PURESTORAGE®
Uncomplicate Data Storage, Forever

REFERENCE ARCHITECTURE

Running MySQL Workloads on VMware vSphere with FlashArray

Architectural guidance and best practices for deploying MySQL workloads in a combined VMware vSphere and Pure Storage® FlashArray™ environment.

Contents

Executive Summary

Server virtualization continues to be a popular infrastructure option among companies of all sizes. The ability to run multiple virtual machines (VMs) on commodity hardware can lead to higher return on investment (ROI) and better utilization of compute, storage, and network resources. As the leading virtualization platform, VMware vSphere offers a wide range of virtualization products to fit any workload or budget.

As compute and storage performance has increased, companies have found that virtualizing database servers can also help increase ROI. MySQL, a popular open-source relational database management system (RDBMS), commonly runs on VMs. A typical VMware vSphere stack consists of physical servers, high-speed networks, and either local or networked storage. VM performance depends on balancing workloads among all three critical areas. If the storage doesn't perform well, VM performance can suffer, especially when running critical MySQL databases.

By making use of the all-flash performance capabilities of Pure Storage FlashArray, organizations can enjoy increased MySQL VM performance while also getting simpler storage management, higher density, and higher availability.

The primary audience for this reference architecture includes IT architects, database administrators, VMware vSphere administrators, and storage administrators.

Benefits of Running MySQL on VMware vSphere with FlashArray Storage

Virtualizing MySQL and utilizing FlashArray storage provides the best outcomes for the infrastructure stack and application environment. FlashArray is simple, easy to use, and storage-space efficient, and it comes with a range of data services that can enhance any VMware vSphere environment's existing benefits. At the virtual infrastructure layer, vSphere enables server consolidation and application flexibility through a range of innovative utilities and functions. The combination of VMware vSphere and FlashArray storage allows for even greater consolidation, operational efficiency, and cost reduction for growing organizations as they both strive towards the same goal—simplifying application environments in a cost-effective way at little to no cost to application flexibility or operational capabilities.

Placing MySQL into this environment lets organizations focus on the core aspects of their businesses while benefiting from the combination of these components.

How to Use This Reference Architecture

This reference architecture serves as a design and deployment guide for organizations deploying MySQL. It also provides an overview of the various technologies involved with virtualizing MySQL workloads, including block storage on Pure Storage FlashArray, VMware virtualization software products, and Microsoft and Linux operating systems. It also provides optimizations for these various components that can help give MySQL the best performance possible.

The concepts, architectures, and methodologies provided in this guide can be used for any FlashArray model.

Ensure Hardware Compatibility

This guide has been validated against the components in Table 1. Operating systems and MySQL versions reflect those that are currently in mainstream support as of this RA's publication date. Older operating system versions, such as Windows Server, are in extended support, meaning that only critical fixes will be released. IT organizations should deploy supported versions where possible to receive end-to-end support from all vendors. See the links later in this paper to see which operating system versions for MySQL are supported for your deployment scenario.

Storage	Hardware	FlashArray//X™ FlashArray//XR2 FlashArray//XR3 FlashArray//XL
	Software	Purity//FA 6.0 Purity//FA 6.1 Purity//FA 6.2 Purity//FA 6.3 Purity//FA 6.4
VMware vSphere	VMware ESXi 7.x (all supported versions and updates) VMware ESXI 8	
Operating Systems	Windows Server 2019 Windows Server 2022 Red Hat Enterprise Linux 8.x SUSE Linux Enterprise Server (SLES) 15 SP3 Ubuntu 20.04	
MySQL	8.0 and above	

TABLE 1 Compatibility matrix for Pure Storage FlashArray, VMware vSphere, and MySQL

MySQL Editions and Features

Originally developed in 1995, MySQL is one of the most widely used open-source RDBMSs. MySQL was originally developed by MySQL AB and is currently maintained by Oracle Corporation. A popular open-source fork, MariaDB, maintains compatibility with MySQL.

As a commercial product, MySQL is popular across multiple industries and can easily scale. It is used to power some of the world's largest websites and is a popular alternative for enterprise applications. MySQL is also available as an open-source RDBMS, and it is a popular tool for those learning to build back-end databases or developing open-source applications.

Table 2 lists the MySQL editions available for on-premises deployments.

MySQL Edition	Description
MySQL Community Edition	This edition is available as a free download and is covered under the GNU Public License (GPL). MySQL Community Edition is supported by a large community of open-source developers and users. For more information, visit www.mysql.com/products/community/.
MySQL Standard Edition	This edition includes all of the features of MySQL Community Edition, but 24/7 commercial support and consulting services are available from Oracle Corporation. For more information, visit www.mysql.com/products/standard/.

TABLE 2 MySQL editions

VMware vSphere

Similar to MySQL, VMware provides multiple vSphere editions that have different features and capabilities, including:

- VMware vSphere Standard Edition
- VMware vSphere Enterprise Plus Edition
- VMware vSphere+

For a comparison of the different editions, see VMware vSphere Product Line Comparison.

VMware vSphere is comprised of the following components and technologies:

- **VMware ESXi:** A hypervisor installed onto physical servers. VMware ESXi is responsible for abstracting processors, memory, storage, networking, and other resources and presenting them to multiple VMs.
- **VMware vCenter Server:** A management tool providing a single user interface (UI) for managing one or more ESXi hosts.
- **VMware vSphere client:** A graphical web-based management tool that provides access to VMware vCenter Server.
- **VMware vSphere software development kits:** Programming interfaces through which users can manage VMware vSphere components.

- **VMware vSphere vMotion:** A feature that enables the live migration of virtual machines between ESXi hosts in a data center.
- **VMware vSphere Storage vMotion:** A feature that enables the live migration of VMs between datastores.
- **VMware vSphere High Availability (HA):** A feature that detects when an ESXi host failure occurs and automatically restarts VMs on other available ESXi hosts within a cluster.
- **VMware vSphere Distributed Resource Scheduler (DRS):** A feature that manages resource utilization across multiple ESXi hosts and balances virtual machines across a collection of hardware resources.
- **VMware vSphere Fault Tolerance (FT):** A feature that creates a duplicate of a selected workload on a different ESXi host within the cluster to provide continuous availability.
- **VMware vSphere Distributed Switch:** Enables a single virtual network to be available across many ESXi hosts. This enables the centralization of network configurations.

Unified Block and File on FlashArray

Pure Storage FlashArray is a software-defined, unified block and file storage product that offers an effortless and consistent experience for both storage administrators and application users. FlashArray also offers data-reduction capabilities that efficiently reduce the size of data without impacting performance.

All Pure Storage products offer the Evergreen® Storage upgrade model that increases capacity and performance without the need to purchase new storage products. Additionally, Pure Storage FlashArray lets organizations drive out direct carbon usage in their data storage systems by up to 80%, compared to competitive all-flash and hard-drive-based systems.

The FlashArray product line caters to multiple business needs and use cases with these distinct offerings:

- FlashArray//C™: An all-QLC FlashArray with consistent performance at 2–4ms latency for capacity-oriented workloads.
- FlashArray//X: FlashArray//X provides latency as low as 150µs to power critical applications and business operations.
- FlashArray//XL™: Enterprise-grade performance and scalability for demanding workloads.

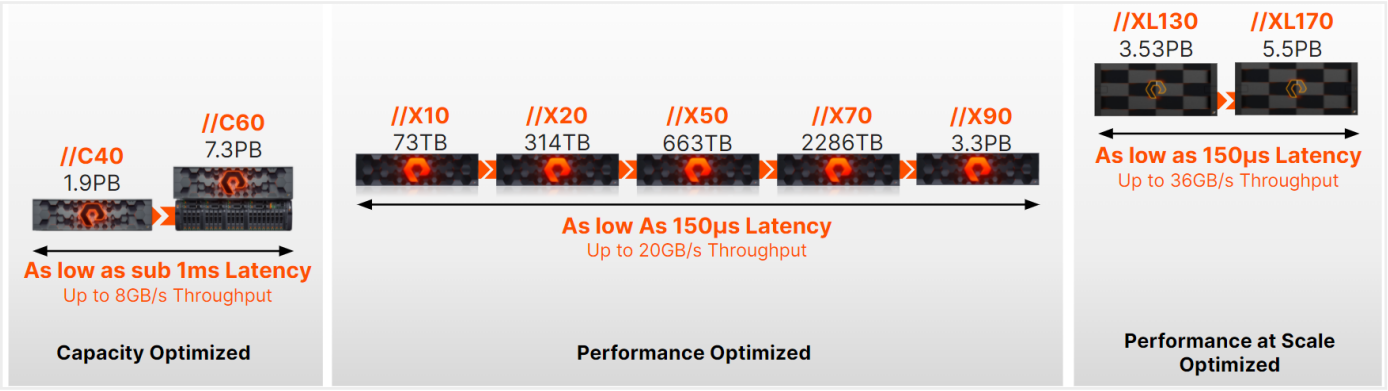


FIGURE 1 Pure Storage FlashArray suite of products

FlashArray and Pure Cloud Block Store™ systems are powered by Purity for FlashArray. Purity delivers rich, enterprise-level data services that ensure data is stored in the most secure and efficient way while providing additional functionality to extend storage capabilities, including:

- **Data reduction:** Purity averages an industry-leading 5:1 data reduction with a total efficiency of 10:1 with thin provisioning.
- **High availability:** Purity protects against concurrent dual-drive failures and initiates re-builds automatically within minutes.
- **Always-on ransomware remediation:** Cost-efficient, portable, SafeMode™ Snapshots prevent cyber attackers from tampering with or maliciously destroying critical recovery data.
- **On-demand data portability:** Quickly and easily move data for both physical and VMs where it most cost-effectively meets service-level agreements (SLAs).
- **Rich data services:** Data service functionality such as ActiveCluster™, ActiveDR™, and asynchronous replication provide increased resilience, availability, and enhanced application workflows.

Figure 2 shows the Purity for FlashArray data services architecture.

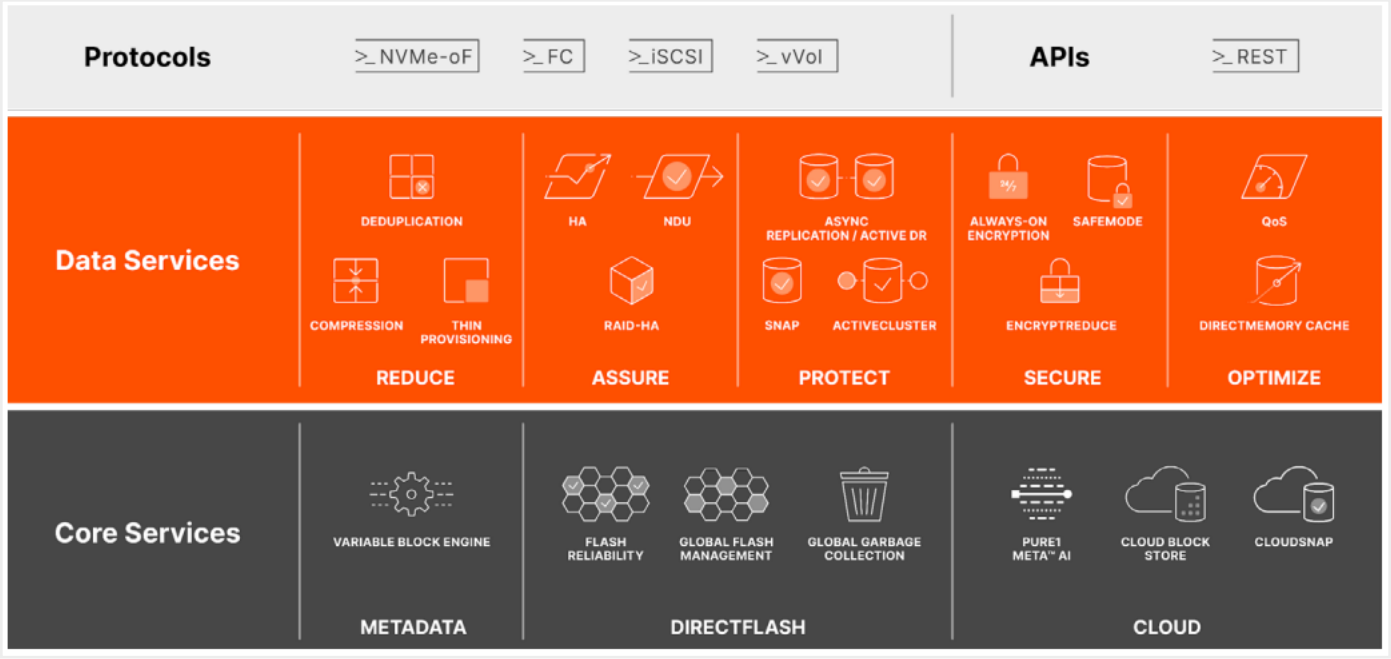


FIGURE 2 Purity for Flash Array data services architecture

Component Architecture and Recommended Configuration

This guide provides an overview of the various components involved with running MySQL VMs in a combined VMware vSphere/FlashArray environment. Within a MySQL stack are a number of components that support MySQL. Figure 3 is a visual representation of a typical MySQL stack running in a FlashArray and VMware vSphere environment.

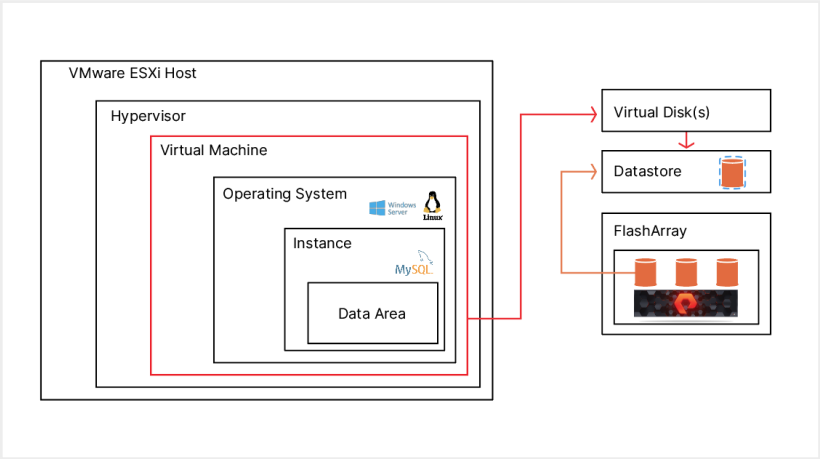


FIGURE 3 A virtualized MySQL stack that uses Pure Storage FlashArray

Table 3 provides an overview of each of the architecture components.

Component	Description
MySQL Database	A MySQL database is a collection of relational tables and data that is stored on a computer or server. Relational data is stored in multiple data files on a file system.
MySQL Instance	A MySQL database instance is a single MySQL database process running as a daemon (Linux) or service (Windows) that manages one or more databases. The database process monitors a data directory where the database files are stored, and it listens for requests on a TCP/IP port. A server can run multiple instances that monitor different directories and listen on different ports. For more information, visit the MySQL 8.0 Reference Manual.
MySQL Storage Engine	Storage engines are modular database components that the instance uses to create, read, and update data in a database. Storage engines allow for different table types to be created, allowing for more flexible use cases to be adopted in a single instance. The default storage engine, InnoDB, provides the most comprehensive database capabilities from transaction processing to data analysis.
Operating System (OS)	The instance runs on an OS, such as Linux or Windows. The OS handles all scheduling, network, CPU, and storage functions.

Component	Description
VM	A VM uses software to run an OS and applications instead of a physical machine. The VM is organized as a group of files that contain configuration information and data. One or more VMs or guests run on a VMware ESXi host. Each VM on a host runs independent of other VMs on the host. The host provides each VM with networking, storage, CPU, and memory. For more information about VMware vSphere VMs, visit the VMware glossary.
VMware ESXi Hypervisor	A hypervisor is a software layer that presents the VM with a virtual hardware profile and manages all hardware-specific calls from the VM to the underlying physical hardware. In a VMware vSphere environment, the ESXi hypervisor is a type-1 with a small installation footprint. For more information about the ESXi hypervisor, visit the VMware glossary.
VMware ESXi Host	The host is the physical server the VMware ESXi hypervisor and VMs run on. Multiple hosts can be pooled to form a cluster that can be managed centrally and can run thousands of VMs.
Datastore	In a VMware vSphere environment, VMware ESXi hosts use datastores to store VM files. These datastores are logical containers that provide a uniform model for storing VM files by hiding the specifics of the underlying storage. VMware ESXi servers can use local disks as datastores, networked file storage, or network block storage. A VM's files, including the configuration file (VMX), VM disk files (VMDKs), and virtual memory file (VMEM), reside on a specific datastore, but when networked storage is used, they can be non-disruptively migrated to a different datastore using Storage vMotion. VMware vSphere provides several datastore types that administrators can utilize for VM storage: • Virtual Machine File System (VMFS version 5 and 6): VMFS is a special, high-performance file system that resides on block storage volumes and is optimized for storing VMs. • Network File System (NFS version 3 and 4.1): This datastore uses the NFS protocol over TCP/IP to access NFS volumes located on a network-attached storage server. The VMware ESXi host mounts this volume as an NFS datastore. • VMware vSphere Virtual Volumes (vVol): This datastore type represents a storage container. All virtual disks in the container receive the benefit of policy-based, granular storage configuration. Raw Device Mapping (RDM) is not a datastore, but it is still a way in which VMs can be provided with storage resources. This storage capability uses raw LUNs or logical disks to provide volume storage directly to one or more VMs. For more information about datastore types, visit the VMware vSphere library.
Storage Protocol	Pure Storage supports Fiber Channel, iSCSI, NVM Express (NVMe) over RoCEv2, and NVMe over Fibre Channel (NVMe-oF).

Component	Description
FlashArray Volumes	Pure Storage provides high-performance storage for both VMs and MySQL databases. These databases can be stored on virtual disks attached to the VM. VMs and their associated virtual disks can be stored on block storage volumes that are hosted on FlashArray storage arrays and connected to a VMware ESXi host. Virtual volumes (vVols) are becoming a common method for storing VM data on storage arrays. When using vVols, each virtual disk's data is contained in a vVol, which is stored in a vVol datastore. Each vVol corresponds to a volume on the FlashArray, and array-based features such as snapshots and replication can be used at the virtual disk level. FlashArray volumes can be connected to VMware ESXi hosts using Fibre Channel, iSCSI, NVMe over RoCEv2 (NVMe-oF/RoCEv2), NVMe-oF, or NVMe over TCP. vVols can only be connected using Fibre Channel or iSCSI. vVols can only be connected to via Fibre Channel or iSCSI. For more information about using FlashArray with VMware, visit the Pure Storage VMware Platform Guide.

TABLE 3 Architectural components of a virtualized MySQL deployment that uses VMware vSphere and Pure Storage FlashArray

FlashArray Storage and Virtual Infrastructure

FlashArray volumes can be used with VMware virtual infrastructure in several ways, including presenting volumes as vVols and VMFS-formatted volumes.

FlashArray Volumes as vVols

VMware vSphere vVols are a storage technology that provides policy-based, granular storage configuration and control of VMs. Through API-based interaction with an underlying array, VMware administrators can maintain storage configuration compliance using native VMware interfaces. vVols offer the ability to remove abstractions between applications running on virtual systems and the underlying storage. This concept ensures that the focus moves away from the infrastructure and focuses more intently on the application running in the virtualized environment.

With FlashArray, vVols let virtual disks and regular block storage volumes be seen as one and the same. FlashArray presents a datastore to the ESXi host and any virtual disks migrated onto the datastore have a corresponding FlashArray volume that can be seen and managed using the array's management tools. In a typical VMFS datastore environment, virtual disks are limited to 62TB in size, and any VM with many large virtual disks would need to be divided among several VMFS datastores. With vVols, large virtual disks can be contained within the single vVol datastore, vastly simplifying management at this scale. Figure 4 demonstrates how virtual disks are stored on a vVol datastore.

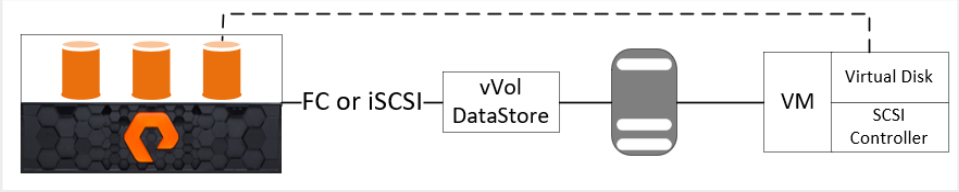


FIGURE 4 vVols allow for a volume to be created on FlashArray that directly represent a virtual disk

Detailed instructions for configuring vVols with FlashArray is beyond the scope of this document. Table 4 provides a high-level overview of the configuration items with links to detailed descriptions and best practices in the Web Guide: Implementing vSphere Virtual Volumes with FlashArray.

vVol Configuration Item	Description and Recommendations
vVol Requirements and Recommendations	vVols are only available on specific versions of VMware vCenter, ESXi, Purity//FA, and FlashArray. Best practices: • Use Purity//FA 6.2.10 or later. • Use VMware vCenter 7.0 Update 3f (build 20051473) or later. • Use VMware ESXi 7.0 Update 3f (build 20036589) or later. • When registering the VMware vSphere APIs for Storage Awareness (VASA) provider, use a local FlashArray user account. • Do not run VMware vCenter VMs on vVols. Visit this section of the Failure Scenario knowledge base (KB) for more information. • The protocol endpoint should be connected to host groups and not individual hosts. • Configure snapshot policies for all configured vVols. • Use VM hardware version 15 or later.
Configuring the FlashArray VASA Provider	The VASA is the interface that VMware vCenter and ESXi use to interact with storage arrays. FlashArray supports VASA version 3.0, which was introduced with VMware vSphere 6.5. As such, the vSphere cluster must be running vSphere 6.5 or higher. A FlashArray VASA provider is a service that is registered with VMware vCenter. Specific steps must be followed to register the FlashArray VASA provider: 1. Create and manage VASA provider certificates. 2. Register the FlashArray VASA provider using the Pure Storage vSphere plugin, the VMware vSphere client, or PowerShell. 3. Verify the VASA provider registration. VMware administrators can also remove and de-register VASA providers using the vSphere client and PowerShell.
Configuring Host Connectivity	A host object is a collection of initiators that refer to a physical host. A FlashArray host object must have a one-to-one relationship with an ESXi host, and every active initiator for an ESXi host must be added to the FlashArray host object. A host group is a collection of host objects. Pure Storage recommends you deploy your ESXi hosts into a vSphere cluster within vCenter, with each cluster that uses FlashArray storage having its own host group. Each ESXi host in the cluster must have a corresponding host that is added to a host group. Pure Storage also recommends that the number of ESXi hosts in a host group match the number of hosts in the vSphere cluster. Each ESXi host object should have the ESXi host personality enabled. Once you've created host objects and host groups, you can then connect FlashArray volumes to the host objects or host groups using the Pure Storage vSphere plugin.

vVol Configuration Item	Description and Recommendations
Protocol Endpoints	VMware uses protocol endpoints (PEs) to avoid limitations of LUNs, such as the 512-device limit per host and time-consuming input/output (I/O) interconnect rescans. Each PE allows up to 16,383 vVols bound to each ESXi host simultaneously and doesn't require an I/O rescan when a new binding is created. FlashArray administrators can manage PEs through either the FlashArray UI or command-line interface (CLI). VMware administrators can view PEs that are connected to ESXi hosts using the VMware vSphere client.
vVol Datastores	vVols replace LUN-based VMware datastores that are formatted with VMFS. vVol datastores do not contain a file system, and vVol-based virtual disks are not encapsulated within files like they are on VMFS. vVol datastores are mounted to an ESXi host with access to a PE on the array that hosts the vVol datastore. Mounting a vVol first requires registering a VASA provider and connecting a PE to the ESXi host. Once these tasks are complete, VMware administrators can use a number of methods to mount vVol datastores.
Types of vVols	FlashArray organizes vVols associated with a VM as a volume group. When a VMware administrator creates a vVol-based VM, the FlashArray that hosts the VM creates a volume group with the following naming schema: vvol-{VM Name}-{unique 8 character string}-vg vVols do not change the structure of a VM: • Each VM has a VMX configuration file that describes the VMs virtual hardware and settings. • Each powered-on VM has a swap file. • Each VM's virtual disks is a storage object that limits guest OS disk capacity. • Each VM has a vmem memory file that stores snapshots of its memory state. While vVols do not contain a file system, vSphere makes the structure appear to be that of a conventional VM on a VMFS datastore. Additionally, vVol-based VMs use four types of vVols: • Configuration vVol, which stores the VM's VMX configuration file, pointers to the virtual disk VMDK files, and other files • Data vVol, which corresponds to a virtual disk • Swap vVol, which contains the VM's swap data • Memory vVol, which contains VM suspension or snapshot data; note that the VM snapshot data is different from FlashArray snapshots VMware administrators can view and manage each vVol using the VMware vSphere client.
Recovering Deleted vVols	VM data and configuration vVols can be recovered within 24 hours of deletion using a variety of methods. vVol recovery can use both FlashArray and vSphere tools.

vVol Configuration Item	Description and Recommendations
vVol Binding	One of the primary goals of using vVols is the ability to scale. Using PEs instead of LUNs helps achieve that goal. With PEs, ESXi hosts can bind and unbind (connect and disconnect) vVols dynamically as needed. ESXi hosts can also provision VMs and power them on and off, even if VMware vCenter is not available. vVols are bound to specific ESXi hosts for as long as they are needed. vVols are bound to an ESXi host under the following circumstances: • The VM is powered on. • A folder is navigated to a vVol datastore using the VMware vCenter graphical user interface (GUI), Secure Shell (SSH) session, or console. • A VM is moved using VMware vSphere vMotion. • A VM is created, deleted, or reconfigured. • A VM clone is created from a VM or template. • A snapshot is taken of a VM. Note that binding and unbinding is automatic. No administrator input is required.
Snapshots of vVols	vVols give VMware administrators the ability to snapshot individual VMs using storage-array snapshot capabilities instead of VMware's built-in snapshot capabilities. vVol snapshots on FlashArray differ from VMware's native snapshots. A VMware snapshot freezes a VM's virtual disk files at the point the snapshot is initiated, and then creates separate VMDK delta files where all subsequent virtual disk writes are stored. VMware vSphere redirects all new writes to the delta files. VMware vSphere directs all reads of data prior to the snapshot to the original VMDKs, and it reads data from after the snapshot to the delta files. While this works well, VMware snapshots introduce I/O latency that can negatively impact VM performance. Because of VMware's snapshot limitations, VMware recommends limiting VMware-based snapshot retention periods. vVol snapshots based on FlashArray differ from VMware-based snapshots in that they are instantaneously created, they don't affect VM performance, they can be retained for longer periods, and they can be easily copied. For these reasons, and due to the limitations of VMware-based snapshots, VMware and Pure Storage recommend using vVol snapshots instead of VMware snapshots. As with VMware-based snapshots, array-based managed snapshots that are created through the VMware vSphere client can include snapshots of the VM's memory and snapshots that quiesce the file system. Administrators can revert VMs to a snapshot and delete snapshots. Snapshots that are created outside of the VMware vSphere client using tools such as the FlashArray GUI, CLI, or REST interfaces are called unmanaged snapshots. This type of snapshot can be used in a VMware environment, but with certain restrictions. VMware administrators can manage vVol snapshots using the FlashArray plugin, including viewing vVol details, creating managed snapshots, restoring a vVol from a snapshot, and creating copies of a vVol from a snapshot.

vVol Configuration Item	Description and Recommendations
Storage Policy-based Management	One of the unique benefits of using vVols is the ability to apply granular storage policies to each vVol. Because vVols are native to VMware vSphere and integrated, VMware administrators do not need storage vendor–specific tools to manage vVols, as array functionality is integrated directly with VMware vSphere. Administrators use native VMware vSphere tools, such as the VMware vSphere client or PowerCLI. Storage policies are collections of storage capabilities that are defined by VMware administrators. Storage capabilities are specific to arrays. When a storage policy is applied, VMware filters out non-compliant storage so that only compliant options are presented for configuring storage for VMs or vVols. FlashArray provides several storage capabilities that are available and cannot be disabled, including data-at-rest encryption, deduplication, compression, RAID protection, and flash storage. Other capabilities are available and can be configured by VMware administrators. Using native VMWare vSphere tools, VMware administrators can also create a storage policy, check VM storage policy compliance, and assign storage policies to a VM or virtual disk.
Replicating vVols	FlashArray can replicate vVols to other FlashArray instances. VMware vCenter is aware of replicated VMs and can fail them over to FlashArray at a remote location. VMware administrators can manage replication using tools such as the VMware vSphere web client, PowerCLI, REST APIs, and VMware vRealize Orchestrator. FlashArray vVol replication is also supported by VMware Site Recovery Manager and is certified by both VMware and Pure Storage. More information is available in SRM User Guide: vVol Periodic Replication SRM Workflows.
vVol Reporting	vVol integration with FlashArray gives VMware vSphere insights into vVol storage. Additionally, the same integration gives FlashArray the ability to recognize and report on both entire vVol-based VMs that are implemented as volume groups, and individual virtual disks that are implemented as volumes. For example, the Volumes tab in the FlashArray Storage pane lists an array's volume groups. The volume group naming schema follows the following pattern: vvol-VMname-vg. The VM name is set when the VM is first created. When a VM is renamed through VMware vCenter, the volume group is not automatically renamed on the FlashArray. The volume name can be renamed or updated using a manual workflow. A FlashArray can also report on VM and vVol performance history. The history of a VM's or vVol's I/O operations per second (IOPS), latency, and data throughput (bandwidth) can be viewed through the FlashArray GUI.

vVol Configuration Item	Description and Recommendations
Migrating VMs to vVols	VMware administrators can use VMware Storage vMotion to migrate VMs from VMFS, NFS, or raw device mappings (RDMs) to vVols. When migrating, VMware administrators select a FlashArray vVol datastore as the migration target. During migration, administrators can select a storage policy for the migrated VM that provides additional features. When migrating from VMFS or NFS datastores, because the administrator is using Storage vMotion, the VM remains online and available during the migration. RDM migration is more involved and might require VM downtime. Contact Pure Storage Technical Support for more information on migrating an RDM to a vVol. Pure Storage is planning to make this option available to FlashArray administrators in a later release.
Data Mobility with vVols	Because a vVol-based VM's storage is not encapsulated in a VMDK file, the VM's data can be easily shared and moved. A data vVol is a virtual block device that is attached to a VM. Therefore, a data vVol or a volume created by copying a snapshot of it can be read by software that understands its contents, such as an NFS or XFS file system created on the vVol by the VM. As such, it is possible to present a data vVol or the volume created from a snapshot of a data vVol to a physical server. Additionally, a volume created by a physical server can be presented to a vVol-based VM as a vVol.

TABLE 4 vVol configuration recommendations for use with Pure Storage FlashArray

FlashArray Volumes with VMFS

A single FlashArray volume can serve as a VMFS datastore of up to 64TB in size. This volume can then contain multiple virtual disks for one or more VMs that collectively do not exceed the datastore size. Figure 5 shows how virtual disks are stored on the datastore.

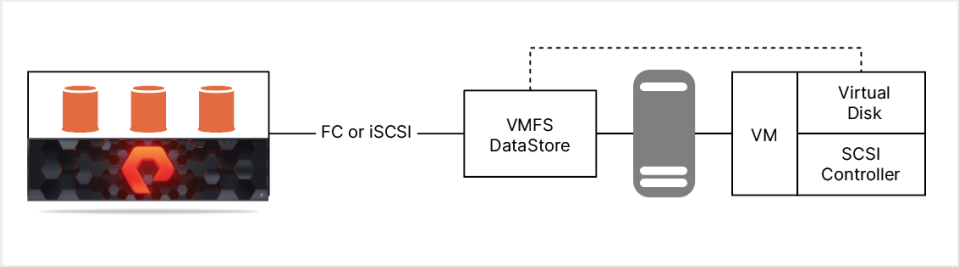


FIGURE 5 A single VMFS datastore that maps to a single volume can store multiple virtual disks

FlashArray Volumes as Mapped Devices

A single volume can serve as a mapped device with a raw device mapping (RDM). RDMs provide up to 64TB of storage for a single VM. The mapped device consumes the entire volume, which can then be shared between one or more VMs. There are a number of considerations and limitations for RDMs, which are listed in the VMware documentation library. Figure 6 shows what an RDM looks like from a FlashArray perspective.

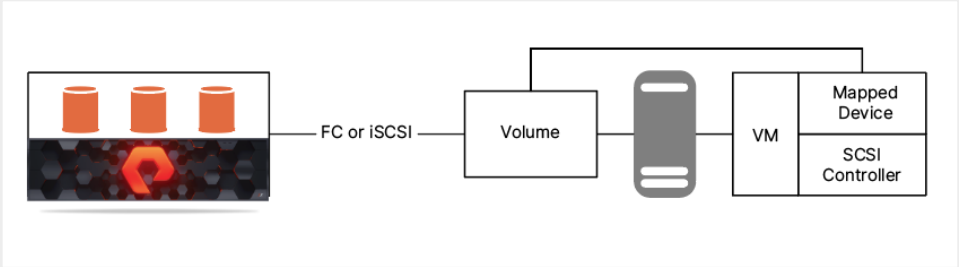


FIGURE 6 An RDM device can be used by one or more VMs, but the entire RDM volume is consumed

MySQL Configuration and Best Practices

Administrators can optimize each component of the MySQL stack. While SQL statements, client code, and database structures can be optimized by software developers and database administrators, this section discusses optimization from the FlashArray level up through the MySQL instance.

Network Optimization

Pure Storage recommends optimizing network settings for Fibre Channel, iSCSI, and host-specific implementations. Table 5 lists various networking optimizations from Web Guide: FlashArray VMware Best Practices. For more information about each topic, click the links in the Network Configuration Item column.

Network Configuration Item	Description and Recommendations
SAN Design and Setup	Storage area network (SAN) design and setup is an important consideration for optimal performance. Pure Storage recommends specific standards when using Fibre Channel, iSCSI, or host-specific protocols. The Web Guide: FlashArray VMware Best Practices page contains the specific standards and links to host-specific best practice guides.
Topology	When designing a SAN network, more hops equal higher latency, which negatively impacts storage performance. Pure Storage recommends a flat fabric where FlashArray is only one hop away from applications that run from it. For example, iSCSI should not be routed. You should also watch for topological bottlenecks and blade server chassis oversubscription.
Physical Paths	This applies to Fibre Channel and iSCSI protocols. Storage devices are often over-subscribed, so you should use all of the FlashArray ports to balance connections between hosts and the FlashArray. This provides more pathways, resiliency, and performance; mitigates physical problems; and takes better advantage of CPU allocation.

Network Configuration Item	Description and Recommendations
Clean Paths	A clean physical cable plant is a requirement for optimal performance. Make sure cable tips are clean and monitor switches for physical layer errors using tools such as the Brocade portErrShow tool. Avoid patch panels if possible.
Port Connection Speeds	This applies to Fibre Channel and iSCSI protocols. For best performance, your network should use 8Gb/s or 16Gb/s host bus adapters (HBAs). If you have 2GB or 4Gb/s HBAs, you should add more physical paths and use multipathing, or fix ports speeds to one step down if you are two or more port speeds down from your highest speed HBA. For example, if you are running 2Gb/s in an 8Gb/s SAN, you should step down to 4Gb/s on the FlashArray. Various methods are available for checking port speed on FlashArray and switches.
Zoning	This applies to Fibre Channel. You can zone any single initiator to as many FlashArray ports as you need. Configuring the same initiator to multiple targets is acceptable, while configuring multiple initiators to multiple targets is not recommended.
Jumbo Frames	Jumbo frames allow Ethernet frames larger than 1,518 bytes. While the default maximum transmission unit (MTU) for most devices is 1,500 bytes, FlashArray can support up to 9,000 bytes. Jumbo frames must be enabled across the full path from initiator to target, including HBAs and switches.

TABLE 5 Networking optimizations and recommendations for Fibre Channel, iSCSI, and host-specific implementations

FlashArray Performance Optimizations

While Pure Storage FlashArray automatically manages many of the performance aspects of the storage array, VMware and storage administrators can help further optimize FlashArray performance and storage utilization. This section provides an overview of the best practices for using Pure Storage FlashArray in a VMware vSphere environment.

While detailed descriptions of MySQL optimizations are beyond the scope of this document, Table 6 lists various storage configuration optimizations from Web Guide: FlashArray VMware Best Practices. For more information about each topic, click the links in the Storage Configuration Item column. Please note that the FlashArray vSphere client plugin automates many of the configuration items in the following sections and is the recommended configuration method.

Storage Configuration Item	Description and Recommendations
Host and Host Group Creation	A host object is a collection of initiators that refer to a physical host. A FlashArray host object must have a one-to-one relationship with an ESXi host, and every active initiator for an ESXi host must be added to the FlashArray host object. A host group is a collection of host objects. Pure Storage recommends you deploy your ESXi hosts into a vSphere cluster within vCenter, with each cluster that uses FlashArray storage having its own host group. Each ESXi host in the cluster must have a corresponding host that is added to a host group. Pure Storage also recommends that the number of ESXi hosts in a host group match the number of hosts in the vSphere cluster. Each ESXi host object should have the ESXi host personality enabled. Once you've created host objects and host groups, you can then connect FlashArray volumes to the host objects or host groups. Best practice: Match FlashArray host groups with vCenter clusters.
Volume Sizing and Count	VMFS supports a maximum size of 64TB. While FlashArray volumes can be much larger, Pure Storage recommends using a smaller number of large volumes up to VMFS's 64TB limit, rather than a large number of smaller volumes.
VMFS Version Recommendations	Pure Storage recommends using the latest supported version of VMFS that is compatible with the version of ESXi running on a host. For example, for ESXI versions 5.x through 6.0, use VMFS-5. For ESXi version 6.5 and later, use VMFS-6. Note that VMFS-6 is not the default option for ESXi 6.5, so use care in choosing VMFS-6 when creating datastores. Best practice: Use the latest VMFS version supported by the ESXi hosts. The VMFS version is dependent on the version of ESXi the host is running.
Datastore Performance Management	Storage administrators and VMware administrators have a number of options for optimizing the performance of datastores. These include managing queue depth limits at both the HBA and hypervisor levels, dynamic queue throttling, storage input/output (I/O) control (SIOC), and storage dynamic resource scheduler (storage DRS). Best practices: Do not modify queue depth (QD) limits; leave them at their default settings. Only raise QDs when performance requirements dictate and Pure Storage Support or VMware Support provide guidance.

Storage Configuration Item	Description and Recommendations
Datastore Capacity Management	Administrators should carefully monitor datastore capacity usage using the capabilities of VMware ESXi and VMware vCenter. VMFS and FlashArray report different volume capacities: VMFS reports how much space is currently allocated to thin or thick virtual disks, whereas FlashArray reports the physical capacity that has been written to the volume after data reduction. Administrators can configure email alerts in VMware vCenter when storage space reported by VMFS reaches certain thresholds. Additionally, administrators can use VMware vRealize Operations Manager with the FlashArray Management Pack to monitor storage space. When a capacity warning occurs, administrators have several options: 1. Increase the volume capacity. 2. Move VMs to a new volume. 3. Add a new volume. While FlashArray has the capability to shrink a volume, Pure Storage does not recommend shrinking volumes with VMFS partitions due to the inability of VMFS itself to shrink. FlashArray has the ability to create point-in-time snapshots of volumes that administrators can use for backup/restore operations or test/dev environments. These snapshots are not a complete copy of the volume; rather, they are a metadata point-in-time reference to data blocks on the array. The snapshot cannot be directly mounted. The snapshot metadata must be copied to an actual volume. Additionally, a new VMFS signature must be written to the volume. If a snapshot is not needed more than once, administrators can create a snap copy from one volume to another. Best practice: Configure capacity alerts to send a message or initiate an action.
VMFS Space Reclamation	"Dead space" on a FlashArray volume that hosts VMFS partitions is space that has been allocated in VMFS, but is no longer being used. For example, when an administrator deletes or moves a virtual disk or VM to a different datastore, the space that was used by the virtual disk or VM is not considered "dead" on the array. The array doesn't know that the space is no longer used, which turns the blocks into dead space. The same concept can be applied to space used inside a VM. When files are moved or deleted from the VM's file system, the underlying VMFS file system and array do not know that the space is no longer being used, turning those blocks into dead space. Depending on the version of ESXi that is in use, administrators can use UNMAP to free up dead space. vSphere 5.5 and 6.0 offer an UNMAP option through the esxcli command. With the release of VMware vSphere 6.5, VMware introduced automatic UNMAP support in VMFS-6. Pure Storage recommends that this capability be set to "low" and not disabled. VMware vSphere 6.7 and later introduce new automatic UNMAP options and configurations. Pure Storage still recommends using the default option of "low." Best practice: For the shortest UNMAP duration, use a large block count.

Storage Configuration Item	Description and Recommendations
Space Reclamation in VMs	As of VMware vSphere 6.0, Windows VMs can make use of UNMAP capabilities on their virtual disks. VMware vSphere 6.5 introduced this capability for Linux VMs. UNMAP operations within the VM are passed down to the VMFS file system. UNMAP must be enabled in the VM, and the VMFS3.EnableBlockDelete parameter must be enabled on VMFS-5 datastores. No additional parameters are needed for VMFS-6 datastores. ESXi requires that UNMAP requests sent from a guest VM be aligned to 1MB. Pure Storage recommends upgrading ESXi 6.5 to patch release ESXi650-201703001 (2148989), which is more tolerant of misaligned UNMAP requests. FlashArray data deduplication changes the behavior of UNMAP commands. Visit "What to expect after UNMAP is run on the FlashArray" for more information. Best practices: • Use the 32KB or 64KB allocation unit size for NTFS to enable automatic UNMAP in a Windows VM. Verify that the allocation unit corresponds with the requirements for a MySQL InnoDB page size. Visit this link for more details. • Mount Linux file system with the "discard" option to enable UNMAP in Linux-based VMs. To manually reclaim space use the fstrim utility.

TABLE 6 Storage optimization recommendations when using Pure Storage FlashArray

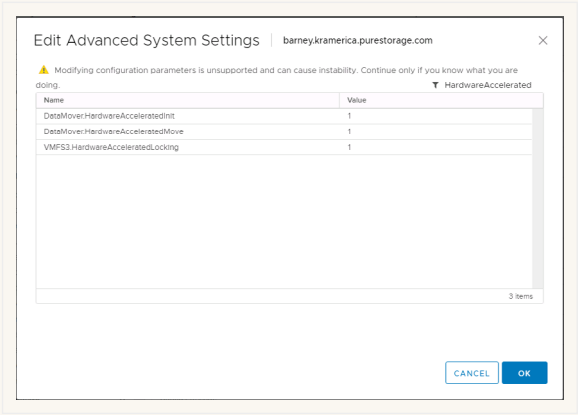
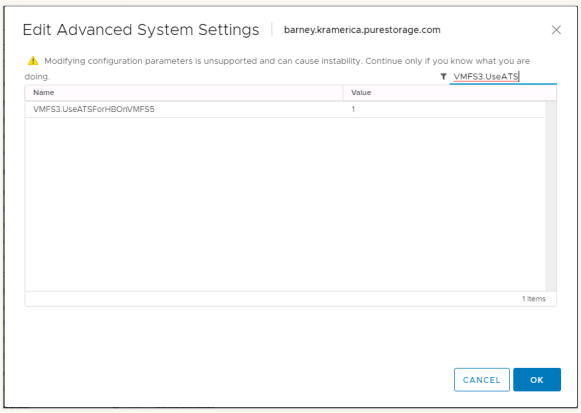
VMware ESXi Host

VMware administrators and infrastructure administrators can optimize VMware ESXi hosts for optimal performance with a FlashArray ecosystem. Table 7 lists VMware ESXi host optimizations and best practices.

VMware ESXi Host Configuration Item	Description and Recommendations
VMware Native Multipathing Plugin (NMP) Configuration	VMware provides a Native Multipathing Plugin (NMP) layer in vSphere as part of the VMware APIs for Pluggable Storage Architecture (PSA). This feature lets VMware aggregate storage I/O across multiple channels, and it provides the capability to send failover commands when a path fails. The path selection can be fixed, most recently used, or round robin.

VMware ESXi Host Configuration Item	Description and Recommendations
Round Robin Path Selection Policy	Pure Storage requires FlashArray volumes use the round robin path selection policy. This policy rotates I/O between all discovered paths for a volume and allows VMware ESXi hosts and their VMs to improve performance by using all available resources, such as HBAs and target ports. Best practice: Use the round robin path selection policy for FlashArray volumes. The round robin path selection policy allows additional tuning of path-switching behavior using the I/O operations limit. This setting governs how often VMware ESXi switches logical paths. By default, this setting is configured to switch to a new logical path every 1,000 I/O operations. Pure Storage recommends setting this value to 1, which makes VMware ESXi change paths for every single I/O. VMware provides a number of methods to change the I/O operations limit. Best practice: Change the round robin I/O operations limit from 1,000 to 1. To fully utilize the VMware ESXi CPU resource, set the host's active power policy to high performance.
Enhanced Round Robin Load Balancing (Latency-Based PSP)	VMware vSphere 6.7 U1 introduced a sub-policy option for round robin, Enhanced Round Robin Load Balancing, that actively monitors individual path performance. The feature lets VMware ESXi hosts make smarter decisions on which paths to use and which paths to exclude. When enabled, this feature assesses the first 16 user I/O requests and calculates their average latency. Once all paths have been analyzed, the Native Multipathing Plugin calculates the average latency of each path to determine which paths are optimal and which paths are not. If a path is not optimal, the path is not used until latency reaches an optimal latency. After the initial assessment, the ESXi host repeats the analysis process every three minutes. In VMware ESXi 7.0 or later, this sub-policy is automatically enabled. Best practice: Enhanced round robin load balancing is configured by default on VMware ESXi 7.0 or later. VMware administrators should enable it on VMware ESXi 6.7 U1 or later.

VMware ESXi Host Configuration Item	Description and Recommendations
Verifying Connectivity	Prior to introducing production workloads on a VMware ESXi host or volume, VMware administrators should verify proper connectivity using the following steps: **1.** Verify multipath settings in VMware ESXi. **2.** Verify the proper number of paths. **3.** Verify I/O balance and redundancy on the FlashArray. Administrators can accomplish each of these tasks using the VMware vSphere client or the esxcli command-line tool. Storage administrators can monitor I/O balance on the array using the purehost command-line tool.
Disk.DiskMaxIOSize	Disk.DiskMaxIOSize is a VMware ESXi host setting that controls the largest I/O size that the host allows to be sent to the underlying storage device. By default, this setting is 32MB. If an I/O operation is larger than Disk.DiskMaxIOSize, the ESXi host splits the I/O request into multiple segments. The 32MB limit affects older releases of VMware ESXi, but it has been resolved in newer patch releases of ESXi 7.0. Changing Disk.DiskMaxIOSize is only required in the following scenarios: **1.** If a VM is using Extensible Firmware Interface (EFI) instead of BIOS and is using VMware hardware version 12 or earlier **2.** If your environment uses VMware vSphere Replication **3.** If your environment contains VMs that run applications that send READ or WRITE requests larger than 4MB **4.** If your environment uses Fibre Channel with one of the previous scenarios, as this issue does not affect iSCSI Best practice: Upgrade VMware ESXi to a release that adheres to the maximum support SCSI size from the FlashArray.

VMware ESXi Host Configuration Item	Description and Recommendations
VAAI Configuration	The VMware API for Array Integration (VAAI) provides methods to offload and accelerate certain operations in a VMware environment. Pure Storage requires that all VAAI features are enabled on every VMware ESXi host that uses FlashArray storage. Disabling VAAI features can dramatically reduce FlashArray performance and efficiency in VMware vSphere environments. All VAAI features are enabled by default (set to a value of 1), so no action is typically required. VMware administrators can verify and modify these settings through the VMware vSphere client or through CLI tools. Best practice: Keep VAAI enabled by verifying DataMover.HardwareAcceleratedInit, DataMover.HardwareAcceleratedMove, and VMFS3.HardwareAcceleratedLocking are set to a value of 1.  Additionally, Pure Storage recommends enabling VMFS3.UseATSForHBOnVMFS5 using a value of 1. Best practice: Keep VMFS3.UseATSForHBOnVMFS5 using a value of 1. If another vendor is present and prefers this feature to be disabled, Pure Storage supports doing so. 

VMware ESXi Host Configuration Item	Description and Recommendations
iSCSI Configuration	Pure Storage recommends the following iSCSI configuration changes and evaluations: 1. Set the login timeout to a larger value. 2. Disable DelayedACK. 3. Use port binding for ESXi software iSCSI adapters when possible. 4. Enable jumbo frames. 5. Determine whether or not to use Challenge-Handshake Authentication Protocol (CHAP). 6. Determine whether or not to keep default iSCSI failover times or to reduce them for sensitive applications.
Configure Network Time Protocol (NTP)	Enabling NTP helps ensure that all timestamps for servers, arrays, switches, and so forth are aligned and synchronized. Pure Storage recommends enabling NTP on all VMware ESXi hosts. For more information, visit the VMware KB Configuring Network Time Protocol (NTP) on an ESXI host using the vSphere Client. Best practice: Enable NTP on all ESXi hosts.
Remote Syslog Server	Pure Storage recommends using the VMware vRealize Log Insight Open Virtualization Appliance (OVA) that provides syslog integration for ESXi hosts and VMware vCenter. Additionally, administrators can use the Pure Storage Content Pack with VMware vRealize Log Insight, which provides a single logging destination for both the vSphere and Pure Storage environments. VMware administrators can configure VMware vRealize Log Insight using the Log Insight web client. Additional remote syslog options are available for customers who do not have VMware vRealize Log Insight installed or available.

TABLE 7 VMware ESXi host configuration recommendations and best practices for use with Pure Storage FlashArray

Virtual Machine Operating System and Guest Optimization

VMware provides a number of recommendations for both Windows and Linux VM optimizations. Table 8 lists optimizations and best practices that can be found in Virtual Machine and Guest Configuration.

VMware ESXi Host Configuration Item	Description and Recommendations
Virtual Disk Choice	VMware vSphere provides three virtual disk formats that can be used with VMs: 1. Thin virtual disk: This type of virtual disk only allocates storage that is used by the VM guest OS. When a thin disk is created, it only uses one block of storage space. As the guest OS writes data to the virtual disk, new blocks are allocated, zeroed out, then data is committed to storage. Therefore, some latency does exist for new writes. 2. Zeroedthick (lazy) virtual disk: This type of virtual disk allocates all of the storage space required by the virtual disk when it is created. When the guest writes to a specific block on the virtual disk for the first time, the block is zeroed, then the data is committed. As with thin virtual disks, there is some latency for new writes, though the difference is small. As such, there is a negligible performance difference between zeroedthick and thin virtual disks. 3. Eagerzeroedthick: Like zeroedthick virtual disks, this type of virtual disk allocates all storage space required by the virtual disk when it is created, but takes the extra step of zeroing the entire capacity. While the creation process takes longer because the disk cannot be used until zeroing is complete, there is no write latency penalty because allocation and zeroing take place in advance, and not on demand. Best practice: While each of the formats has certain benefits and drawbacks, Pure Storage recommends using thin virtual disks for most VMs and vVols, and eagerzeroedthick virtual disks for VMs stored on VMFS volumes that require high performance levels. Do not use zeroedthick virtual disks, as they do not provide any performance or storage space benefits over thin and eagerzeroedthick virtual disks.

VMware ESXi Host Configuration Item	Description and Recommendations
Virtual Hardware Configuration	Pure Storage recommends the following configuration items for VMs: • Use a dedicated paravirtual SCSI (PVSCSI) adapter for each virtual disk used for MySQL. • Provision dedicated virtual disk(s) for MySQL. In the event that data and log directories are separated, multiple virtual disks will need to be provisioned, each of which should be attached to a dedicated PVSCSI adapter. Use the latest virtual hardware version that the VMware ESXi host supports. • Install the latest VMware tools in all VMs. • Provision vCPUs and memory as applications require. • Do not enable VM encryption in VMware vSphere. While VM encryption is supported by Pure Storage, all data at rest on FlashArray is encrypted by default. Data reduction using deduplication and compression is impossible on VMs that use VM-level encryption. • Use the built-in ESXi IOPS limits if you want to limit a VM's virtual disk IOPS.
Template Configuration	VM templates give VMware administrators the ability to quickly deploy VMs. For example, an organization might create a VM with a specific MySQL configuration that can be used for all new MySQL VMs. When a new VM is created from a template, the template is copied to the datastore specified by the VMware administrator. If a template resides on a storage array that is different from where the VM will reside, the amount of time required to create the VM is increased. Placing templates on the same datastore where VMs will reside lowers the amount of time required to create a VM from a template. Best practice: Place templates on the same FlashArray as the target datastore.
High-IOPS VMs	For VMs that host applications that need IOPS higher than 50,000, change the PVSCSI adapter QD limits from the default of 256 and 64 to 1,024 and 256. Visit the VMware KB for information on how to configure queue limits for both Windows and Linux guests. Best practice: Do not change VM QD limits unless the VM requires higher performance.
File System Configuration	For Linux guests, use the XFS file system for any virtual disks that MySQL data files reside on. The default options are typically all that is required. XFS is the only recommended file system for MySQL data files. Visit MySQL on FlashArray Implementation and Best Practices for more information about mount options. For Windows guests, use the NTFS file system with an allocation size of 64KB. If you are setting the MySQL InnoDB page size from the default of 16KB to 32KB or 64KB, use an NTFS allocation size that matches the InnoDB page size. Visit MySQL on FlashArray Implementation and Best Practices for more details.

TABLE 8 VMware guest configuration recommendations and best practices for use with Pure Storage FlashArray

MySQL Instance Optimization

While SQL statements, client code, and database structure can be optimized by software developers and database administrators, other optimization recommendations and best practices for MySQL instances running in a VMware vSphere and FlashArray environment are available and are listed in Table 9. More information is available on Pure Storage's website in MySQL on FlashArray Implementation and Best Practices and InnoDB Configuration Parameters.

VMware ESXi Host Configuration Item	Description and Recommendations
MySQL Default Storage Engine	By default, MySQL uses the InnoDB storage engine. If InnoDB is not currently set as the default storage engine, add the following to the [mysqld] section of the MySQL server file: `default-storage-engine=innodb` From MySQL 8 onwards InnoDB is the default storage engine and this parameter does not need to be set explicitly.
InnoDB Configuration Parameters	InnoDB has a number of parameters that can control various functional aspects in addition to performance. These parameters are found in the MySQL option files for Linux and Windows. Visit InnoDB Configuration Parameters for a list of Pure Storage parameter recommendations.
MySQL Option File Examples	MySQL uses an option file that controls various aspects of MySQL configuration and performance. Pure Storage provides examples of these option files with recommended parameters and their values for both Linux and Windows.

<table>
<tr><th>VMware ESXi Host Configuration Item</th><th>Description and Recommendations</th></tr>
<tr><td>Volume and File System Architectural Layout</td><td>
Pure Storage recommends using separate virtual disks and PVSCSI controllers for the OS and MySQL database files.
The following table lists typical InnoDB-based MySQL file system layouts for Windows and Linux:
<table>
<tr><th>Windows</th><th>Linux</th></tr>
<tr><td>C:\ProgramData\MySQL\MySQL Server<Vx.x>\Data</td><td>/var/lib/mysql</td></tr>
<tr><td>datadir</td><td>datadir</td></tr>
<tr><td>innodb_data_home_dir</td><td>innodb_data_home_dir</td></tr>
<tr><td>innodb_log_group_home_dir</td><td>innodb_log_group_home_dir</td></tr>
<tr><td>innodb_undo_directory</td><td>innodb_undo_directory</td></tr>
<tr><td>lc_messages_dir</td><td>lc_messages_dir</td></tr>
<tr><td>log_bin_basename</td><td>log_bin_basename</td></tr>
</table>
In most cases, using a separate virtual disk for the MySQL directories is sufficient. However, depending on your use case, you can create separate virtual disks for each of the data, binary, and transaction log directories. This is useful when using applying quotas or quality-of-service (QoS) restrictions on a per-volume basis.
The following table lists a sample layout where each directory is a separate virtual disk. Note that these directory locations must be specified in the MySQL options file.
<table>
<tr><th>Windows</th><th>Linux</th></tr>
<tr><td>datadir = C:/MySQL/Base/Data</td><td>datadir = /mysql/base/data</td></tr>
<tr><td>innodb_data_home_dir = C:/MySQL/Base/Data</td><td>innodb_data_home_dir = /mysql/base/data</td></tr>
<tr><td>innodb_log_group_home_dir = C:/MySQL/Log/Data</td><td>innodb_log_group_home_dir = /mysql/log</td></tr>
<tr><th>Windows</th><th>Linux</th></tr>
<tr><td>innodb_undo_directory = C:/MySQL/Undo/Data</td><td>innodb_undo_directory = /mysql/undo</td></tr>
<tr><td>log_bin_basename = C:/MySQL/BinLog/Data</td><td>log_bin_basename = /mysql/binlog/binlog</td></tr>
</table>
Pure Storage recommends mounting volumes to a base directory, such as C:\ProgramData\MySQL or C:\MySQL on Windows or /var/lib/mysql or /MySQL on Linux before install MySQL using the following permissions:
Windows: NETWORK SERVICE, Full Control
Linux: user mysql, group mysql, drwxr-x--x
</td></tr>
</table>

TABLE 9 MySQL configuration recommendations and best practices for use with Pure Storage FlashArray

FlashArray Data Service: Volume Snapshots and Asynchronous Replication

A feature of Pure Storage FlashArray is the ability to create instantaneous snapshots of block storage volumes, regardless of volume size. These snapshots are immutable point-in-time images of one or more volumes. Snapshots can be created individually, or as part of a protection group, which helps ensure that snapshots of one or more volumes within the protection group are consistent with each other.

FlashArray snapshots provide the following key features:

- **Efficiency:** FlashArray snapshots are thin-provisioned, deduplicated, compressed, and don't require any snapshot capacity reservation. Snapshots also inherit all data-reduction capabilities to reduce snapshot size.
- **No snapshot hierarchy:** FlashArray snapshots do not rely on other snapshots for data consistency, which lets administrators create and destroy snapshots without affecting other snapshots.
- **No performance degradation:** FlashArray snapshots are full volumes that are created instantaneously without any performance degradation.
- **Portability:** FlashArray snapshots can be transferred to one or more FlashArray or Pure Cloud Block Store instances using space-efficient copies accompanied by volume metadata.
- **Rapid recovery:** Administrators can recover data from any snapshot. The data can be recovered directly to the original volume the snapshot was created from, or it can be recovered to a new volume.

FlashArray supports asynchronous replication of volume snapshots between FlashArray devices. With asynchronous transfers, only data that does not exist on the target FlashArray is transferred. This replication works well for applications that require a recovery point objective (RPO) as low as five minutes, or for data mobility scenarios that require data to be copied to multiple sites.

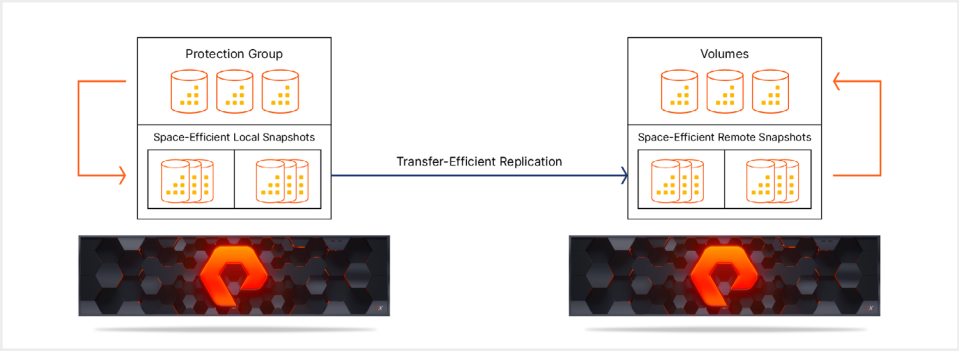


FIGURE 7 Asynchronous snapshot replication only transfers data that does not exist on the target FlashArray

Using Volume Snapshots with Virtualized MySQL Databases

As MySQL deployments grow, organizations might want to use horizontal scaling to increase MySQL performance and availability. In this scenario, change operations can be directed to a primary database, while read operations are directed to secondary database replicas within the MySQL environment. Such scaling enables better query response times by spreading loads across multiple servers. Additionally, deploying replicas to multiple secondary servers provides high availability such that if a primary server goes offline, secondary servers can continue to service queries.

Horizontal scaling requires database replicas to be synchronized across multiple servers. Any changes to a replicated database's object structures, such as tables, views, and indexes, are synchronized across all of the replicas. MySQL offers replication capabilities that can keep all replicas synchronized while providing high availability, disaster recovery, and scalability.

MySQL replication technologies include:

- **Primary/replica replication:** This replication technology captures database changes from a primary source database and applies them to secondary replicas.
- **Group replication:** This replication technology is provided as a MySQL plugin. Group replication can operate with multiple primary systems or a single primary system with automatic election. Only the InnoDB storage is supported by group replication.
- **Galera Cluste**r: Galera Cluster provides virtually synchronous replication between three or more nodes, with each node in the cluster being a primary node. Only the InnoDB storage is supported by Galera Cluster, and it is only available on Linux-based MySQL servers.

When additional replicas are added to a replication topology, each replica will require an initial data synchronization, or initial seeding, from a source database. The data synchronization method is determined by which replication technology is used:

- Primary/replica replication uses mysqldump or a physical file copy method for the initial data transfer.
- Group replication uses mysqldump, binary log replication, or the clone plugin for the initial data transfer.
- Galera Cluster replication uses mysqldump, rsync, clone, or XtraBackup for the initial data transfer.

The initial seeding process has several limitations that are dependent on the size and level of activity on the primary database, including:

- **Transfer time:** Large databases can take a considerable amount of time to transfer, which impacts the time required to make a replica available.
- **Database blocking:** Some initial seeding techniques block the primary database from making updates during the transfer, which can negatively impact both the availability of the primary and replica.
- **Heavy usage delays:** In heavily loaded environments, the replicas might struggle to synchronize with the rapidly changing database state on the primary.
- **Single primary/replica seeding**: Some techniques can only seed one replica from one primary at a time, which can slow down the process when multiple replicas are required.

Each of these limitations can negatively impact applications and business processes that rely on MySQL and hamper scaling efforts.

Administrators can use FlashArray snapshots to create recovery points for MySQL instances and clone MySQL data to reduce seeding time for new replicas. Adding a new replica only requires creating a volume snapshot of the primary MySQL instance. The snapshot can then be copied to a new or existing volume that will be used by the replica.

Note that to ensure database and snapshot consistency, MySQL database tables must be flushed from the table cache and locked by running the flush tables with read lock SQL statement, and then optionally flushing the logs with the flush logs statement. These statements halt all database operations and are only available on the InnoDB storage engine. For more information, see the Application Consistency section in Creating Volume Snapshot for MySQL on FlashArray.

Virtual Disks in VMFS Datastores

If the database files reside in one or more virtual disks located on one or more VMFS datastores, the following recommendations should be followed when using volume snapshots:

- Only virtual disks for a single database instance should be stored on the VMFS datastore.
- Larger databases should be isolated onto their own virtual disks and VMFS datastores. These databases should not share VMFS datastores with other databases, even from the same instance.
- Virtual disks used with volume and filesystem layouts that separate out instance components can be located on separate VMFS datastores, but snapshots need to be created from a protection group that contains all applicable volumes to preserve consistency.

Example: Creating a Volume Snapshot for a Database Instance with a Virtual Disk on a Single VMFS Datastore

The following example consists of a MySQL database that uses the default volume and file system layout, a single virtual disk within a single VMFS datastore, and a crash-consistent volume snapshot of the underlying volume. This database instance is installed on Linux and is configured to use the /var/lib/mysql data directory. A virtual disk formatted with the XFS filesystem is mounted at this location. Figure 8 shows the virtual disk configuration for the data directory. The virtual disks associated with the MySQL instance are located on a single VMFS datastore, "MySQL-Linux-01."

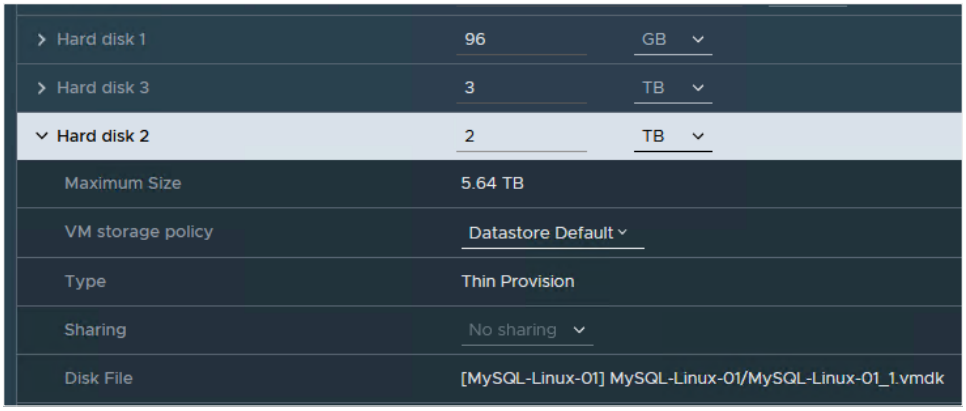
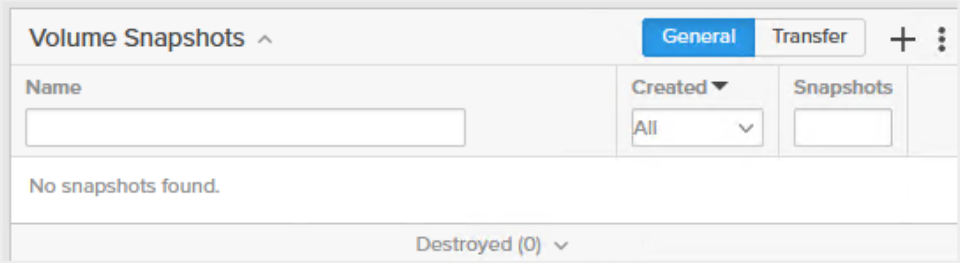


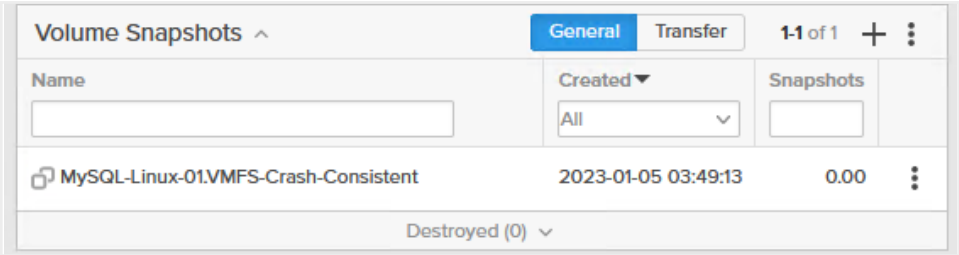
FIGURE 8 MySQL virtual disk layout and datastore

To create a volume snapshot:

1. In the FlashArray UI, navigate to the volume that contains the MySQL virtual disks.
2. In the **Volume Snapshots** section, select the **+** to create a volume snapshot.



3. In the **Create Snapshot** dialog box, enter an optional suffix for the snapshot to identify it, and then select **Create**. The volume snapshot appears in the volume snapshot list in the **Volume Snapshots** section.

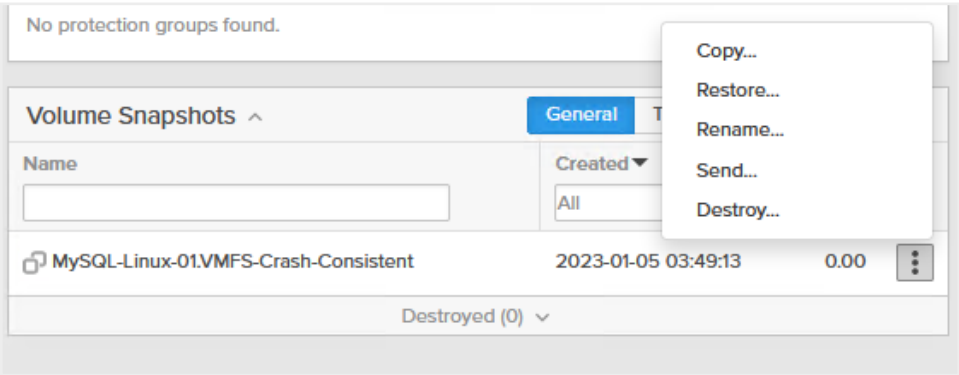


Example: Recovering a Volume Snapshot for a Single Database with Virtual Disks on a Single VMFS Datastore

Using the snapshot created in the previous example, the following steps demonstrate how a MySQL instance can be recovered to the same system using volume snapshots.

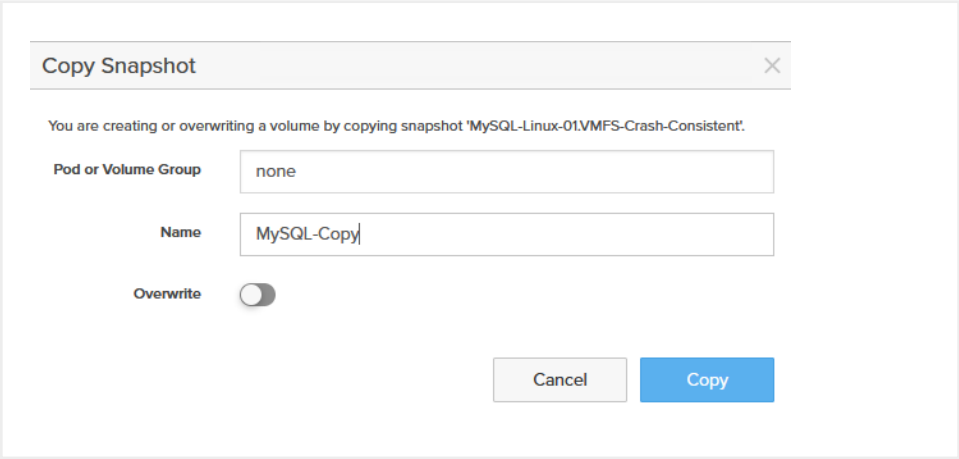
To recover a volume snapshot:

1. In the FlashArray UI, navigate to the volume that contains the MySQL virtual disks.
2. Identify the snapshot to recover, and then select the three ellipses next to it to display the context menu. In this context menu, select **Copy**.

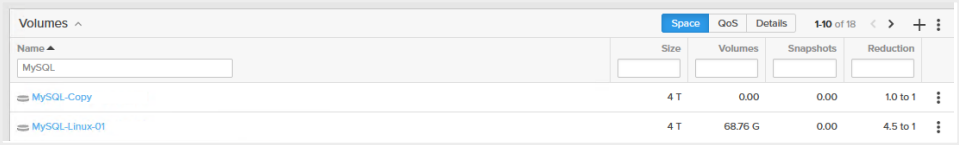


3. Caution: Restore is also an option, but it overwrites the original volume.

4. In the Copy Snapshot dialog box, enter a name for the new volume, and then click **Copy**.



5. The snapshot volume copy appears in the volume view, and can now be attached to an ESXi host.

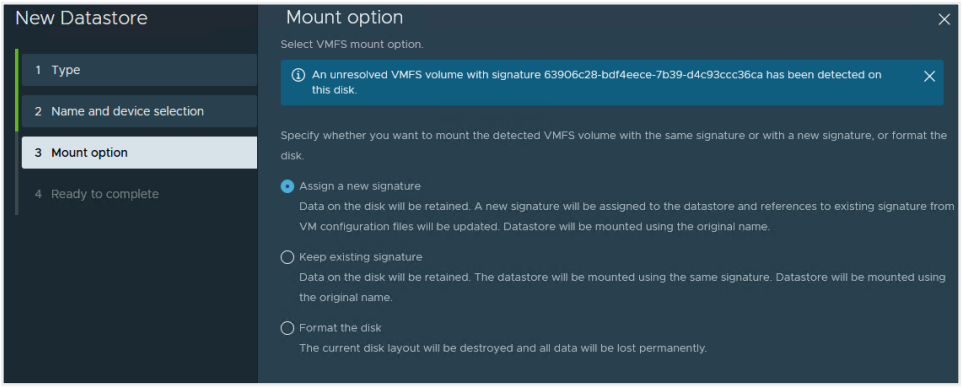


6. A volume that is a clone of a VMFS datastore must be re-signatured prior to accessing its contents. The process of re-signaturing is described in detail in Resignaturing Datastores.

7. After the volume has been attached to ESXi hosts in vCenter, the hosts must scan for new storage devices and VMFS volumes.

8. Once the scan completes, launch the New Datastore wizard in vCenter to perform the re-signature process.

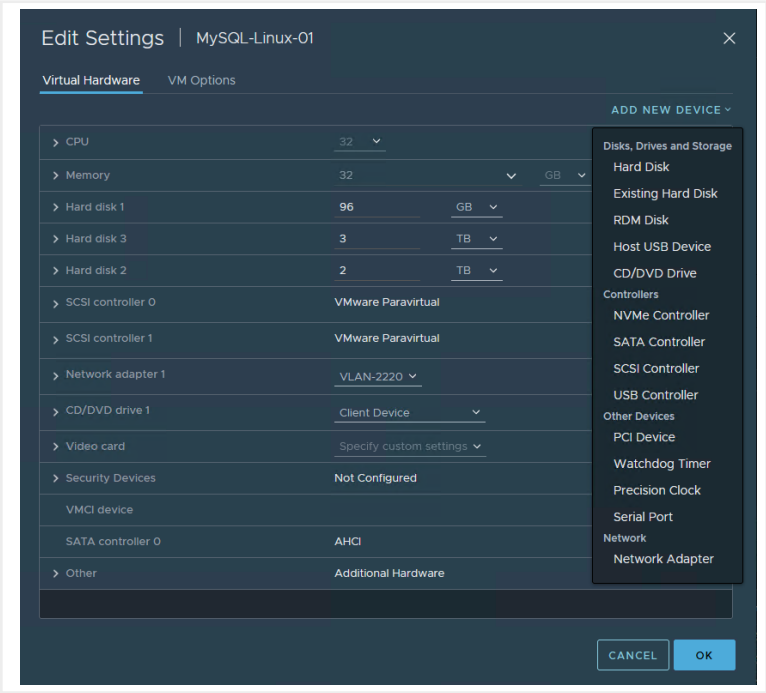
9. On the Mount option step, select the Assign a new Signature option. Do not format the disk, as this will erase all existing data.



10. In the vCenter client, navigate to the datastore browser and identify the virtual disks in the re-signatured snapshot volume copy. Make note of their names and location.

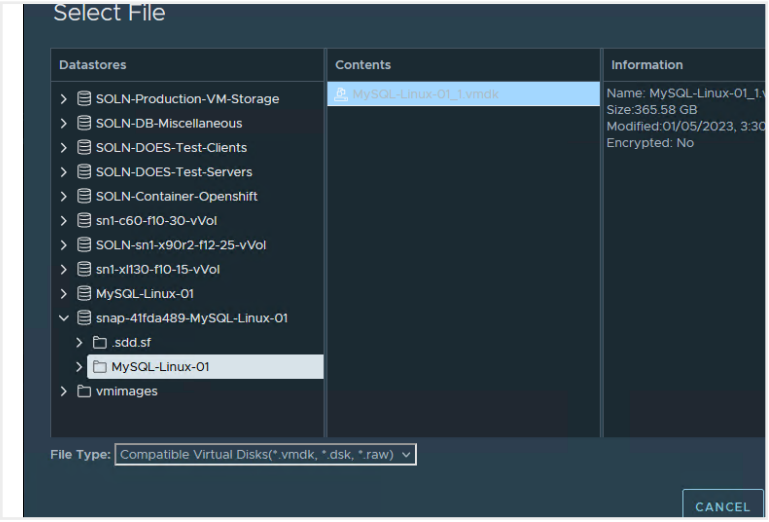
11. Navigate to the existing VM that you want to attach the virtual disks to, and then select Edit Settings from the Action menu.

12. Select Add New Device, and then select Existing Hard Disk from the drop-down menu.



13. In the Select File dialog, navigate to the cloned datastore.

14. Select the snapshot volume copy, and then select **OK**.



15. Once all of the disks have been added, select OK.

If you are adding the cloned virtual disks back to the same VM, the error message “disks with same UUID should not be assigned to a VM” might appear. To work around this issue, follow the steps in this VMware knowledge article to assign a new UUID to the cloned virtual disks.

After the virtual disks have been added, they will show up as devices with file systems that can be mounted.

Virtual Disks in vVols

The process for creating volume snapshots of virtual disks in vVols is similar to creating VMFS volume snapshots.

Example: Creating Volume Snapshots of Database Files in vVols

This example uses the same database layout as in the previous VMFS snapshot example. The virtual disks on which the MySQL data area resides is migrated with Storage vMotion from the VMFS datastore to the vVol datastore. The result of this move is that a volume group on FlashArray is created that contains all of the relevant volumes, one of which is the virtual disk containing the MySQL instances data files.

This volume can be used to create a snapshot following a similar process as the VMFS datastore. The differences when compared to snapshots of VMFS datastores are that the volume can be copied, cloned, or moved between physical and virtual systems, no resignaturing is necessary, and that recovery will only require that the destination virtual disks already exist in a virtual volume datastore.

FlashArray Data Service: ActiveDR Continuous Replication

ActiveDR is a technology included with every FlashArray that can asynchronously replicate databases to a remote site as part of a disaster recovery (DR) strategy. ActiveDR can replicate data to other FlashArray deployments across long distances, while still retaining low RPOs. Figure 9 shows a typical ActiveDR configuration.

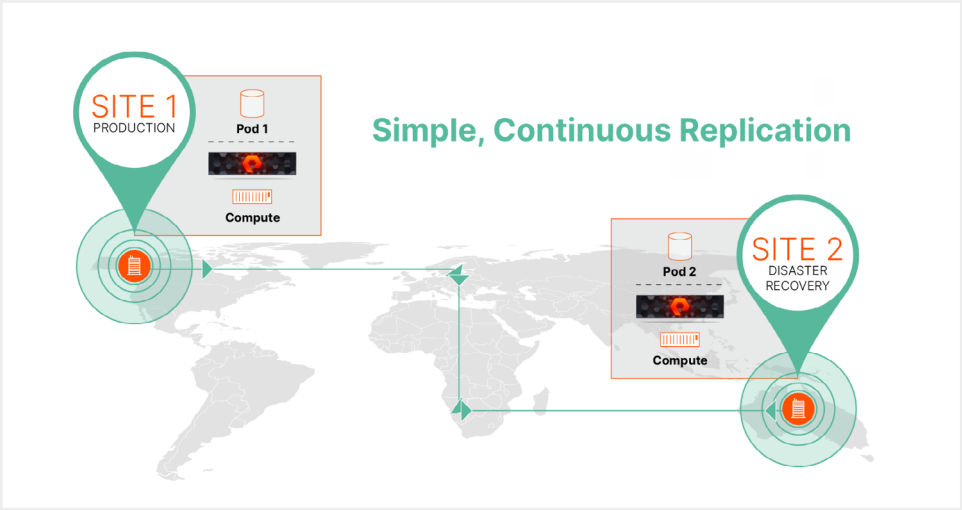


FIGURE 9 ActiveDR can continuously replicate MySQL data to remote DR sites

Synchronous replication technologies such as Pure Storage ActiveCluster provide high availability, but they can require network latency between the FlashArray environments to be as low as possible because the FlashArray at the secondary site must acknowledge that data has been received before the primary site can continue. ActiveDR doesn't require an acknowledgement from the secondary site, which eliminates the need for a low-latency network and allows continuous replication over longer distances.

Managing ActiveDR is simple, which makes it useful across business requirements that require a near-zero RPO. Some of the simplified management aspects include:

- **Pod replication**: Pods are management containers for volumes that use ActiveCluster or ActiveDR and that provide a simple management construct for organizing data volumes and their settings into groups. When pods are linked together on separate systems using a replica link, the data in the source pod automatically starts replicating to the secondary pod.
- **Continuous change tracking**: ActiveDR automatically manages changes without the need to provision or monitor journal devices.
- **Single-command failover**: ActiveDR makes it simple to implement, test, and manage disaster recovery. To minimize risk, orchestration steps for the entire environment stay the same during the test or in an actual failover event.
- **Multi-direction replication**: Pods can replicate in multiple directions between two FlashArray environments.

Configuring ActiveDR in a VMware vSphere Environment

Pure Storage FlashArray uses the continuous replication of volumes between Pure Cloud Block Store instances to enable ActiveDR. In this context, ActiveDR uses pods that contain all of the volumes that are to be protected. Each pod is a replication group and a consistency group, and all volumes within a pod fail over together.

In a VMware vSphere environment, the following guidelines for ActiveDR in a VMware vSphere environment should be followed:

- If a VM spans multiple datastores, all of the VM's datastores should be in the same pod, so as to ensure that the VM can be entirely failed over and its storage is consistent at the recovery site.
- A VM that uses both a VMFS datastore or a Raw device mapping (RDM) should contain both in the same pod, so as to ensure that the VM can be entirely failed over and its storage is consistent at the recovery site.
- If ActiveDR datastores are in use in a VMware vSphere storage DRS cluster, all of the datastores should be in the same pod.
- Related applications that need to be recovered at the same time should all reside in the same pod. This ensures cross-application data consistency after a failover.
- The ESXi host personality must be set on all FlashArray objects that represent ESXi hosts to ensure that the ESXi hostd process doesn't lock up trying to update VMFS metadata on demoted read-only ActiveDR pods. For more information, see Setting the FlashArray ESXi Host Personality.

Keep the following in mind when working with VAAI and ActiveDR pods:

- XCOPY does not work across pod boundaries. Normal host copy processes occur when performing Storage vMotion, Clone, or Deploy from Template operations when the source datastore is not in a pod or is not a pod, and the target is in a different pod.
- UNMAP, ATS, and WRITE SAME work within pods.
- UNMAP operations that occur will reclaim space on source volumes. The UNMAP is transmitted to the target volume in the target pod, but the reclamation does not occur immediately after the UNMAP process executes.

Using ActiveDR and VMware vSphere together are subject to the following limitations:

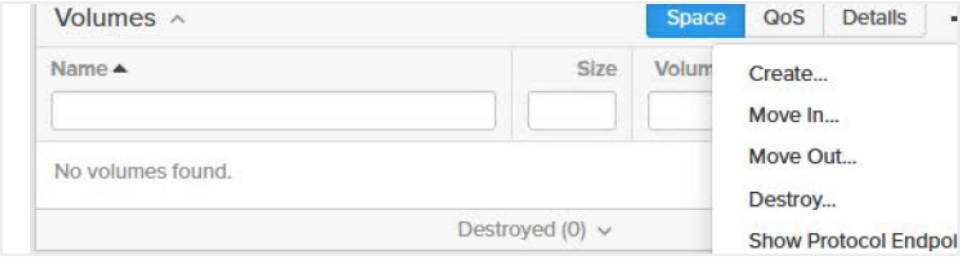
- FlashArray volumes in use as vSphere vVols are not currently supported and cannot be protected with ActiveDR replication.
- The Pure Storage vRealize Orchestrator plugin does not have built-in workflows for ActiveDR.
- The Pure Storage vRealize Operations Manager Management Pack does not currently support ActiveDR replication relationships, alerts, or metrics.
- ActiveDR provisioning in the Pure Storage FlashArray plugin for the vSphere client is only supported with plugin version 4.4.0 or higher.
- VMware Site Recovery Manager support of VMFS or RDMs protected by ActiveDR is only supported with FlashArray SRA version 4.0.0 or higher.

Configuring an ActiveDR Pod

The following example uses an existing MySQL data volume that will be moved into a newly-created pod. The first step is to create an ActiveDR pod and add the existing MySQL data volume to it.

To create an ActiveDR pod and add a MySQL data volume:

1. From the Pure Storage GUI, select Storage in the navigation pane, and then select the Pods tab.
2. Click the **+** icon in the Pods group to create a new pod.
3. Enter a name for the pod in the Name field in the Create Pod dialog box, and then click **Create**. The pod appears in the Pods group.
4. In the Pods group, click the name of the pod you just created. The Pod management view displays.
5. In the Pod management view, locate the Volumes group, click the ellipses, and then click **Move In** to display the Move Volumes In dialog box.

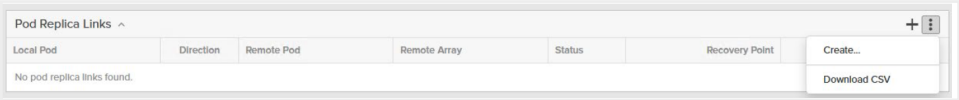


6. Select the MySQL data volume you want to move into the pod, and then select **Move**. The volume appears in the Volumes group using a new naming convention: `<pod name>::<volume name>`.

The next step is to enable ActiveDR for the volumes that have been added by creating a replica link to a target instance.

To create a replica link:

1. In the Pod Replica Links group, select the ellipses, and then click **Create**.



2. In the Create Replica Link dialog box, select a remote FlashArray from the Remote Array drop-down list, and then select a remote pod.
   Note: If no remote pod exists on the target FlashArray, select Create Remote Pod, enter a pod name in the Name field, and then select **OK**.
3. Select **Create**.

Once you have created a replica link, the source pod begins an initial baseline replication of the volume to the target pod, and the Status field in the Pod Replica Links group displays “baselining” next to the pod link. Once the baseline replication is complete, the Status field displays “replicating.”

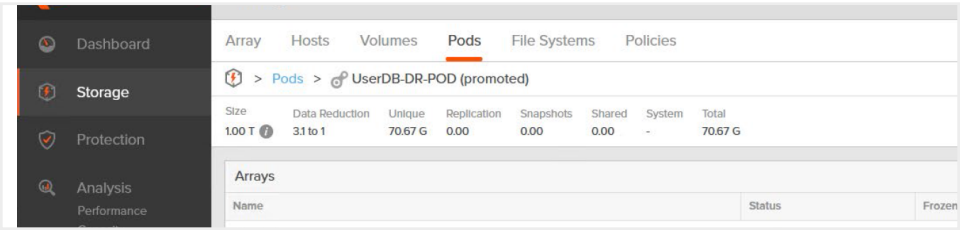
While the source pod is replicating to the target pod, the source pod remains in a promoted state that permits read and write operations, while the target pod remains in a demoted state that only permits read-only operations. A demoted pod’s volumes can be connected to a host, but they should not be mounted or set as offline until a recovery process is required.

Failing Over to a Remote Pod

To fail over to a remote pod, the remote pod must be promoted and the database volume attached to a remote MySQL instance. If the source pod is still available and replicating, the pod must first be demoted before promoting the remote pod on the target to protect against data loss. Note that downtime is required to fail over a MySQL workload to a remote pod.

To demote the source pod and promote the target pod:

1. In the Pod management view, select the ellipses, and then select Demote.



2. From the Demote dialog, select Quiesce, and then select Demote. The source pod is demoted and placed in a read-only state.
3. In the Pod management view, select the ellipses, and then select Promote. The Promote dialog box appears.
4. Select Promote to promote the target pod.

In the Pod Replica Links group, the target pod shows that it is in a promoted state, while the source pod shows that it is in a demoted state. Volumes on the promoted target pod can now be attached to a remote MySQL VM, and MySQL can start.

A failback is the reverse of this process, using the following steps:

1. If the source pod is available, verify that the promoted target pod is replicating to the demoted source pod.
2. Stop MySQL on the remote MySQL VM and unmount the target volume.
3. Demote the target pod using the Quiesce option.
4. Promote the source pod.
5. Once the source pod is in a promoted read-write state, attach the MySQL data volume to a VM and restart MySQL.

FlashArray Data Service: ActiveCluster High Availability

ActiveCluster is a storage volume–based synchronous replication technology included with every FlashArray that provides high-availability data services. This service provides that ability to protect the volumes on which MySQL data files reside and helps ensure they remain available in the event of a site or array failure. ActiveCluster also lets organizations configure zero RPOs and recovery time objectives (RTOs) between two FlashArray environments with active/active bi-directional synchronous replication and transparent failover.

ActiveCluster uses the following components:

- **Pure1® Cloud Mediator**: This component determines which array will continue data services should an outage occur. Pure Storage can also provide an on-premises mediator VM.
- **Active/active clustered array pairs**: These FlashArray pairs use synchronous replication to maintain a copy of the protected data on each array and present both copies as one consistent copy to hosts attached to one or both arrays.
- **Stretched storage containers**: Management containers that collect storage objects such as volumes are stretched between two arrays. Stretched storage containers also provide consistent I/O behavior for the storage objects within them.

FIGURE 10 Synchronous active/active replication with two FlashArray environments

Configuring ActiveCluster in a VMware vSphere environment is beyond the scope of this RA. For more information about using ActiveCluster in a VMware vSphere environment, please visit the following:

- ActiveCluster VMware vMSC (51656) Knowledge Base article
- ActiveCluster with VMware User Guide

Protecting MySQL Databases with ActiveCluster

ActiveCluster protects the availability of the FlashArray volumes that contain the VMware vSphere datastores and virtual disks used by MySQL VMs. While ActiveCluster provides a high level of protection, it requires a considerable amount of overhead for synchronous replication. Pure Storage recommends the following:

- Ensure that the MySQL VM has dedicated separate virtual disks for each of the data, binary, and transaction log directories.
- Do not share the file systems and virtual disks among multiple MySQL databases.
- For VMFS datastores, ensure that the virtual disks for individual databases do not share datastores. The virtual disks used for a single datastore should contain the data, binary, and transaction log directories for a single database. It is possible to share datastores where multiple databases need to have the same availability level. Putting the virtual disk in separate locations allows for more granular control at the virtual disk level.

To configure ActiveCluster with VMFS or pass-through RDM (pRDM) datastores, see Implementing vSphere Metro Storage Cluster with ActiveCluster: Configuring ActiveCluster.

Purity//FA does not currently support vVol objects in FlashArray pods. Additionally, VMware does not support stretched storage with vVols. Therefore, vVols are not supported with ActiveCluster.

Pure Storage Solutions for VMware vSphere

Pure Storage offers several storage-based solutions for VMware vSphere. More information about these solutions is available in the Pure Storage Resource Center.

Pure Storage FlashArray Plugin for the vSphere Client

The Pure Storage FlashArray plugin for the VMware vSphere client provides the ability to configure FlashArray host groups, configure iSCSI, create and manage Virtual Machine File System (VMFS) datastores, and configure and use VMware vSphere Virtual Volumes (vVols).

For more information about the plugin, visit Pure Storage Support site or the VMware Marketplace.

Pure Storage FlashArray Storage Replicator Adapter for VMware Site Recovery Manager

The Pure Storage FlashArray Storage Replication Adapter (SRA) for VMware Site Recovery Manager ensures simple and automated recovery of VMs across sites with minimal or no downtime. Additionally, the solution centralizes recovery plans for thousands of VMs. When combined with the Pure ActiveCluster and Pure ActiveDR solutions, the SRA delivers non-disruptive recovery testing and automated workflows for failover, migration, and failback. Pure Storage's suite of modern data-protection capabilities requires no third-site mediation, no extra infrastructure, no extra licenses, no additional fees, and no more than a few minutes to configure.

For more information about the adapter, visit Pure Storage Support or the VMware Marketplace.

VM Analytics in Pure1 Cloud-Based Management

Pure1 is a cloud-based as-a-service data-management platform. Storage administrators can use Pure1 for common and complex data-management tasks for Pure Storage products such as FlashArray, FlashBlade®, and Portworx®. With Pure1, outcomes can be delivered in seconds instead of hours or days, and costly downtime can be eliminated by making use of predictive analytics to respond to dynamic changes from anywhere in the world.

Pure1 provides the following benefits for modern data-infrastructure management:

- **Centralized setup and monitoring:** Setting up Pure1 is easy: Administrators simply log in to the Pure1 portal, and the software does the rest. As soon as a system is online, the Pure1 Meta® cloud-based management feature is hard at work gathering analytics. Live monitoring is available within minutes and accessible from anywhere in the world.
- **Full-stack analysis:** Storage administrators can access critical information about the health and functioning of an entire stack, including predictive fault analysis and alerting. Auditing for ransomware protection is also included, which helps with the investigation of vulnerabilities in an environment.
- **Reporting:** Pure1 has an intuitive, built-in reporting engine that storage administrators can use to generate shareable reports on commonly requested information such as capacity, performance, or even service subscription status.

FIGURE 11 Pure1 helps organizations manage and analyze Pure Storage FlashArray deployments across data centers

Pure1 extends the visibility of its deep analytics on storage infrastructure up the stack to provide performance metrics on volumes and VMs in a VMware vSphere environment. This visibility enables fast and efficient troubleshooting through the stack, with insight into the latency, bandwidth, and IOPS of the objects in the environment to help storage administrators discover problems and solve issues quickly. Support is provided for VMFS, vVols, NFS, and VMware vSAN, with comprehensive analytics and visibility into virtual environments.

Summary

Pure Storage provides a high-performance, reliable environment to run MySQL workloads on VMware vSphere. This solution provides a proven, integrated platform for scaling MySQL and providing robust cloning, replication, and disaster recovery capabilities in any size of environment.

Organizations can use the suggestions and best practices in this reference architecture to plan their MySQL deployment while optimizing the various components within a combined VMware vSphere and Pure Storage FlashArray environment. Further information is available on the Pure Storage website. Additional guidance is available through Pure Storage Support.

PS2386-01 03/2023