255999	223714	0	223714	0	N	RMAN_DATA_DG/

```
grid@oracle19c-ol8-vvol-rman:+ASM:/home/grid>

[root@oracle19c-ol8-vvol-rman ~]# oracleasm deletedisk DATA_01
Clearing disk header: done
Dropping disk: done
[root@oracle19c-ol8-vvol-rman ~]#

[root@oracle19c-ol8-vvol-rman ~]# oracleasm deletedisk FRA_01
Clearing disk header: done
Dropping disk: done
[root@oracle19c-ol8-vvol-rman ~]#

[root@oracle19c-ol8-vvol-rman ~]# oracleasm scandisks ; oracleasm listdisks
Reloading disk partitions: done
Cleaning any stale ASM disks...
Scanning system for ASM disks...
MGMT_DATA01
RMAN_DATA_01
[root@oracle19c-ol8-vvol-rman ~]#
```

12. As the RMAN backup is typically a regular database Day 2 operation, we do not have to delete the newly added ASM disks nor the VMDKs from RMAN VM **Oracle19c-OL8-RMAN.**

We can use the Pure Plugin **Overwrite Disk** option to overwrite the newly added VMDKs for **vvol19c** with the same snapshot taken of production VM **Oracle19c-OL8-VVOL.**

On RMAN VM **Oracle19c-OL8-RMAN**, choose the newly added DATA_DG disk 1TB and click **Overwrite Disk**.

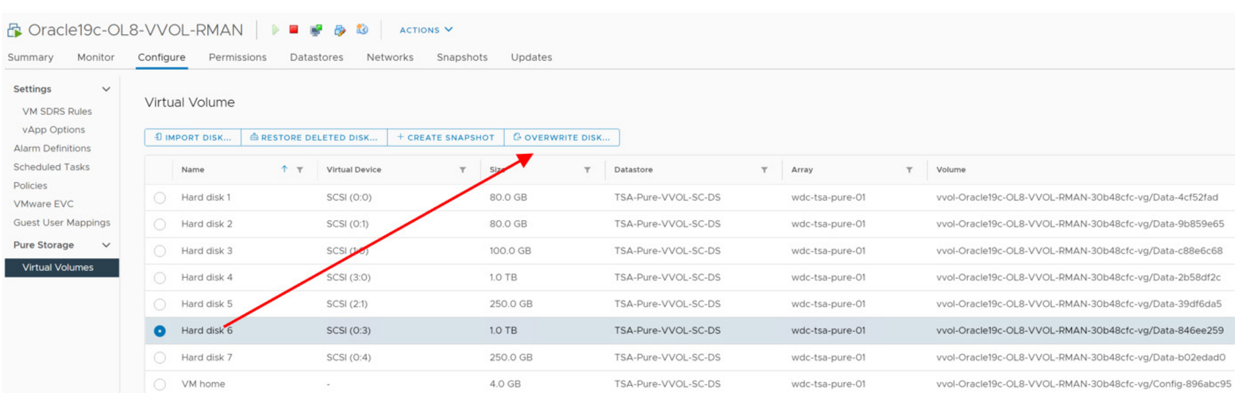


FIGURE 58. DISK OVERWRITE

Under production VM **Oracle19c-OL8-VVOL**, choose **Hard Disk 4 (1TB)** and Click **Overwrite**.

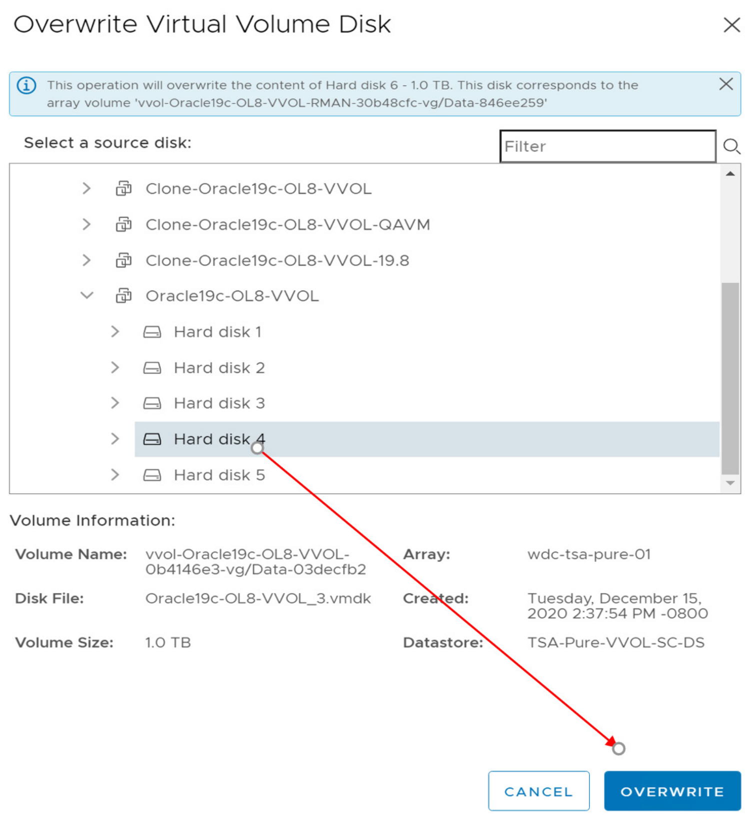


FIGURE 59. DISK OVERWRITE

Perform the same steps for newly added FRA_DG (250GB) disk. Under production VM **Oracle19c-OL8-VVOL**, Choose **Hard Disk 5 (250GB)** and click **Overwrite**.

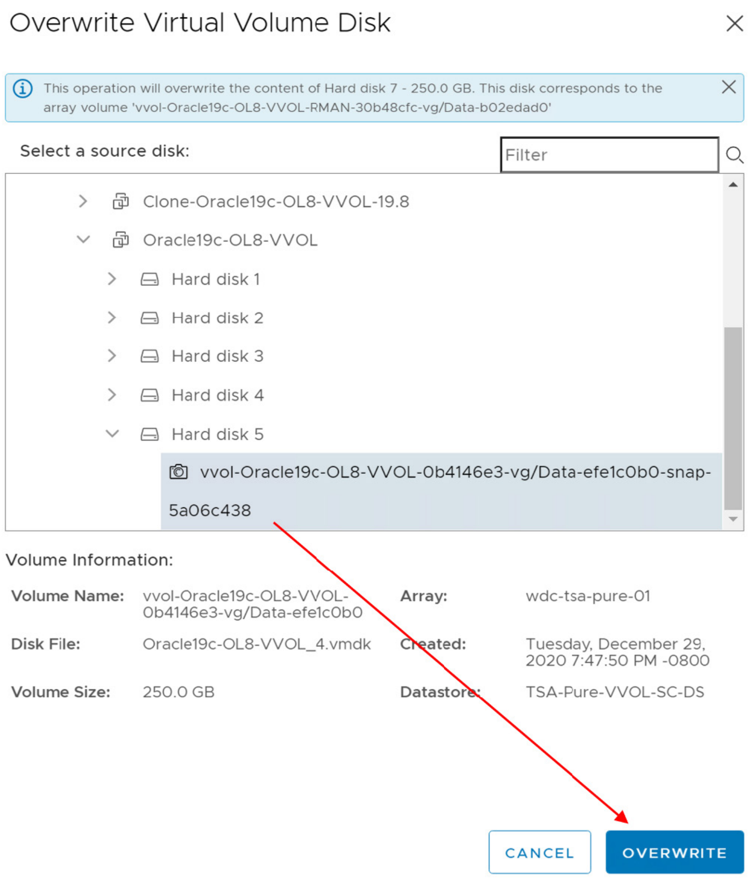


FIGURE 60. DISK OVERWRITE

Perform rescan of ASM disks on RMAN VM **Oracle19c-OL8-RMAN** using the **oracleasm scandisks** command.

```
[root@oracle19c-ol8-vvol-rman ~]# oracleasm scandisks ; oracleasm listdisks
Reloading disk partitions: done
Cleaning any stale ASM disks...
Scanning system for ASM disks...
Instantiating disk "DATA_01"
Instantiating disk "FRA_01"
DATA_01
FRA_01
MGMT_DATA01
RMAN_DATA_01
[root@oracle19c-ol8-vvol-rman ~]#
```

grid@oracle19c-ol8-vvol-rman:+ASM:/home/grid> asmcmd mount DATA_DG
grid@oracle19c-ol8-vvol-rman:+ASM:/home/grid> asmcmd mount FRA_DG

grid@oracle19c-ol8-vvol-rman:+ASM:/home/grid> asmcmd lsdg

State	Type	Rebal	Sector	Logical_Sector	Block	AU	Total_MB	Free_MB	Req_mir_free_MB	Usable_file_MB	Offline_disks	Voting_files	Name
MOUNTED	EXTERN	N	512	512	4096	1048576	1048575	1036823	0	1036823	0	N	DATA_DG/
MOUNTED	EXTERN	N	512	512	4096	1048576	255999	253665	0	253665	0	N	FRA_DG/
MOUNTED	EXTERN	N	512	512	4096	4194304	102396	102296	0	102296	0	N	MGMT_DATA/
MOUNTED	EXTERN	N	512	512	4096	1048576	255999	223714	0	223714	0	N	RMAN_DATA_DG/

grid@oracle19c-ol8-vvol-rman:+ASM:/home/grid>

oracle@oracle19c-ol8-vvol-rman:rmandb:/home/oracle> export ORACLE_SID=vvol19c

oracle@oracle19c-ol8-vvol-rman:vvol19c:/home/oracle> sqlplus / as sysdba
SQL*Plus: Release 19.0.0.0.0 - Production on Wed Dec 30 10:21:16 2020
Version 19.8.0.0.0
Copyright (c) 1982, 2020, Oracle. All rights reserved.
Connected to an idle instance.

SQL> startup mount pfile=/u01/app/oracle/product/19.0.0/dbhome_1/dbs/initvvol19c.ora
ORACLE instance started.
Total System Global Area   1.2885E+10 bytes
Fixed Size   12691536 bytes
Variable Size   1811939328 bytes
Database Buffers   1.1039E+10 bytes
Redo Buffers   20860928 bytes
Database mounted.
SQL>

The newly mounted database **vvol19c** is now ready to be backed up via Oracle RMAN.

Some caveats to keep in mind when using **Overwrite Disk**:

– Ensure that the production database **vvol19c** VMDKs (DATA_01 and FRA_01) match in size with the newly added VMDKs of the RMAN VM **Oracle19c-OL8-RMAN** in order to use **Overwrite Disk**.

– RMAN VM **Oracle19c-OL8-RMAN** requires that target disks be pre-configured for **Overwrite Disk**, which may result in consumption of storage space.

Oracle Backup of RAC

A 2-node Oracle RAC **vvolrac** with RAC VMs **vvolrac1** and **vvolrac2** is used in this use case example.

As previously mentioned, in the case of an Oracle RAC cluster, independent-persistent disk mode is not required to enable multi-writer for shared VMDKs. However, the default dependent disk mode causes a “cannot snapshot shared disk” error when a VM-level snapshot is taken of an Oracle RAC VM.

Use of independent-persistent disk mode allows taking a snapshot of the non-shared disk or disks (e.g., OS, Oracle binaries, standalone file system), while the shared disk(s) are backed up separately via a VMware-level snapshot mechanism (e.g., vVol-level backup of the shared VMDKs).

In the case of an Oracle RAC cluster, the snapshot process occurs in two steps:

– VM-level snapshot for non-shared VMDKs with disk mode set to **Dependent** for all RAC VMs vvolrac1 and vvolrac2

– Storage-based snapshot for shared VMDKs with disk mode set to **Independent-Persistent** from any RAC VM (e.g., **vvolrac1**).

The snapshot details of a 2-node Oracle 19c RAC cluster with VMs **vvolrac1** and **vvolrac2** follow with a VM-level snapshot of RAC VM **vvolrac1** shown immediately below:

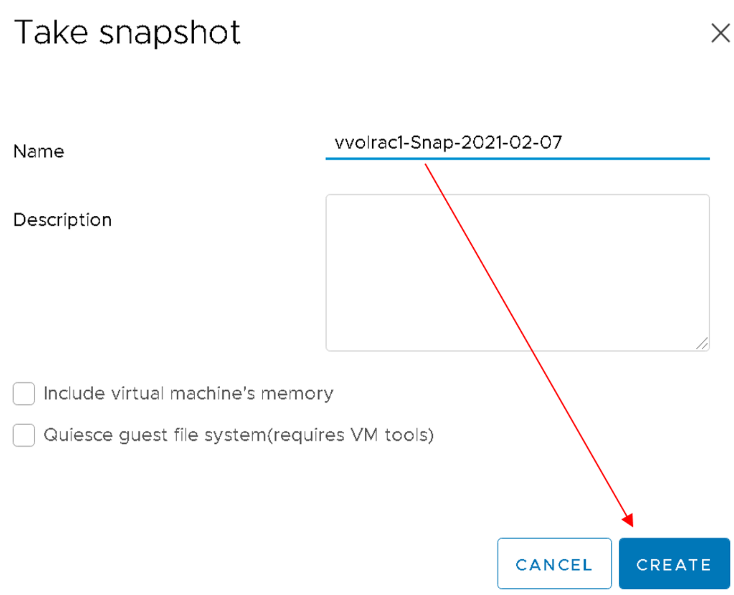


FIGURE 61. TAKING A RAC VM-LEVEL SNAPSHOT

The VM-level snapshot has been successfully taken for production of RAC VM **vvolrac1**.

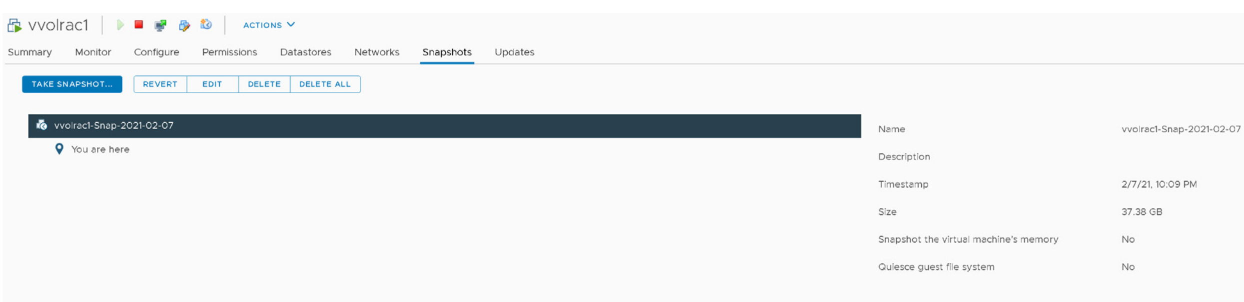


FIGURE 62. RAC VM VVOLRAC1 SNAPSHOT

Snapshot of the RAC VM **vvolrac2** is taken using the same process shown above.

The VM-level snapshot has been successfully taken for production of RAC VM **vvolrac2**.

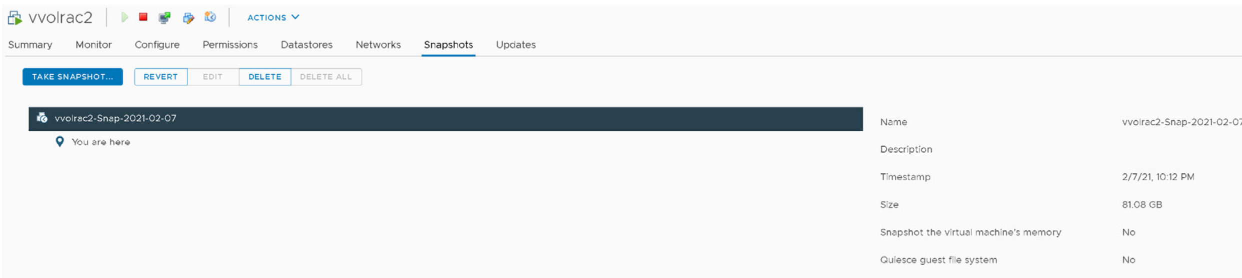


FIGURE 63. RAC VM VVOLRAC2 SNAPSHOT

Logging into the Pure Storage GUI indicates snapshots are created for all both RAC VM non-shared VMDKs.

Details of RAC VM **vvolrac1** snapshot are shown below:

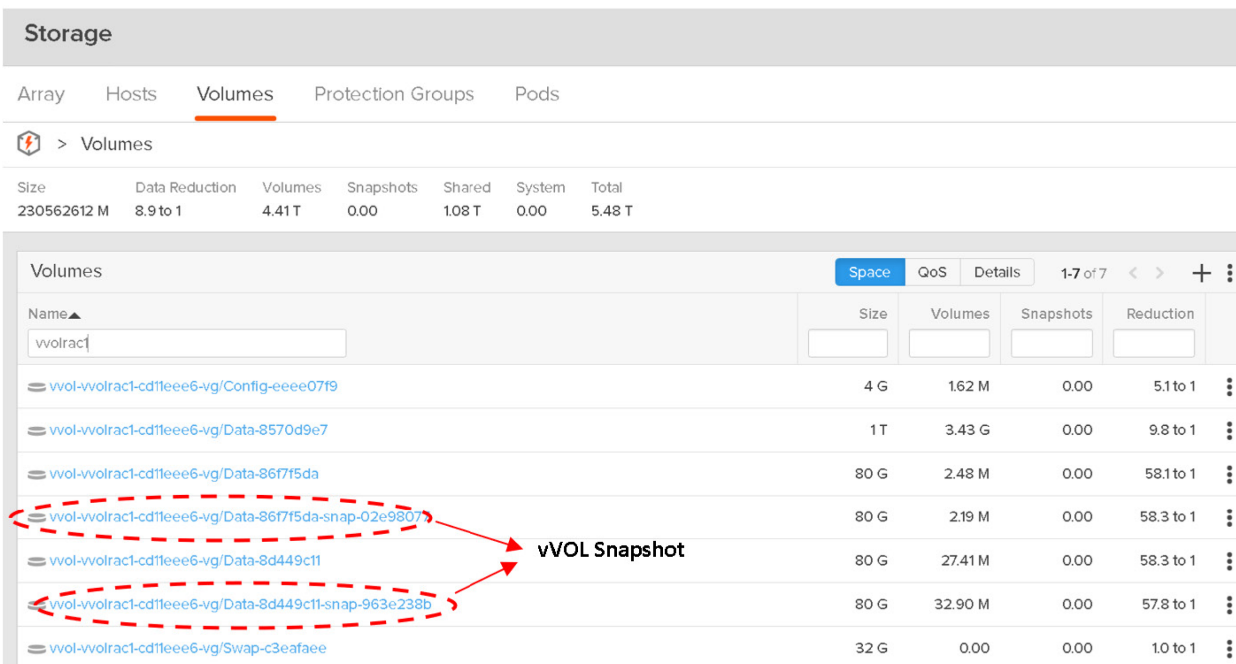


FIGURE 64. PURE STORAGE GUI – RAC VM VVOLRAC1 VOLUMES

Details of RAC VM vvolrac2 snapshot are shown below:

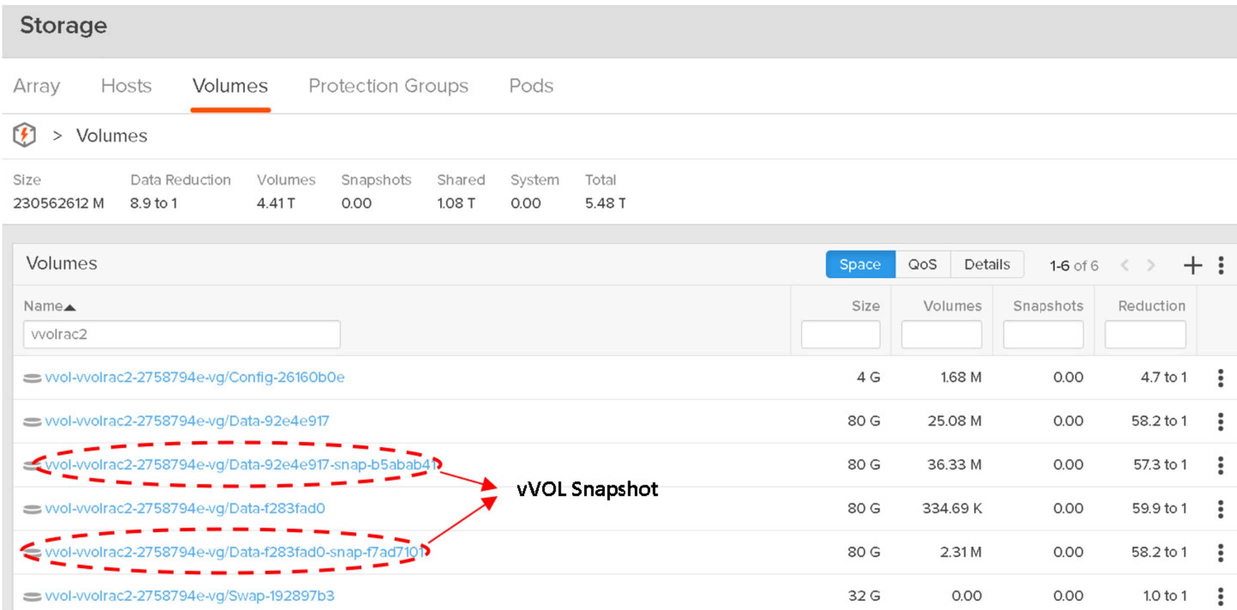


FIGURE 65. PURE STORAGE GUI – RAC VM VVOLRAC2 VOLUMES

Notice that the shared VMDK (1TB) does not have a snapshot associated with it.

A storage-based vVol snapshot is then taken within the Pure GUI of the database VMDKs. Click **Create** to produce the snapshot **vvolrac-DBSnap**.

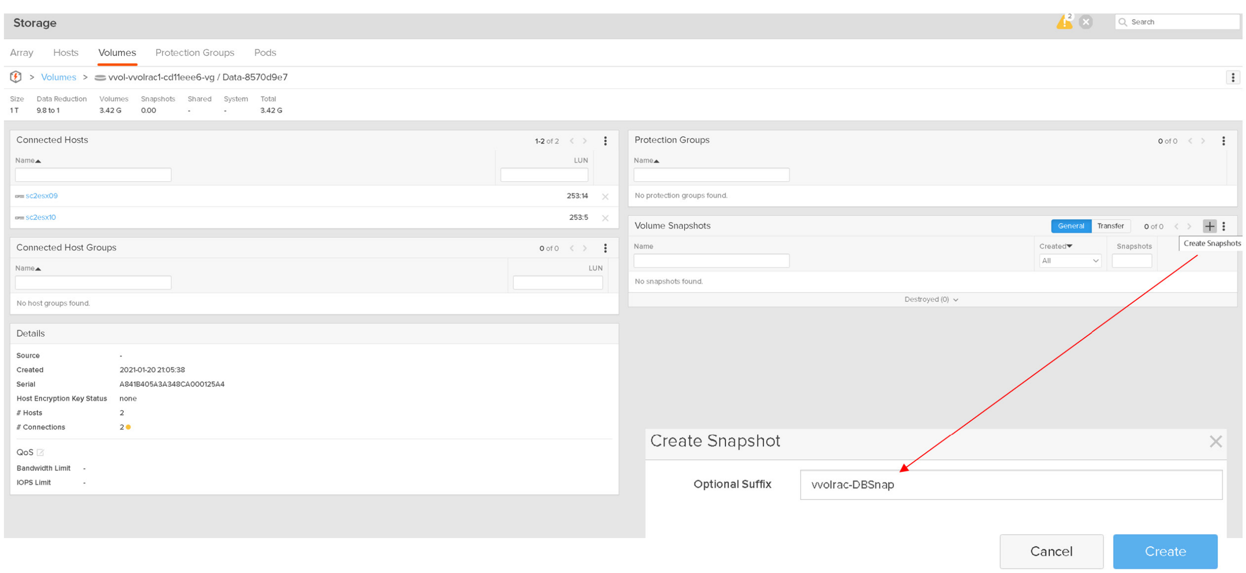


FIGURE 66. STORAGE VVOL-BASED SNAPSHOT CREATION PROCESS

vVol-based snapshot **vvolrac-DBSnap** is created for the database VMDKs.

FIGURE 67. STORAGE VVOL-BASED SNAPSHOT

Follow the same steps described above to import/overwrite the RAC database VMDKs on the RMAN VM **Oracle19c-OL8-RMAN** and back up the RAC database via Oracle RMAN.

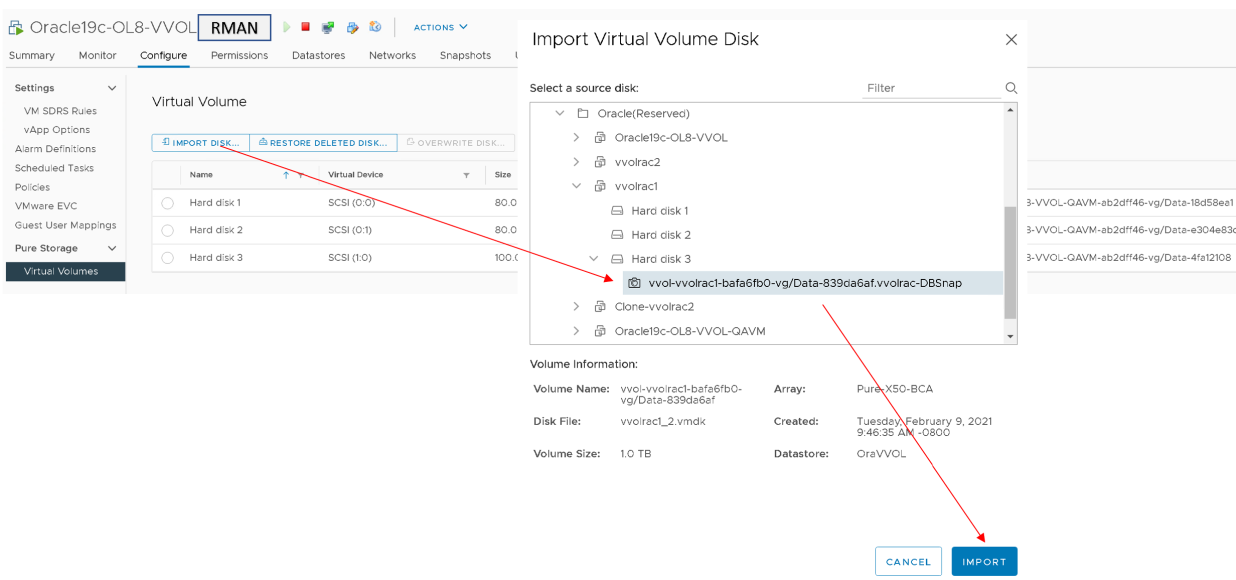


FIGURE 68. PURE PLUGIN – IMPORT DISK

The RAC database VMDKs are now present on the RMAN VM **Oracle19c-OL8-RMAN** and ready to be backup up via **Oracle RMAN**.

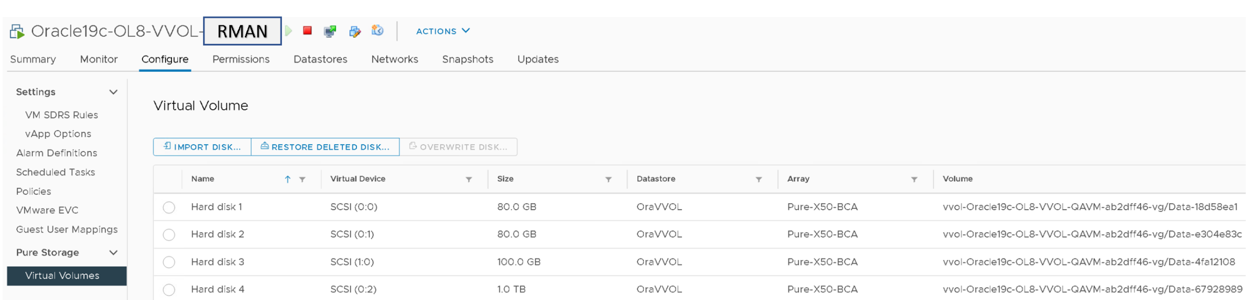


FIGURE 69. RMAN VM ORACLE19C-OL8-RMAN VMDK

Delete the vVol-based snapshot **vvolrac-DBSnap** after the disk import is completed as part of cleanup.

Oracle Restore and Recovery

This section validates database restore using VMware vVol-based snapshots of an Oracle single-instance and Oracle RAC on vVols using Pure x50 Storage.

In the event a database restore and recovery is required, one could use:

1. Oracle database restore using RMAN backup (this backup was created in the previous Oracle Backup section) followed by database-level recovery using Oracle
2. Restore of database vVol followed by database-level recovery using Oracle

Database restore could involve one of two scenarios:

1. A full database restore is required, in which case the production database is shut down before the contents of the database VMDKs are restored form VMware vVol snapshot using Pure Plugin **Overwrite Disk**
2. A partial database restore is required, in which case the production database is shut down and new VMDKs are imported from VMware vVol snapshot using the Pure Plugin **Import Disk**. We can now change the database name and/or database ID using Oracle MySupport Note *How to Change the DBID, DBNAME Using NID Utility (Doc ID 863800.1).* We can then use Oracle to copy the data from the new imported database over to the production database.

This section examines:

- vVol-level restore of the database VMDKs followed by database-level recovery using Oracle.
- Full database restore from a VMware vVol snapshot using Pure Plugin **Overwrite Disk**

Oracle Restore and Recovery of Single Instance

Ensure that a VM-level snapshot is taken for production VM **Oracle19c-OL8-VVOL** before beginning any database operations.

VM-level snapshot **VMSnap-2021-02-08** is shown below:

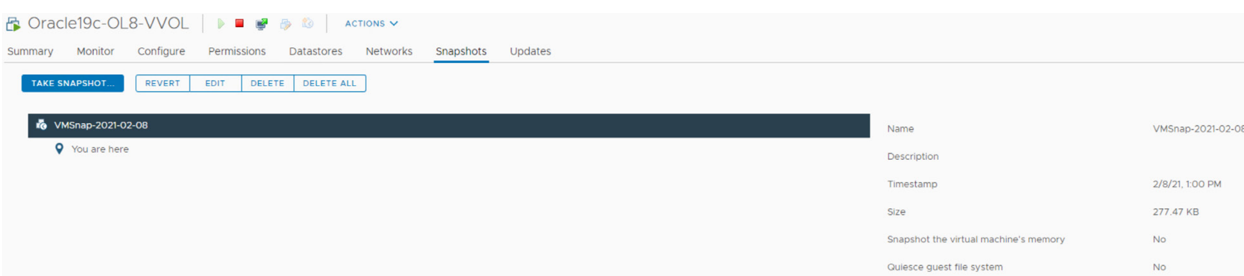


FIGURE 70. VM ORACLE19C-OL8-VVOL VM SNAPSHOT

In order to restore the database on production VM Oracle19c-OL8-VVOL from a VMware vVol snapshot, the production database is shut down cleanly first before the contents of the database VMDKs are restored using Pure Plugin **Overwrite Disk**.

After the production database is shut down, select the VMDKs to be restored and choose Pure Plugin **Overwrite Disk**.

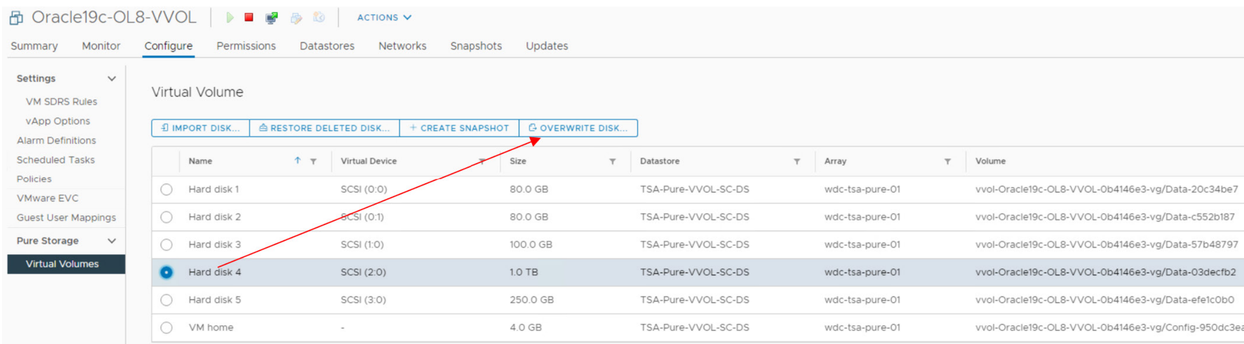


FIGURE 71. PURE PLUGIN – OVERWRITE DISK

Select the VMDKs snapshot and click **Overwrite**.

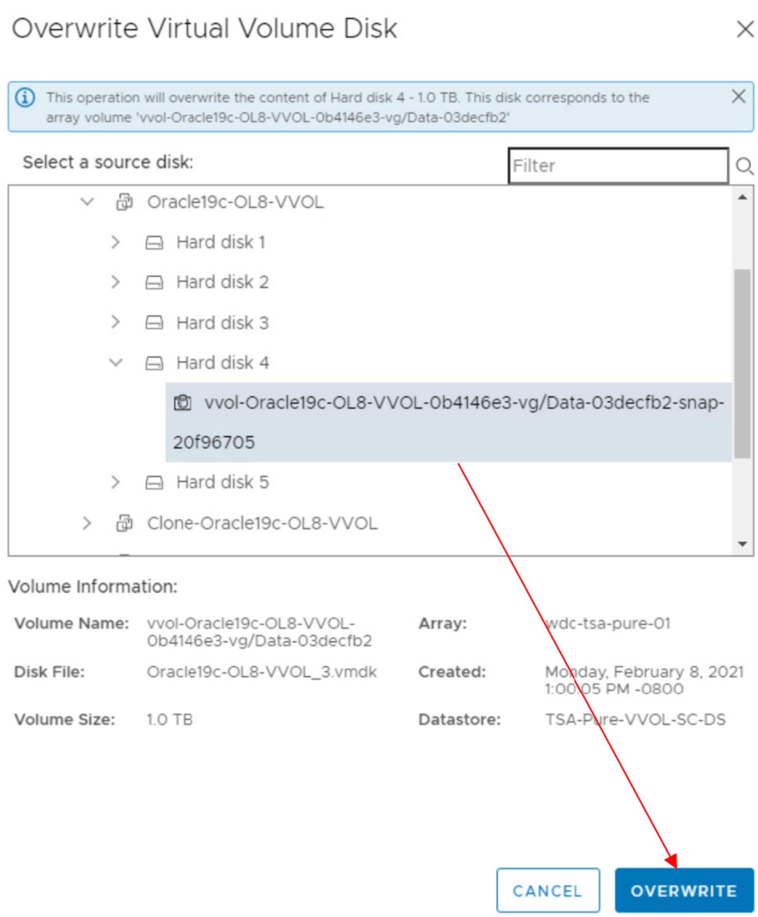


FIGURE 72. OVERWRITE VMDK WITH SNAPSHOT

Delete the VM-level snapshot **VMSnap-2021-02-08** after the disk overwrite operation completes. Login to the VM and follow the Oracle database recovery steps to recover the database if needed.

Oracle Restore and Recovery of RAC

Ensure that storage vVol-level snapshot **vvolrac-DBSnap** is taken from the Pure GUI for production RAC **vvolrac** database VMDKs before beginning any database operations.

It's recommended to shut the RAC cluster down in order to get a clean, high-quality snapshot of the database VMDKs.

A vVol-based snapshot **vvolrac-DBSnap** is created for the database VMDKs.

FIGURE 73. STORAGE VVOL-BASED SNAPSHOT VVOLRAC-DBSNAP

Proceed with planned database operations.

In the event a database restore is needed, before restoring the RAC database **vvolrac** from the storage vVol snapshot, shut down the RAC cluster. Cleanly shutting down RAC VMs **vvolrac1** and **vvolrac2** is recommended before proceeding.

After the production database is shut down, click on the database vVol snapshot and select **Restore**.

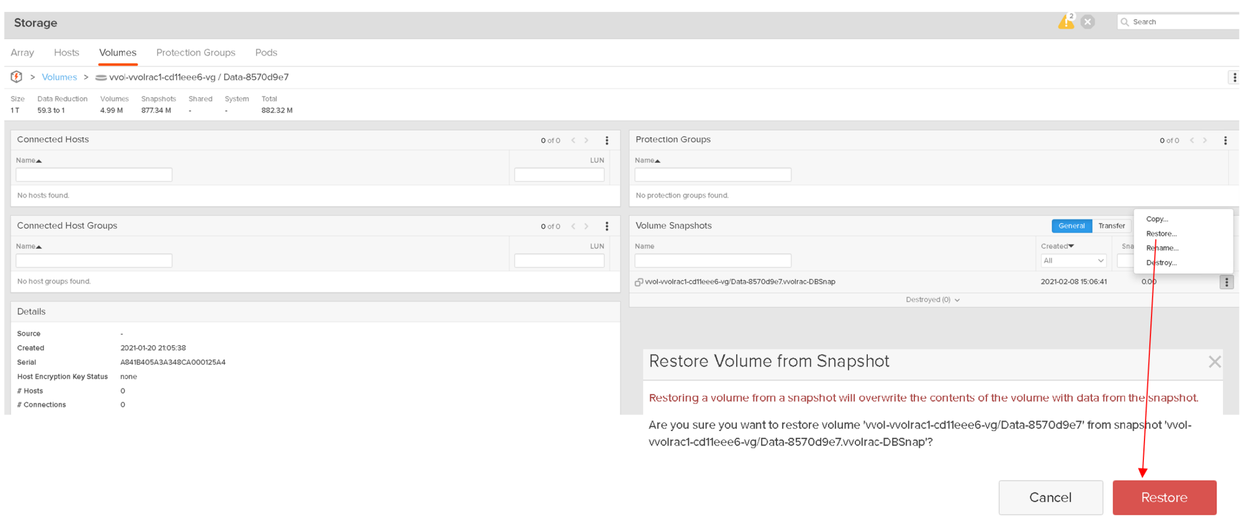


FIGURE 74. RESTORE RAC DATABASE VMDKS FROM STORAGE SNAPSHOT VVOLRAC-DBSNAP

The database VMDKs are now restored from the vVol-based snapshot **vvolrac-DBSnap**, taken before the start of the database operations.

Power on the RAC VM **vvolrac1** and **vvolrac2**. Start the RAC cluster if needed and check RAC services. The RAC cluster is up. Delete the vVol snapshot **vvolrac-DBSnap** as part of the cleanup.

RAC cluster **vvolrac** service is successfully up after the restore.

```
[root@vvolrac1 ~]# /u01/app/19.0.0/grid/bin/crsctl status res -t
--------------------------------------------------------------------------------
Name           Target  State        Server                   State details
--------------------------------------------------------------------------------
Local Resources
--------------------------------------------------------------------------------
ora.LISTENER.lsnr
               ONLINE  ONLINE       vvolrac1                 STABLE
               ONLINE  ONLINE       vvolrac2                 STABLE
ora.chad
               ONLINE  ONLINE       vvolrac1                 STABLE
               ONLINE  ONLINE       vvolrac2                 STABLE
ora.net1.network
               ONLINE  ONLINE       vvolrac1                 STABLE
               ONLINE  ONLINE       vvolrac2                 STABLE
ora.ons
               ONLINE  ONLINE       vvolrac1                 STABLE
               ONLINE  ONLINE       vvolrac2                 STABLE
--------------------------------------------------------------------------------
Cluster Resources
--------------------------------------------------------------------------------
ora.ASMNET1LSNR_ASM.lsnr(ora.asmgroup)
      1        ONLINE  ONLINE       vvolrac1                 STABLE
      2        ONLINE  ONLINE       vvolrac2                 STABLE
      3        ONLINE  OFFLINE                               STABLE
ora.DATA_DG.dg(ora.asmgroup)
      1        ONLINE  ONLINE       vvolrac1                 STABLE
      2        ONLINE  ONLINE       vvolrac2                 STABLE
      3        OFFLINE OFFLINE                               STABLE
ora.LISTENER_SCAN1.lsnr
      1        ONLINE  ONLINE       vvolrac1                 STABLE
ora.LISTENER_SCAN2.lsnr
      1        ONLINE  ONLINE       vvolrac2                 STABLE
ora.LISTENER_SCAN3.lsnr
      1        ONLINE  ONLINE       vvolrac2                 STABLE
ora.MGMTLSNR
      1        ONLINE  ONLINE       vvolrac2                 169.254.25.178 192.1
                                                             68.14.242,STABLE
ora.asm(ora.asmgroup)
      1        ONLINE  ONLINE       vvolrac1                 Started,STABLE
      2        ONLINE  ONLINE       vvolrac2                 Started,STABLE
      3        OFFLINE OFFLINE                               STABLE
ora.asmnet1.asmnetwork(ora.asmgroup)
      1        ONLINE  ONLINE       vvolrac1                 STABLE
      2        ONLINE  ONLINE       vvolrac2                 STABLE
      3        OFFLINE OFFLINE                               STABLE
ora.cvu
      1        ONLINE  ONLINE       vvolrac2                 STABLE
ora.mgmtdb
      1        ONLINE  ONLINE       vvolrac2                 Open,STABLE
ora.qosmserver
      1        ONLINE  ONLINE       vvolrac2                 STABLE
ora.scan1.vip
      1        ONLINE  ONLINE       vvolrac1                 STABLE
ora.scan2.vip
      1        ONLINE  ONLINE       vvolrac2                 STABLE
ora.scan3.vip
      1        ONLINE  ONLINE       vvolrac2                 STABLE
ora.vvolrac.db
      1        ONLINE  ONLINE       vvolrac1                 Open,HOME=/u01/app/o
                                                             racle/product/19.0.0
                                                             /dbhome_1,STABLE
      2        ONLINE  ONLINE       vvolrac2                 Open,HOME=/u01/app/o
                                                             racle/product/19.0.0
                                                             /dbhome_1,STABLE
ora.vvolrac1.vip
      1        ONLINE  ONLINE       vvolrac1                 STABLE
ora.vvolrac2.vip
      1        ONLINE  ONLINE       vvolrac2                 STABLE
--------------------------------------------------------------------------------
[root@vvolrac1 ~]#
```

FIGURE 75. RAC DATABASE VVOLRAC SERVICES

Oracle Database Cloning

This section validates database cloning using VMware snapshots of an Oracle single-instance and Oracle RAC on vVols using Pure x50 Storage.

Oracle Single-Instance Database

Two VMs are utilized for this use case:

- Production VM **Oracle19c-OL8-VVOL**
- QA VM **Oracle19c-OL8-VVOL-QAVM**

This use case focuses on cloning an existing Oracle database **vvol19c** to a QA VM **Oracle19c-OL8-QAVM** for a variety of database purposes (e.g., testing a new version of the software code, testing database patching).

The following steps were used for this testing:

1. Clone the DATA_DG and FRA_DG ASM disks of the production database **vvol19c**. In production VM **Oracle19c-OL8-VVOL**, the database data and redo log files were on ASM disk group DATA_DG, the archive log files were on ASM disk group FRA_DG.

   We can clone the production database **vvol19c** disks in two ways:

   - Take a snapshot of the database, either a traditional VM-level snapshot using the web client or a VMDK-level snapshot of DATA_DG and FRA_DG VMDKs using the Pure storage Plugin, then create clones of the two VMDKs from the above snapshot. For example, the Pure Plugin **Import Disk** option will indicate that the VMDK is being cloned from a snapshot.

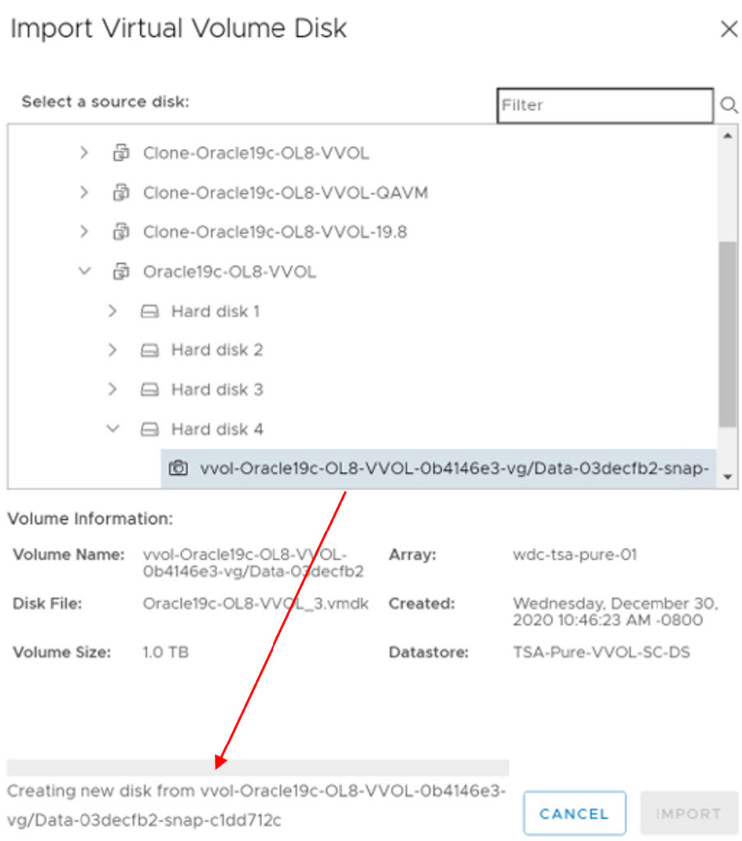


FIGURE 76. PURE STORAGE PLUGIN – IMPORT DISK

– Clone the DATA_DG and FRA_DG VMDKs directly without taking a snapshot. For example, the Pure Plugin **Import Disk** option will indicate that the VMDK is being cloned from the source VMDK directly.

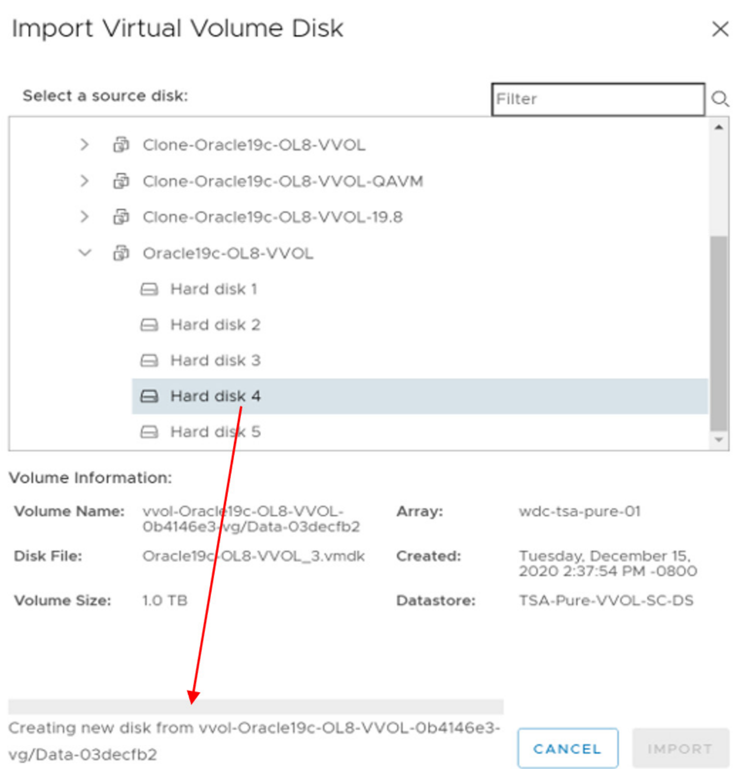


FIGURE 77. PURE STORAGE PLUGIN – IMPORT DISK

For production database **vvol19c**, all redo log files, controlfiles and database files are part of the **DATA_DG ASM** disk group, so this use case used a VMDK-level snapshot for **DATA_DG** and **FRA_DG** VMDKs.

2. Take VMDK-level snapshots for DATA_DG and FRA_DG VMDKs on production VM **Oracle19c-OL8-VVOL** as shown below:

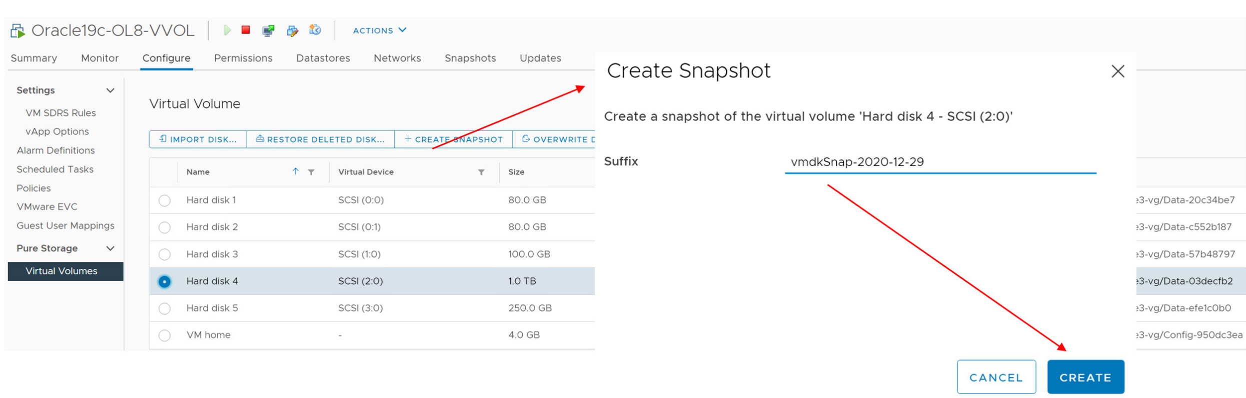


FIGURE 78. DATA_DG VMDK-LEVEL SNAPSHOT

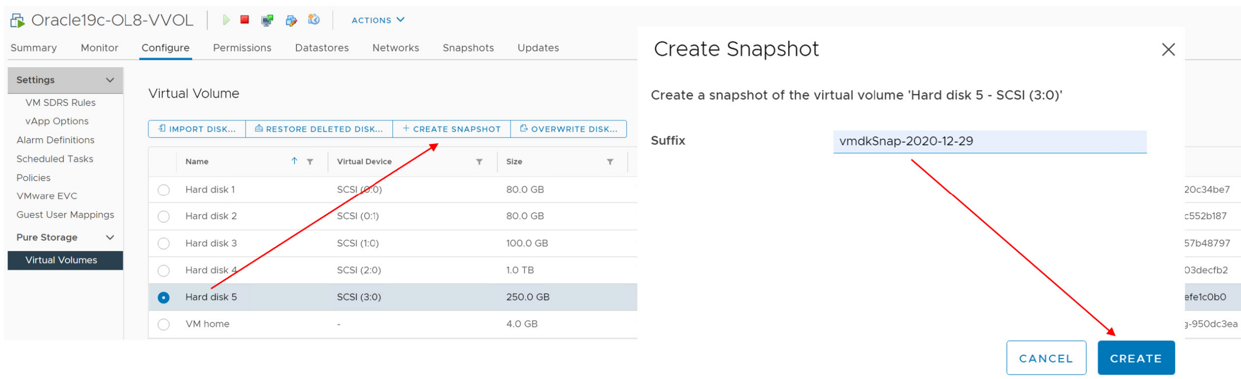


FIGURE 79. FRA_DG VMDK-LEVEL SNAPSHOT

The Pure Storage GUI shows the snapshots at the vVol level:

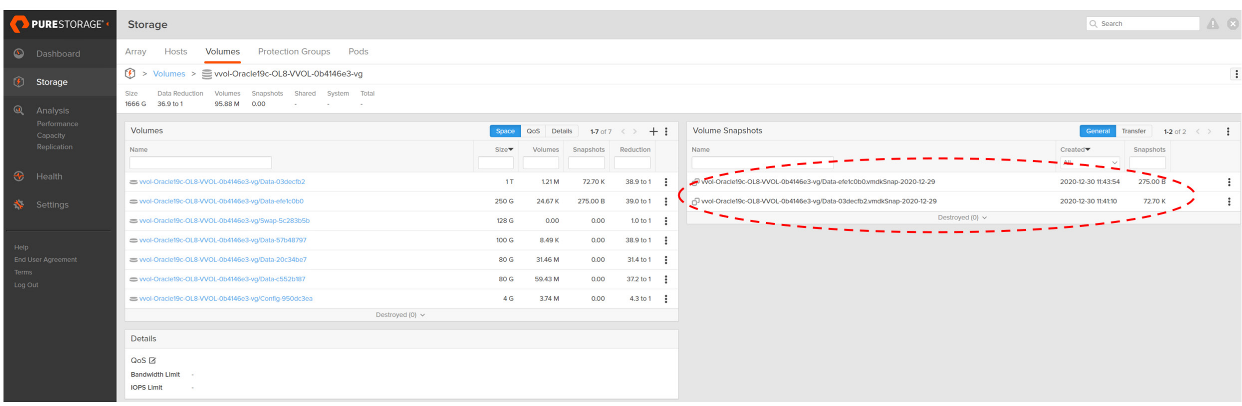


FIGURE 80. PURE STORAGE GUI – VOLUMES

3. On VM **Oracle19c-OL8-QAVM**, using the Pure Plugin, choose **Import Disk** and Import DATA_DG ASM disk (1TB).

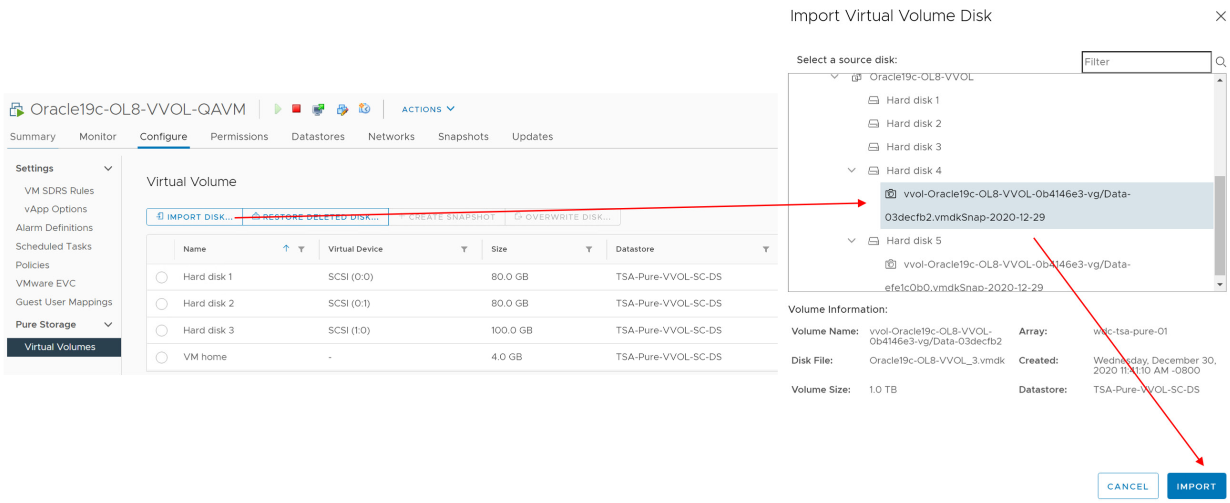


FIGURE 81. PURE STORAGE PLUGIN – IMPORT DISK

On VM **Oracle19c-OL8-QAVM**, using the Pure Plugin, choose **Import Disk** and Import FRA_DG ASM disk (250GB).

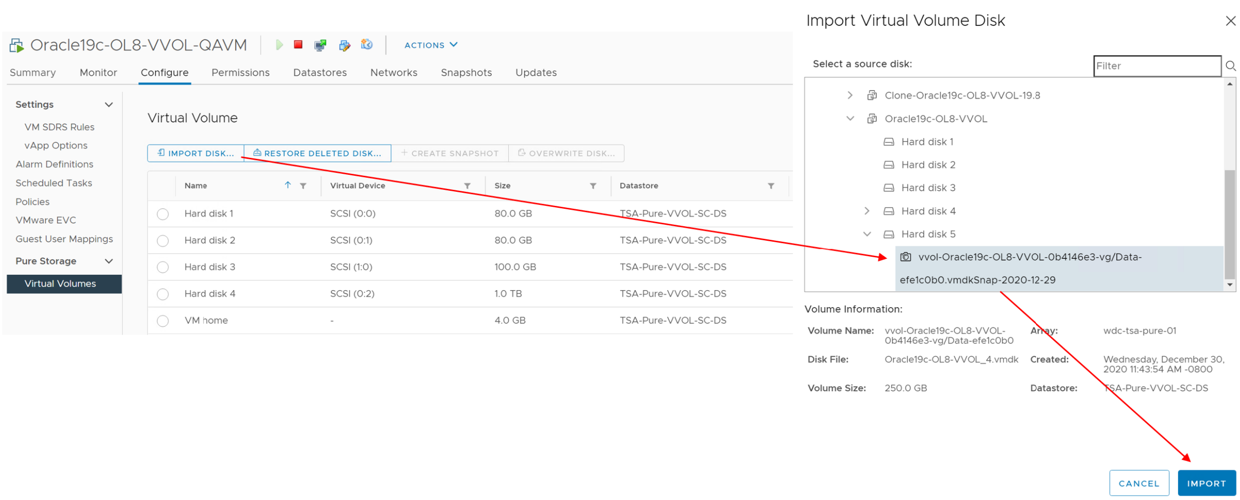


FIGURE 82. PURE STORAGE PLUGIN – IMPORT DISK

VM **Oracle19c-OL8-QAVM** now has two new VMDKs: DATA_DG ASM disk (1TB) and FRA_DG ASM disk (250GB).

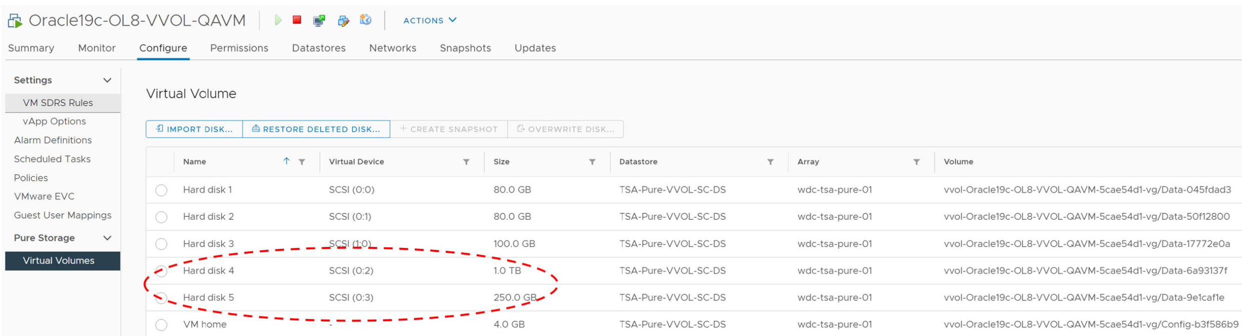


FIGURE 83. VM ORACLE19C-OL8-QAVM VMDKS

4. Delete the snapshot taken of the VMDKs from the Pure Storage GUI as shown below:

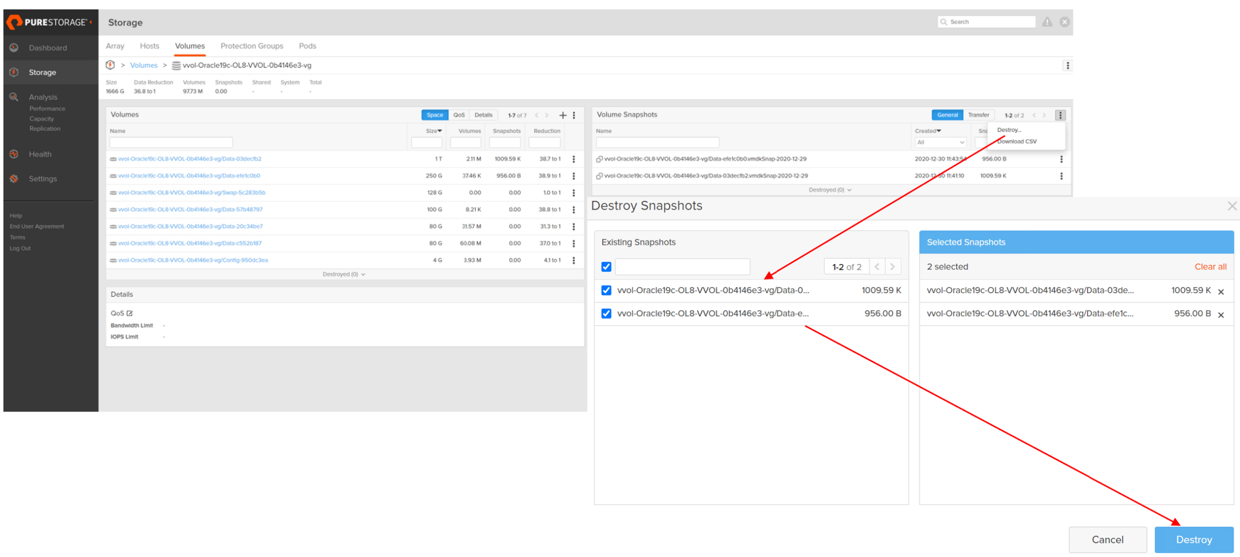


FIGURE 84. DELETING A SNAPSHOT IN PURE STORAGE GUI

5. On QA VM **Oracle19c-OL8-QAVM**, scan for new ASM disks using the Oracle **oracleasm** command to discover new VMDKs **DATA_01** and **FRA_01** of production database vvol19c.

```
[root@oracle19c-ol8-vvol-qavm ~]# oracleasm scandisks ; oracleasm listdisks
Reloading disk partitions: done
Cleaning any stale ASM disks...
Scanning system for ASM disks...
Instantiating disk "DATA_01"
Instantiating disk "FRA_01"
DATA_01
FRA_01
MGMT_DATA01
[root@oracle19c-ol8-vvol-qavm ~]#
```

Mount the new disk groups **+DATA_DG** and **+FRA_DG**

```
grid@oracle19c-ol8-vvol-qavm:+ASM:/home/grid> asmcmd mount DATA_DG
grid@oracle19c-ol8-vvol-qavm:+ASM:/home/grid> asmcmd mount FRA_DG
```

```
grid@oracle19c-ol8-vvol-qavm:+ASM:/home/grid> asmcmd lsdg
```

State	Type	Rebal	Sector	Logical_Sector	Block	AU	Total_MB	Free_MB	Req_mir_free_MB	Usable_file_MB	Offline_disks	Voting_files	Name
MOUNTED	EXTERN	N	512	512	4096	1048576	1048575	1036823	0	1036823	0	N	DATA_DG/
MOUNTED	EXTERN	N	512	512	4096	1048576	255999	253526	0	253526	0	N	FRA_DG/
MOUNTED	EXTERN	N	512	512	4096	4194304	102396	102296	0	102296	0	N	MGMT_DATA/

```
grid@oracle19c-ol8-vvol-qavm:+ASM:/home/grid>
```

6. On the QA VM **Oracle19c-OL8-QAVM**, start up the database **vvol19c**.

Make appropriate changes to the database **vvol19c** init.ora file for memory and file directory setings.

Copy the **orapwvvol19c** file from Production VM **Oracle19c-OL8 VVOL** to QA VM **Oracle19c-OL8-QAVM**.

```
oracle@oracle19c-ol8-vvol-qavm:vvol19c:/home/oracle> export ORACLE_SID=vvol19c

oracle@oracle19c-ol8-vvol-qavm:vvol19c:/home/oracle> sqlplus / as sysdba
SQL*Plus: Release 19.0.0.0.0 - Production on Wed Dec 30 12:11:44 2020
Version 19.8.0.0.0
Copyright (c) 1982, 2020, Oracle. All rights reserved.

Connected to an idle instance.

SQL> startup pfile=/u01/app/oracle/product/19.0.0/dbhome_1/dbs/initvvol19c.ora
ORACLE instance started.
Total System Global Area     3.4360E+10 bytes
Fixed Size                   12697696 bytes
Variable Size                3892314112 bytes
Database Buffers             3.0400E+10 bytes
Redo Buffers                 54407168 bytes
Database mounted.
Database opened.
SQL> Disconnected from Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.8.0.0.0
oracle@oracle19c-ol8-vvol-qavm:vvol19c:/home/oracle>
```

7. We can now change the database name and/or database ID using Oracle MySupport Note *How to Change the DBID, DBNAME Using NID Utility (Doc ID 863800.1).*

8. The new database is now ready for a variety of database purposes (e.g., testing a new version of the software code, testing database patching).

Oracle RAC

A 2-node Oracle RAC **vvolrac** with RAC VMs **vvolrac1** and **vvolrac2** is used in the RAC cloning use case example.

As explained earlier, to clone RAC database VMDKs, it's recommended to take a consistent storage vVol-based snapshot **vvolrac-DBSnap** for shared VMDKs from the Pure GUI.

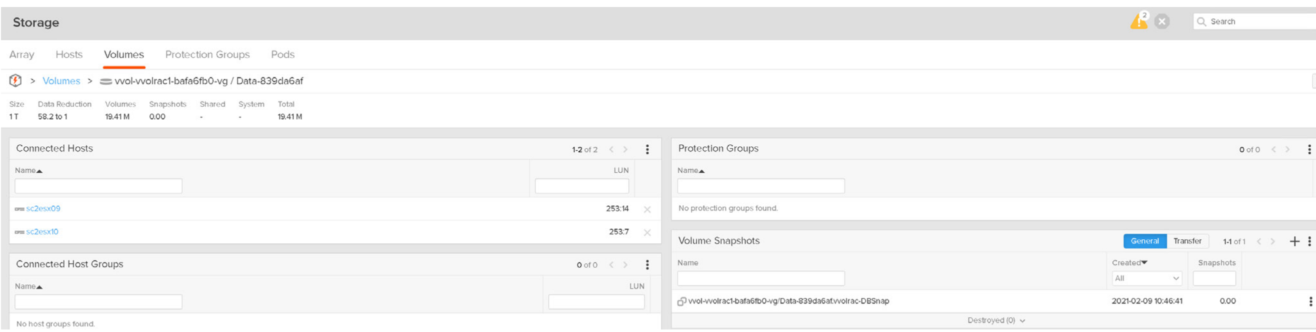


FIGURE 85. STORAGE VVOL SNAPSHOT OF RAC DATABASE VVOLRAC VMDKS

One possible requirement could be to simply clone the RAC database **vvolrac** and mount the database clone **vvolrac-db-clone** on a target VM to be used for database operations.

In this instance, use Pure Plugin **Import Disk** on VM **Oracle19c-OL8-QAVM** and import a new disk from the storage vVol-based snapshot **vvolrac-DBSnap**.

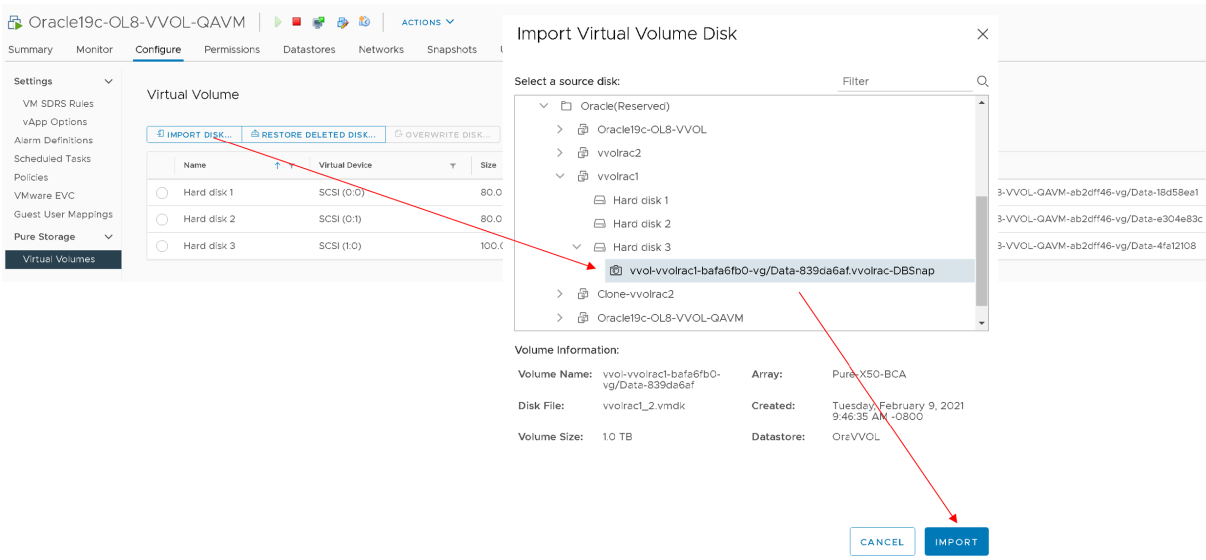


FIGURE 86. PURE PLUGIN – IMPORT DISK (VM ORACLE19C-OL8-QAVM)

For example, we have another RAC 19c cluster **rac19c** with 2 RAC nodes **rac19c1** and **rac19c2**.

Another possible requirement could be to clone the current RAC database **vvolrac** and add it as a new database to a RAC cluster **rac19c** with existing databases.

In this case, use Pure Plugin **Import Disk** to mount the RAC database **vvolrac** VMDKs on RAC VM **rac19c1**.

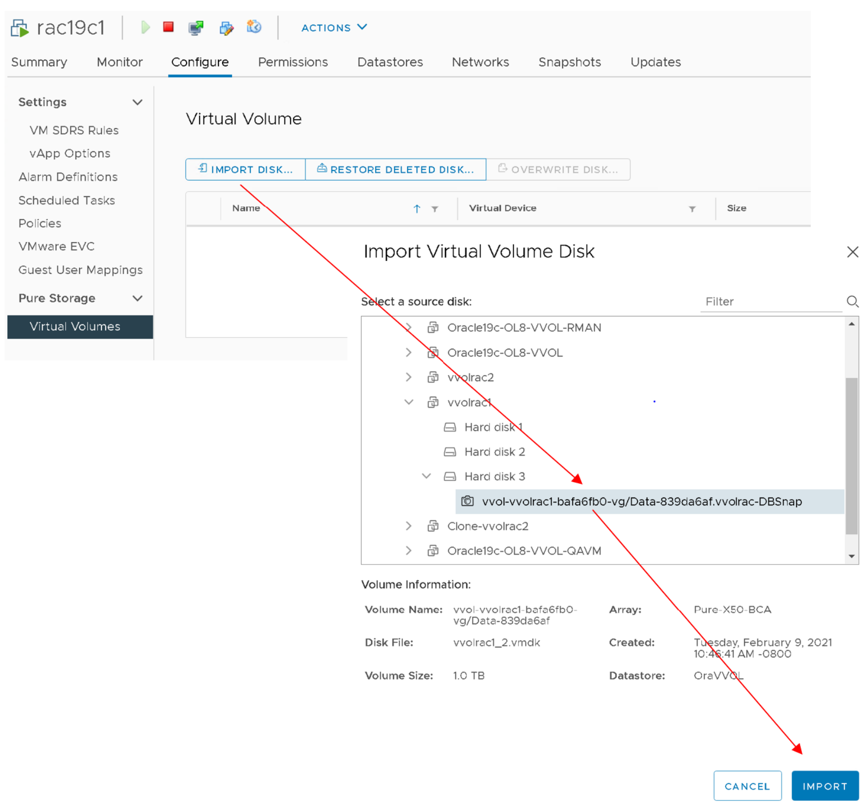


FIGURE 87. PURE PLUGIN – IMPORT DISK (RAC VM RAC19C1)

The RAC database **vvolrac** is now added to the RAC VM **rac19c1**.

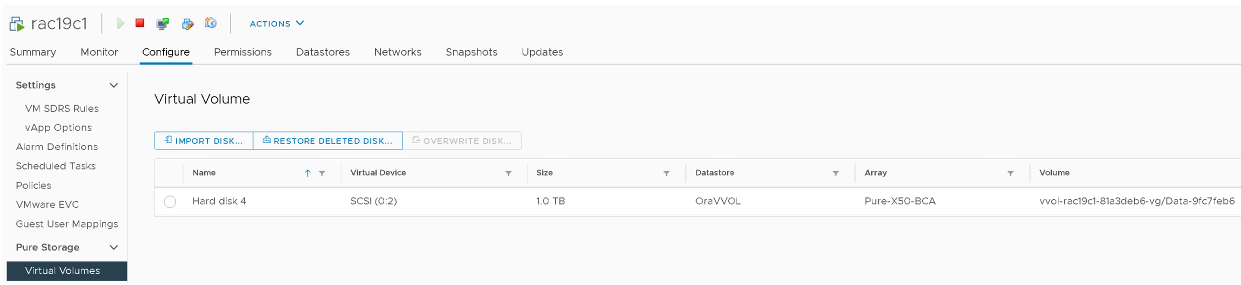


FIGURE 88. RAC VM RAC19C1 WITH VVOLRAC VMDKS

On RAC VM **rac19c1**, set the disk sharing to **Multi-writer**. The RAC node **rac19c2** then refers to the shared VMDKs on RAC VM **rac19c1** with disk sharing set to **Multi-writer**.

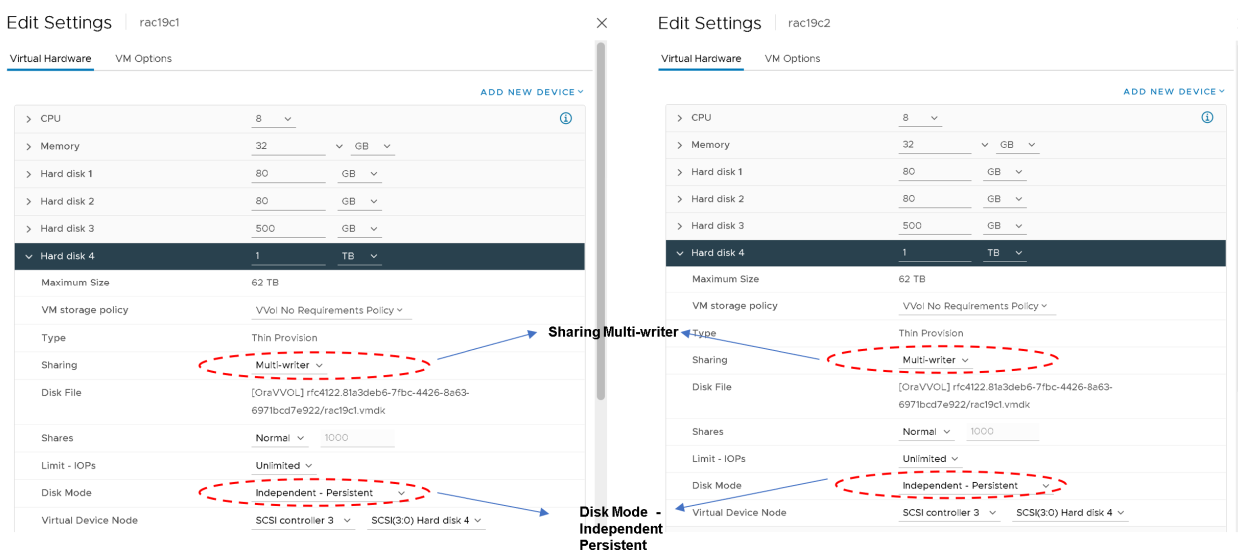


FIGURE 89. RAC VM RAC19C1 AND RAC19C2 WITH VVOLRAC DATABASE VMDKS SHARED WITH MULTI-WRITER

The subsequent steps to add the cloned database **vvolrac** to RAC cluster **rac19c** are outside the scope of this document.

Oracle Database Refresh

This section validates database refresh using VMware vVol-based snapshots of an Oracle single-instance and Oracle RAC using Pure x50 Storage.

Oracle Single Instance Database

Two VMs are utilized for this use case:

- Production VM **Oracle19c-OL8-VVOL**
- QA VM **Oracle19c-OL8-VVOL-QAVM**

Database refresh is typically the process of refreshing the contents of a development or testing database from a copy of the production database so that they have the latest data to develop or test against.

This use case focuses on refreshing an older version of database **vvol19c** running on **QA VM Oracle19c-OL8-QAVM** from the latest copy of the production database **vvol19c** running on production VM **Oracle19c-OL8-VVOL**.

1. On QA VM **Oracle19c-OL8-QAVM**:
   - Shutdown the running database **vvol19c**
   - Dismount the **vvol19c** ASM disk groups **DATA_DG** and **FRA_DG**

```
oracle@oracle19c-ol8-vvol-qavm:vvol19c:/home/oracle> sqlplus / as sysdba
SQL*Plus: Release 19.0.0.0.0 - Production on Wed Dec 30 12:33:04 2020
Version 19.8.0.0.0
Copyright (c) 1982, 2020, Oracle. All rights reserved.

Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.8.0.0.0

SQL> shutdown immediate;
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> exit
Disconnected from Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.8.0.0.0
oracle@oracle19c-ol8-vvol-qavm:vvol19c:/home/oracle>

grid@oracle19c-ol8-vvol-qavm:+ASM:/home/grid> asmcmd umount DATA_DG
grid@oracle19c-ol8-vvol-qavm:+ASM:/home/grid> asmcmd umount FRA_DG

grid@oracle19c-ol8-vvol-qavm:+ASM:/home/grid> asmcmd lsdg
```

State	Type	Rebal	Sector	Logical_Sector	Block	AU	Total_MB	Free_MB	Req_mir_free_MB	Usable_file_MB	Offline_disks	Voting_files	Name
MOUNTED	EXTERN	N	512	512	4096	4194304	102396	102296	0	102296	0	N	MGMT_DATA/

```
grid@oracle19c-ol8-vvol-qavm:+ASM:/home/grid>

[root@oracle19c-ol8-vvol-qavm ~]# oracleasm scandisks ; oracleasm listdisks
Reloading disk partitions: done
Cleaning any stale ASM disks...
Scanning system for ASM disks...
DATA_01
FRA_01
MGMT_DATA01
[root@oracle19c-ol8-vvol-qavm ~]#
```

2. We can use Pure Plugin **Overwrite Disk** to overwrite the older version of database **vvol19c** VMDKs on QA VM **Oracle19c-OL8-QAVM** from current production database vvol19c VMDKs running on production VM **Oracle19c-OL8-VVOL**.

Take VMDK-level snapshots for DATA_DG and FRA_DG VMDKs on production VM **Oracle19c-OL8-VVOL** as shown below:

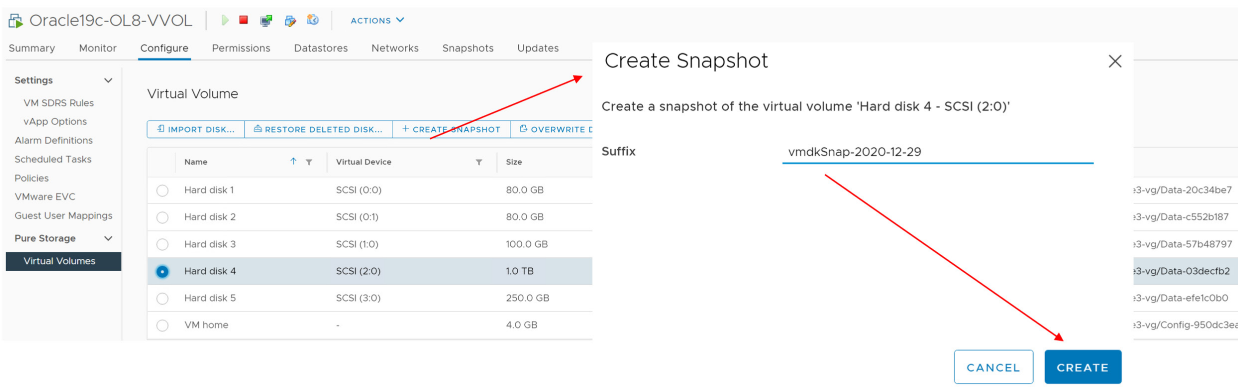


FIGURE 90. DATA_DG VMDK-LEVEL SNAPSHOT

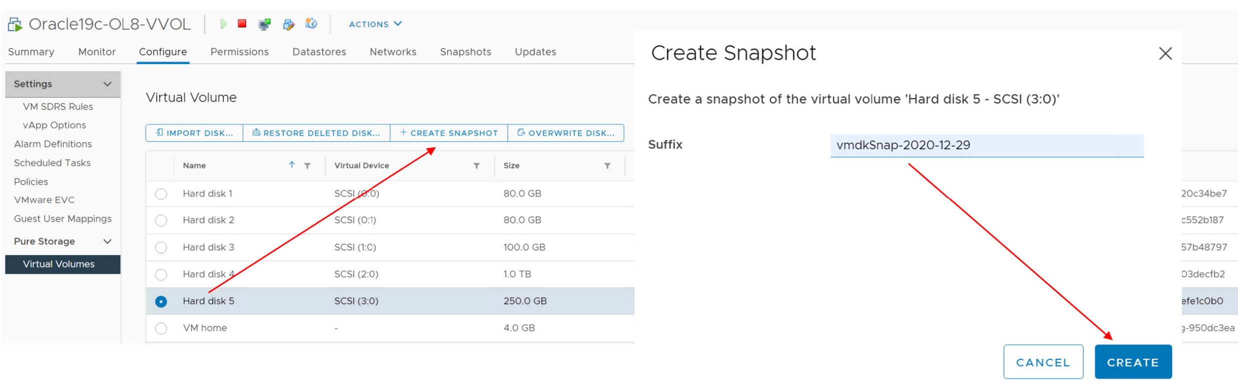


FIGURE 91. FRA_DG VMDK-LEVEL SNAPSHOT

VMDK snapshots can be viewed by logging into the Pure Storage GUI.

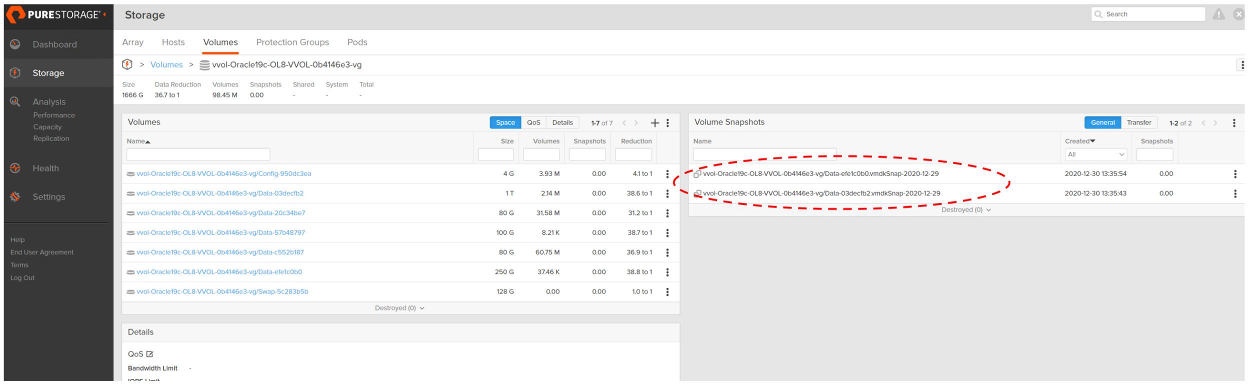


FIGURE 92. VIEWING SNAPSHOTS USING PURE STORAGE PLUGIN

3. On VM **Oracle19c-OL8-QAVM**, using the Pure Plugin, click on the DATA_DG VMDK (1TB), Choose **Overwrite Disk** and overwrite the current DATA_DG ASM disk from contents of the production VM **Oracle19c-OL8_VVOL**.

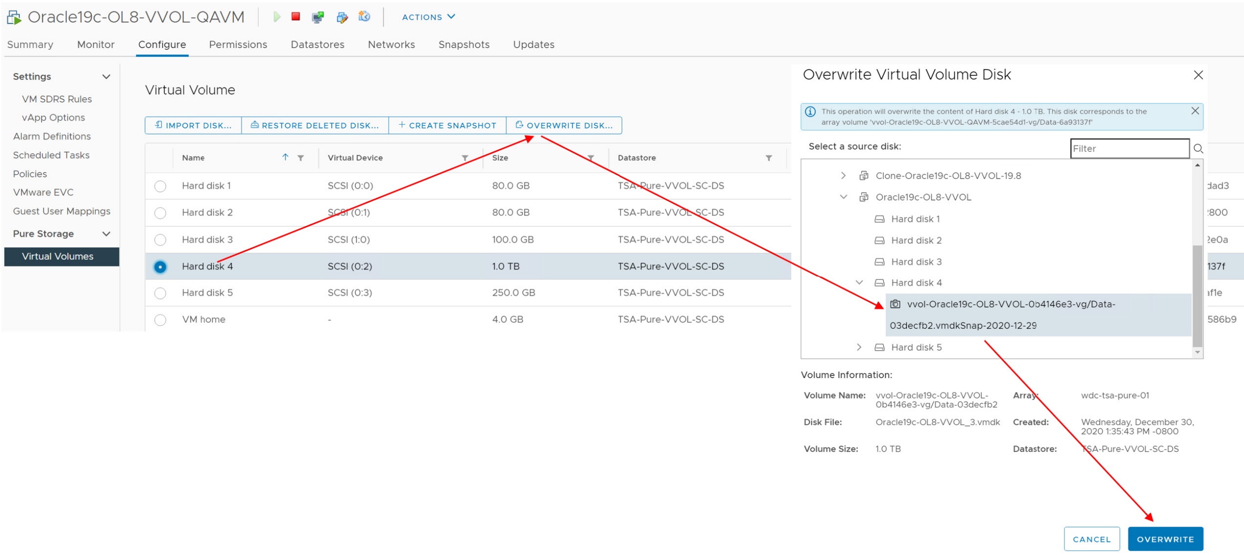


FIGURE 93. DISK OVERWRITE USING PURE STORAGE PLUGIN

On VM **Oracle19c-OL8-QAVM**, using the Pure Plugin, click on the FRA_DG VMDK (250GB), Choose **Overwrite Disk** overwrite the current FRA_DG ASM disk from contents of the production VM **Oracle19c-OL8_VVOL**.

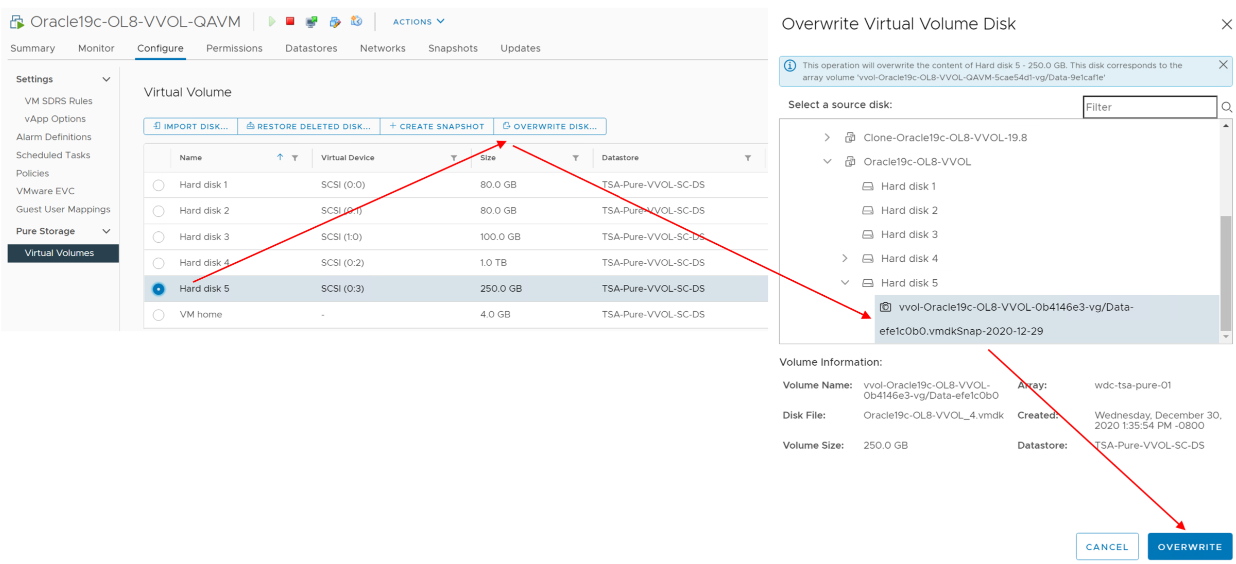


FIGURE 94. DISK OVERWRITE USING PURE STORAGE PLUGIN

VM **Oracle19c-OL8-QAVM** DATA_DG ASM disk and FRA_DG ASM disk have now been refreshed from production VM **Oracle19c-OL8-VVOL**.

4. Delete the snapshot taken of the VMDKs from the Pure Storage GUI as shown below:

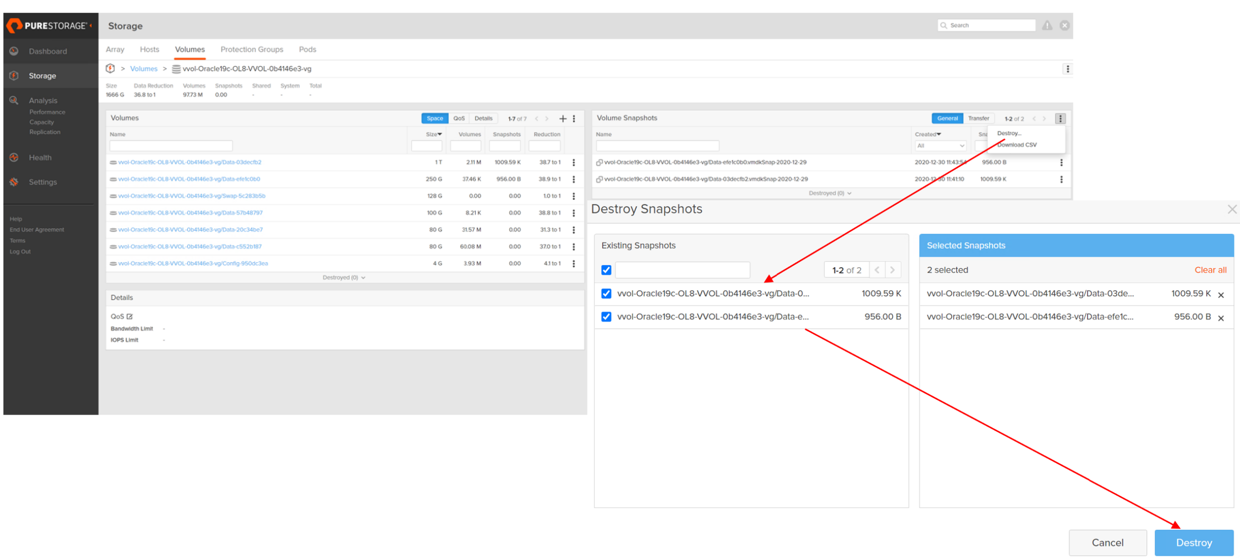


FIGURE 95. DELETING A SNAPSHOT IN PURE STORAGE GUI

5. On QA VM **Oracle19c-OL8-QAVM**, scan for ASM disks using the Oracle **oracleasm** command.

```
[root@oracle19c-ol8-vvol-qavm ~]# oracleasm scandisks ; oracleasm listdisks
Reloading disk partitions: done
Cleaning any stale ASM disks...
Scanning system for ASM disks...
Instantiating disk "DATA_01"
Instantiating disk "FRA_01"
DATA_01
FRA_01
MGMT_DATA01
[root@oracle19c-ol8-vvol-qavm ~]#
```

Mount the new disk groups **+DATA_DG** and **+FRA_DG**.

```
grid@oracle19c-ol8-vvol-qavm:+ASM:/home/grid> asmcmd mount DATA_DG
grid@oracle19c-ol8-vvol-qavm:+ASM:/home/grid> asmcmd mount FRA_DG
```

```
grid@oracle19c-ol8-vvol-qavm:+ASM:/home/grid> asmcmd lsdg
```

State	Type	Rebal	Sector	Logical_Sector	Block	AU	Total_MB	Free_MB	Req_mir_free_MB	Usable_file_MB	Offline_disks	Voting_files	Name
MOUNTED	EXTERN	N	512	512	4096	1048576	1048575	1036823	0	1036823	0	N	DATA_DG/
MOUNTED	EXTERN	N	512	512	4096	1048576	255999	253526	0	253526	0	N	FRA_DG/
MOUNTED	EXTERN	N	512	512	4096	4194304	102396	102296	0	102296	0	N	MGMT_DATA/

```
grid@oracle19c-ol8-vvol-qavm:+ASM:/home/grid>
```

6. On the QA VM **Oracle19c-OL8-QAVM**, start up the database **vvol19c**.

Make appropriate changes to the database vvol19c init.ora file for memory and file directory setings.

Copy the **orapwvvol19c** file from production VM **Oracle19c-OL8 VVOL** to QA VM **Oracle19c-OL8-QAVM**.

```
oracle@oracle19c-ol8-vvol-qavm:vvol19c:/home/oracle> export ORACLE_SID=vvol19c

oracle@oracle19c-ol8-vvol-qavm:vvol19c:/home/oracle> sqlplus / as sysdba
SQL*Plus: Release 19.0.0.0.0 - Production on Wed Dec 30 12:11:44 2020
Version 19.8.0.0.0
Copyright (c) 1982, 2020, Oracle. All rights reserved.

Connected to an idle instance.

SQL> startup pfile=/u01/app/oracle/product/19.0.0/dbhome_1/dbs/initvvol19c.ora
ORACLE instance started.
Total System Global Area      3.4360E+10 bytes
Fixed Size                    12697696 bytes
Variable Size                 3892314112 bytes
Database Buffers              3.0400E+10 bytes
Redo Buffers                  54407168 bytes
Database mounted.
Database opened.
SQL> Disconnected from Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.8.0.0.0
oracle@oracle19c-ol8-vvol-qavm:vvol19c:/home/oracle>
```

7. We can now change the database name and/or database ID using Oracle MySupport Note *How to Change the DBID, DBNAME Using NID Utility (Doc ID 863800.1).*

8. The new database is now ready for a variety of database purposes (e.g., testing a new version of the software code, testing database patching)

Oracle RAC

A 2-node Oracle RAC **vvolrac** with RAC VMs **vvolrac1** and **vvolrac2** is used in the RAC cloning use case example.

The steps to take a consistent storage vVol-based snapshot **vvolrac-DBSnap** for shared VMDKs from Pure GUI are the same as described in the above database cloning of Oracle RAC section.

In this case, use Pure Plugin **Overwrite Disk** on VM **Oracle19c-OL8-QAVM** and overwrite existing database VMDKs from the storage vVol-based snapshot **vvolrac-DBSnap**.

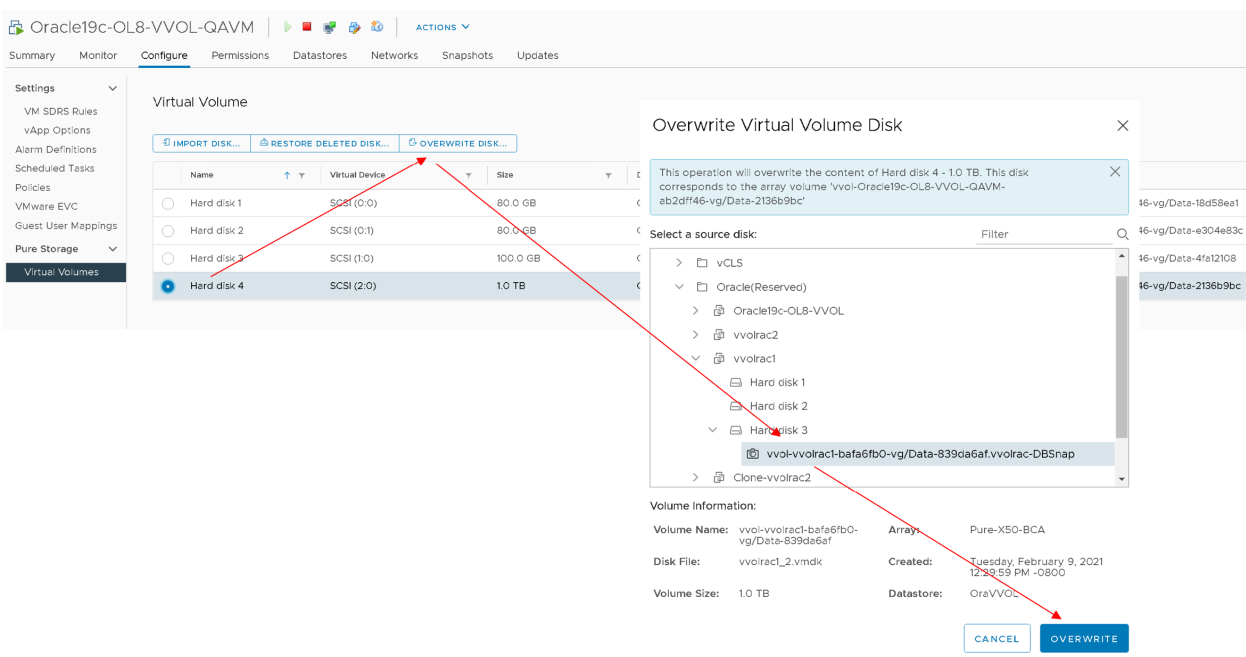


FIGURE 96. PURE PLUGIN – OVERWRITE DISK (VM ORACLE19C-OL8-QAVM)

For example, we have another RAC 19c cluster **rac19c** with 2 RAC nodes **rac19c1** and **rac19c2**.

Another requirement could be to refresh the contents of an existing RAC database in another RAC cluster rac19c from the current RAC database vvolrac.

Use the Pure Plugin **Overwrite Disk** to overwrite an RAC database in RAC cluster **rac19c** with the contents of the **vvolrac** database. Shuting down the target RAC database before it's refreshed is recommended.

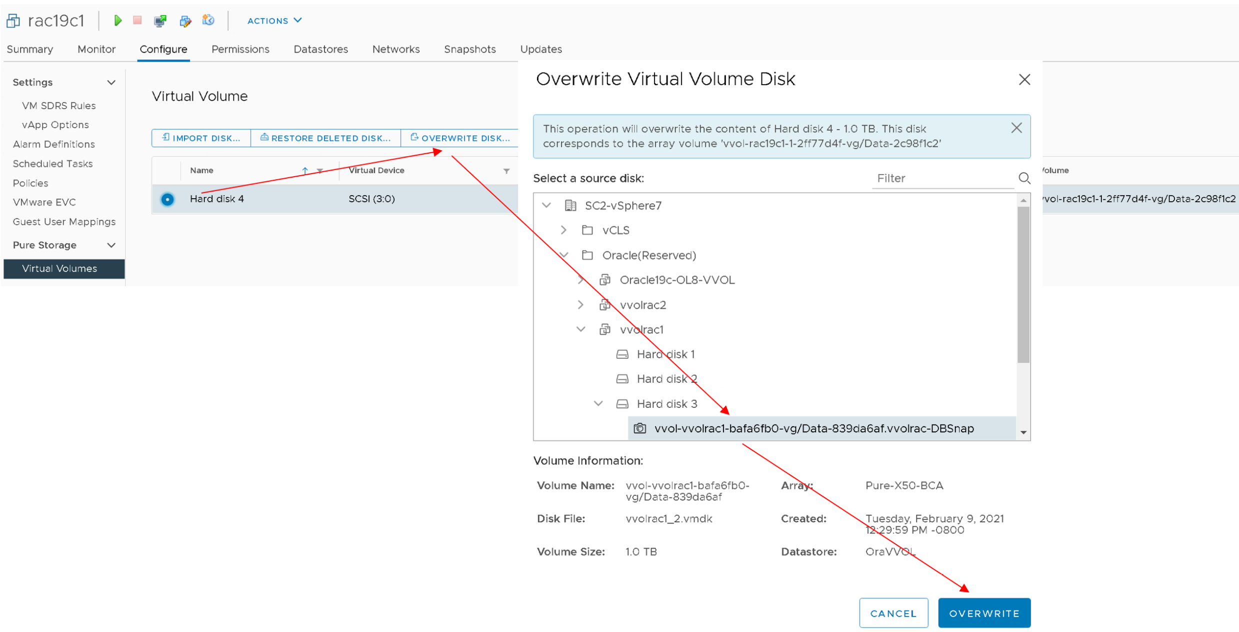


FIGURE 97. PURE PLUGIN – OVERWRITE DISK (RAC VM RAC19C1)

Oracle Database Patching

This section validates the Database patching using VMware snapshots use case of an Oracle Single-instance and Oracle RAC on vVols using Pure x50 Storage:

Oracle Single Instance Database

Two VMs are utilized for this use case:

- Production VM **Oracle19c-OL8-VVOL**
- QA VM **Oracle19c-OL8-VVOL-QAVM**

For Oracle DBAs patching databases can be a tedious and time-consuming process. As explained earlier, however, it's important that these patches are safely applied.

Two approaches also explained earlier in this paper are available:

- Clone a copy of the production database **vvol19c** from production VM **Oracle19c-OL8-VVOL** to QA VM **Oracle19c-OL8-QAVM**, using the steps explained in the Oracle Database Cloning section.
- Refresh a copy of the production database **vvol19c** from production VM Oracle19c-OL8-VVOL to QA VM **Oracle19c-OL8-QAVM** using the steps explained in the Oracle Database Refresh section.

The patching procedures can then be tested on the target VM before being implemented on the Production VM.

It's recommended to take a VM-level snapshot of the production VM before starting the patching process. In the event of any patching related issues, the database can be simply shut down down before restoring the older version of the VMDKs from the VM-level snapshot using **Overwrite Disk** in the Pure Plugin.

Oracle RAC

A 2-node Oracle RAC **vvolrac** with RAC VMs **vvolrac1** and **vvolrac2** is used in the RAC patching use case example.

Before starting the patch, it's recommended to take a snapshot of the Oracle RAC VM along with a storage-based snapshot for shared VMDKs.

In the event of any patching-related issues, the database can be simply shutdown down before restoring from the VMware snapshot, then using Pure Plugin **Overwrite Disk** to restore from the storage vVol snapshot **vvolrac-DBSnap** of the RAC database **vvolrac**.

VM Provisioning Using vVols SPBM

This section validates the VM provisioning using vVols SPBM for both an Oracle Single-instance and Oracle RAC on vVols using Pure x50 Storage:

Production VM **Oracle19c-OL8-VVOL** is utilized in this use case.

We can assign a VM storage policy in an initial deployment of a VM or when we perform other VM operations, such as cloning or migrating. This use case examines how to assign the VM storage policy when cloning a VM using vVols SPBM.

The same storage policy can be applied to the VM virtual disks and, if needed, different storage policies can be associated with different VMDKs.

The steps for cloning a VM from production VM **Oracle19c-OL8-VVOL** using vVols SPBM are shown below:

1. Production VM **Oracle19c-OL8-VVOL** currently uses the VM storage policy **VVol No Requirements Policy**.

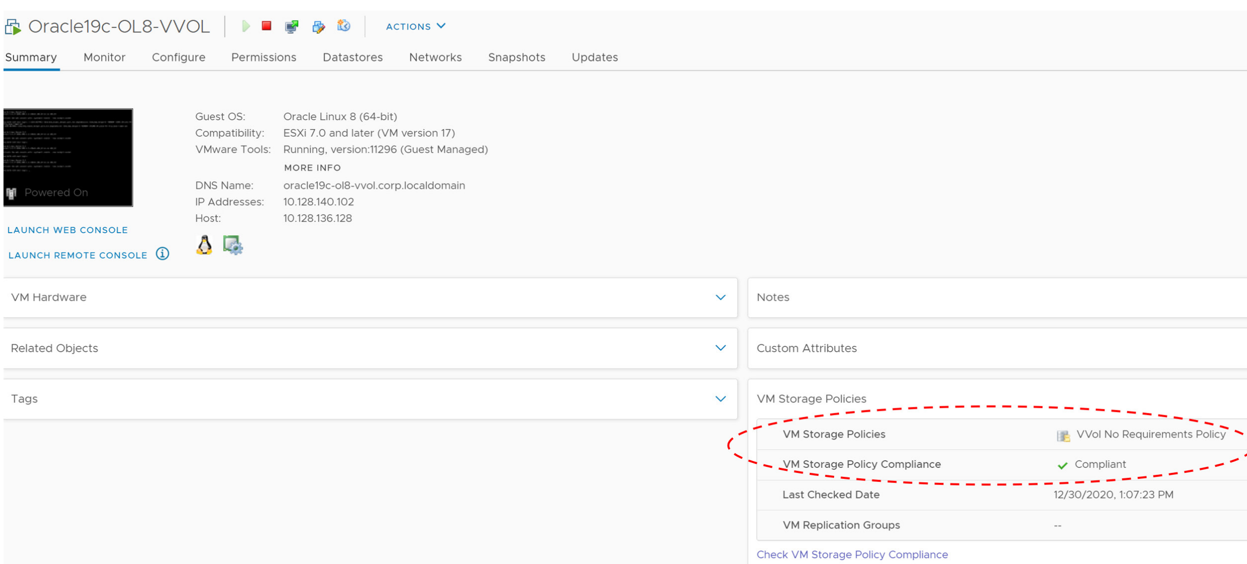


FIGURE 98. vVOL NO REQUIREMENTS POLICY

2. Clone a VM **TestClone** from production VM **Oracle19c-OL8-VVOL** using vVols SPBM.

Choosing **VM Storage Policy** as **FlashArray** automatically shows datastores which are compatible and incompatible with this storage policy. This saves VM admins a considerable amount of time in choosing the correct datastore based on the storage characteristics needed.

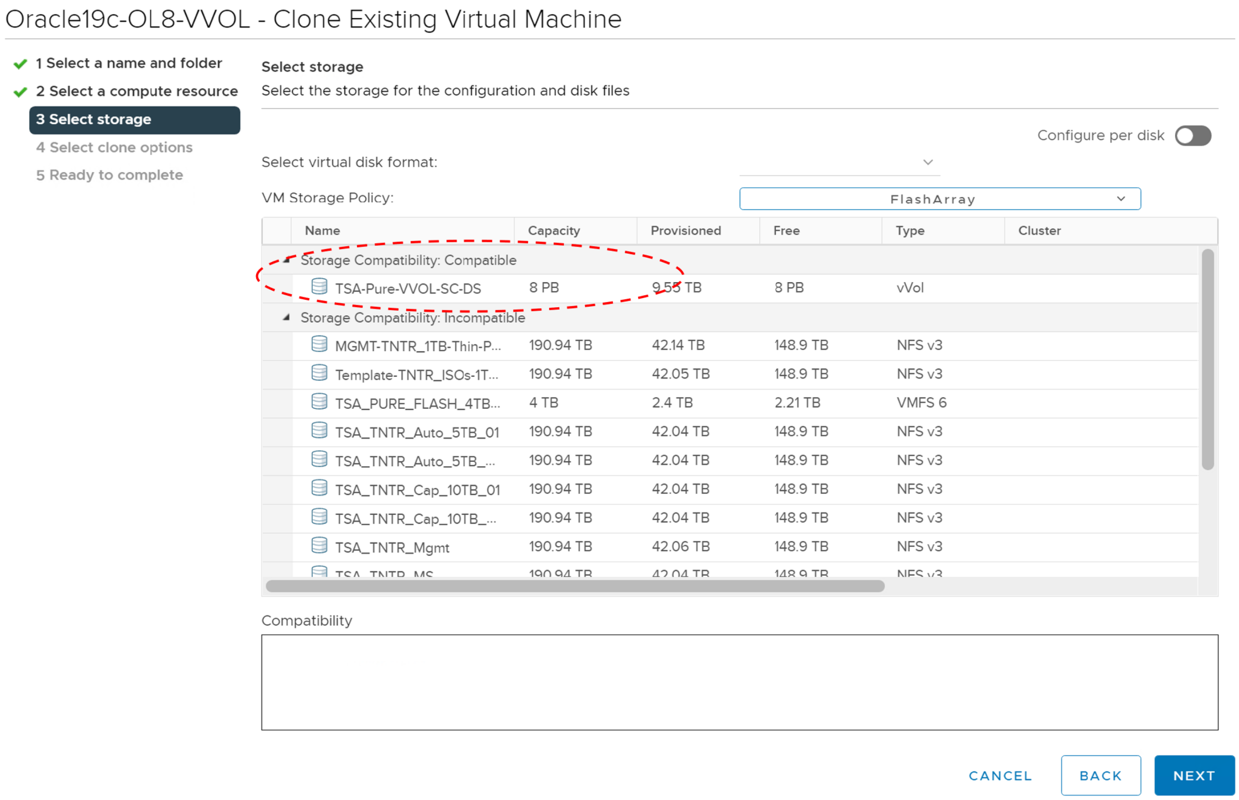


FIGURE 99. CLONING A VM

Follow the remaining steps as indicated to complete cloning of the VM.

Conclusion

Business-critical databases are among the last workloads to be virtualized in most enterprises, primarily because of the challenges posed as workloads grow and scale. In a typical virtualization project, once a proof of concept (POC) is successfully run, development, testing and staging databases come relatively easy, leaving production databases as the final hurdle to overcome.

Meeting strict business SLAs for performance, managing rapidly growing production databases, and simultaneously reducing backup windows and their impact on system performance often force DBAs to delay virtualization of business-critical databases and workloads. Frequent demands for database cloning and refreshing further complicate matters.

VMware Virtual Volumes addresses the challenges of day-to-day operations, including backup and recovery, cloning, and database provisioning.

vVols is an integration and management framework that virtualizes SAN/NAS arrays, enabling a more efficient operational model that is optimized for virtualized environments and centered on the application instead of the infrastructure.

This solution validated the following Day 2 operations of both Oracle Single Instance and RAC database on vVols using Pure Storage FlashArray//x50.

- Database backup using vVol-based snapshots and Oracle RMAN
- Database restore and recovery vVol-based snapshots and Oracle mechanism
- Database cloning using vVol-based snapshots
- Database refresh using vVol-based snapshots
- Database patching using vVol-based snapshots
- VM provisioning and cloning using vVols SPBM

Appendix A Oracle Initialization Parameter Configuration

Oracle Initialization Parameters

*.audit_file_dest='/u01/admin/vvol19c/adump'

*.audit_trail='db'

*.audit_sys_operations=TRUE

*.awr_pdb_autoflush_enabled=TRUE

*.compatible=12.1.0.0.0

*.control_files='+DATA_DG/vvol19c/control01.ctl','+DATA_DG/vvol19c/control02.ctl','+DATA_DG/vvol19c/control03.ctl'

*.db_block_size=8192

*.db_domain=''

*.db_name='vvol19c'

*.db_create_file_dest='+DATA_DG'

*.db_cache_advice='ON'

*.db_recovery_file_dest='+FRA_DG'

*.db_recovery_file_dest_size=100G

*.diagnostic_dest='/u01/admin/vvol19c'

*.enable_pluggable_database=true

*.instance_number=1

*.instance_name='vvol19c'

*.log_archive_format='%t_%s_%r.dbf'

*.open_cursors=1000

*.processes=2000

*.parallel_instance_group='vvol19c'

*.parallel_max_servers=100

*.pga_aggregate_target=256M

*.pga_aggregate_limit=6G

*.pre_page_sga=TRUE

*.remote_login_passwordfile='exclusive'

*.resource_manager_plan=''

*.result_cache_max_size=4M

*.sga_max_size=96G

*.sga_target=96G

*.thread=1

*.undo_tablespace='UNDOTBS01'

*.use_large_pages='only'

Reference

White Paper

For additional information, see the following white papers:

- *VMware Hybrid Cloud Best Practices Guide for Oracle Workloads*
- *Enabling or disabling simultaneous write protection provided by VMFS using the multi-writer flag (1034165)*
- *Oracle VMware Hybrid Cloud High Availability Guide*
- *What's New in Virtual Volumes (vVols) 2.0*

Product Documentation

For additional information, see the following product documentation:

- *Oracle 19c Database Online Documentation*
- *VMware Virtual Volumes (vVols)*
- *vVols collateral: Demos, Getting Started, FAQs, etc. on Core.vmware.com/vVols*
- *vSphere Plugin User Guide: Installing the vSphere Plugin*

Other Documentation

For additional information, see the following document:

- *VMware Solutions Lab*

Acknowledgements

Author: Sudhir Balasubramanian, Senior Staff Solution Architect, works in the Cloud Services Business Unit (CSBU). Sudhir specializes in the virtualization of Oracle business-critical applications. Sudhir has more than 25 years' experience in IT infrastructure and database, working as the Principal Oracle DBA and Architect for large enterprises focusing on Oracle, EMC storage, and Unix/Linux technologies. Sudhir holds a master's degree in computer science from San Diego State University. Sudhir is one of the authors of the *Virtualize Oracle Business Critical Databases* book, which is a comprehensive authority for Oracle DBAs on the subject of Oracle and Linux on vSphere. Sudhir is a VMware vExpert, Alumni Member of the VMware CTO Ambassador Program and an Oracle ACE.

Thanks to the following for their reviews and inputs:

- **Jason Massae**, Storage Technical Marketing Architect, VMware Core Storage (Storage Product Marketing)