Dell EMC SC Series Arrays with MySQL

Abstract

Best practices, configuration options, and sizing guidelines for Dell EMC™ SC Series storage in Fibre Channel environments when deploying MySQL®.

August 2018

Dell EMC Best Practices

Revisions

Date	Description
April 2016	Initial release
August 2018	Content and format change

Acknowledgements

This document was produced by the following members of the Dell EMC storage engineering team:

Author: Mark Tomczik

Table of contents

Audience

This document is intended for information technology professionals seeking to integrate MySQL® with Dell EMC™ SC Series storage.

Prerequisites

Readers should have prior experience with or training in the following:
- RAID, Fibre Channel, multipathing, and IP networking administration
- Operation and configuration of SC Series storage arrays
- Using and administrating Oracle® Linux® and MySQL

DELLEMC

1 Introduction

According to the DB-Engines April 2016 ranking, MySQL™ is the world's second most popular database and most popular open-source database. In addition, it is used in a wide variety of industries (see MySQL Customers by Industry). To deliver cost-effective, high-performance, and scalable solutions, MySQL can be coupled with Dell EMC™ SC Series storage. SC Series arrays help resolve many of the storage challenges facing IT organizations by providing advanced features like dynamic block architecture, storage virtualization, Dynamic Capacity (thin provisioning), snapshots (replays), Data Progression (automatic tiering), and Fast Track.

This document provides best practices, recommended settings, tips, configuration, and examples for integrating MySQL with SC Series arrays to create manageable, reliable, and scalable solutions. This paper focuses on InnoDB and MyISAM storage engines within MySQL. Because application design and requirements vary greatly, the recommendations presented in this document may not be relevant to all environments, and therefore should be evaluated against and adjusted accordingly to meet specific business needs.

This document should not be considered an exhaustive or authoritative source on MySQL or SC Series storage. For detailed information on MySQL, see the MySQL 5.7 Reference Manual. For information on SC Series arrays, see the *Dell Storage Center System Manager Administrator's Guide* available on the Knowledge Center at the SC Series Customer Portal (login required).

Beyond the scope of this document are MySQL performance tuning, multiple MySQL server environments, replication, detail on database backups, and all MySQL editions other than the Community edition.

Note: The information presented in this document is demonstrated using the Community Edition of MySQL 5.7, Oracle Unbreakable Linux kernel 6.5, and Dell SC8000 storage with Storage Center OS (SCOS) version 6.7. Unless otherwise noted, all references to MySQL in this document refer to MySQL 5.7 Community Edition, references to Linux refer to Oracle Unbreakable Linux kernel 6.5, and references to SC Series storage refer to the Dell SC8000 with SCOS version 6.7. The information and concepts presented in this guide should be portable to other operating system versions, and other editions and versions of MySQL.

---

2 Fibre Channel connectivity

SC Series arrays have been tested to work with Dell™, Emulex®, and QLogic® HBA cards. They also support simultaneous transport protocols including Fibre Channel (FC), iSCSI, and FCoE. Although this document was created for FC environments with QLogic HBAs, much of it should also apply to iSCSI and FCoE. For information on other transport protocols supported by SC Series arrays, see the *Dell Storage Manager Administrator's Guide* available on the [Dell Support website](). The [*Dell EMC Storage Compatibility Matrix*]() provides a list of compatible arrays and switches.

To maximize throughput and HA protection, Dell EMC recommends using at least two 8 Gbps quad-port HBAs, or at least two 16 Gbps dual- or quad-port HBAs in database servers, and two 16 Gbps quad-port FC HBAs in the array. Additional HBAs may be required in the server for greater throughput requirements.

2.1 Zoning

Zoning that delivers port-level redundancy and high availability in the event of a SAN component outage is recommended. One way to provide this level of service is to implement the following:

- Multiple fabrics
- Multiple Fibre Channel switches (one or more switches per fabric VSAN and SC fault domain)
- SC Series dual controllers
- Multiple Fibre Channel HBAs per SC Series controller
- SC Series virtual port mode
- Multiple dual- or quad-port 16Gb Fibre Channel server HBAs per server
- Server multipathing
- Soft zones

The example in Figure 1 shows four FC zones, one per server HBA port. Zones in each of the fabrics share the same targets, but have a different initiator.
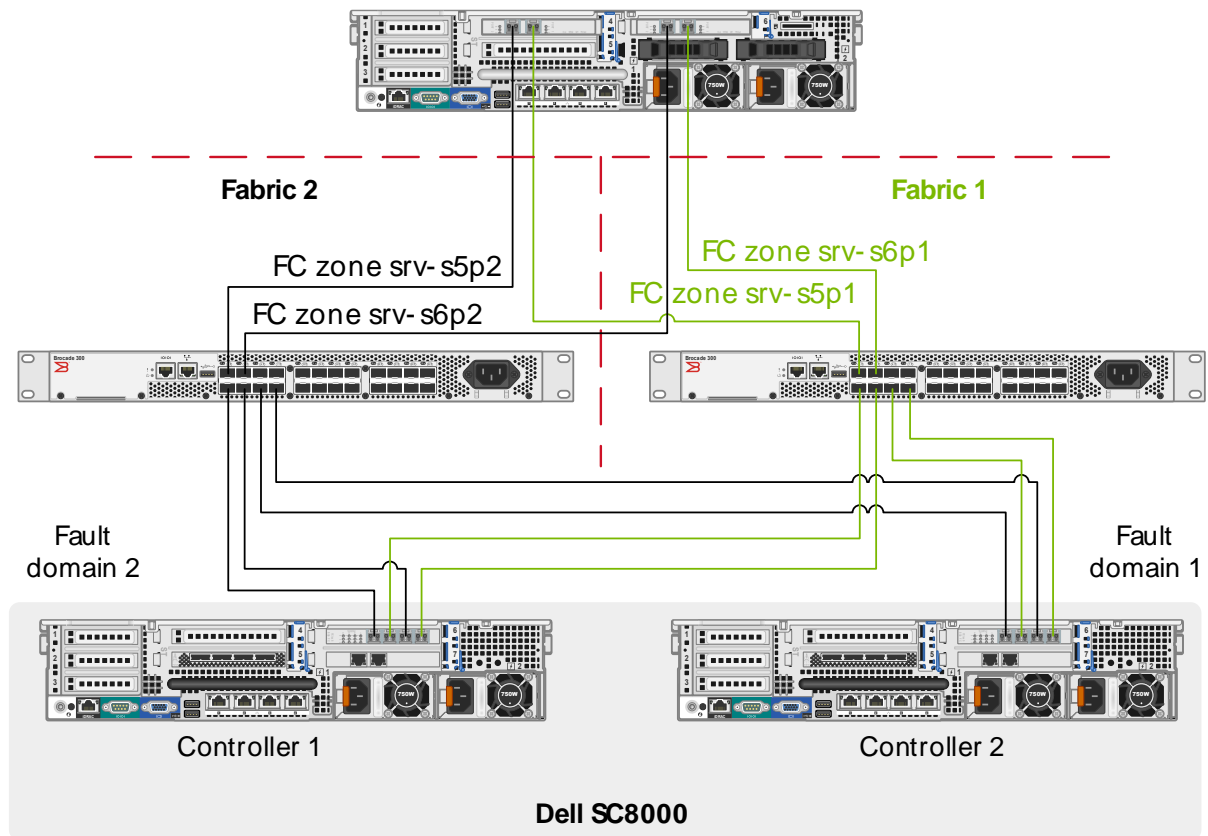
Figure 1　Two FC fabrics

2.1.1　Soft (WWN) zoning

Dell EMC recommends using soft zoning rather than hard zoning. Zones must be created to use a single initiator host port and multiple SC Series virtual ports. Soft zoning is based on the WWN of the initiator and targets. It provides the ability to move the server or the SC Series array to another switch without updating any zone. SC Series virtual port mode requires N_Port ID Virtualization (NPIV) be enabled on the attached FC switch. A soft zone (zone srv-s5p1 shown in Figure 2) may look like the following in Brocade® switchExplorer:

WWN of physical port 1 of the server HBA in slot 5

WWNs of controller 1 virtual ports for the front-end primary ports

WWNs of controller 2 virtual ports for the front-end primary ports
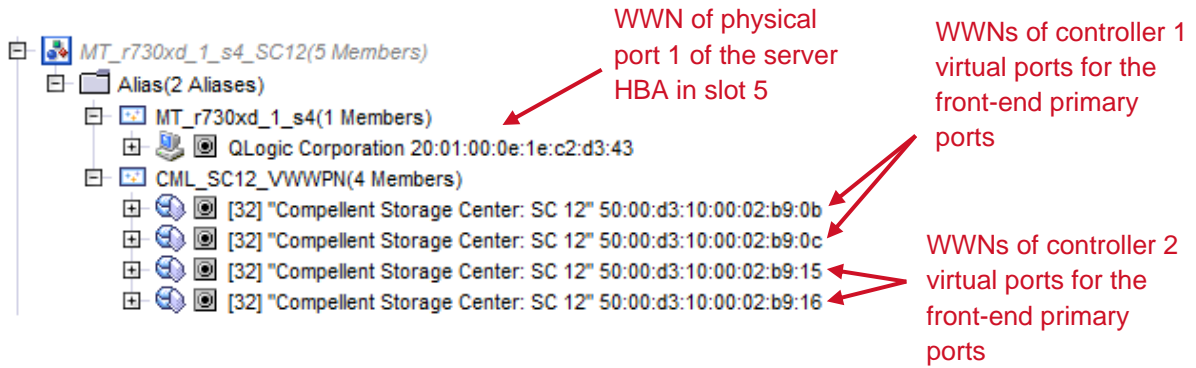
Figure 2   Soft zone

When soft zoning with multiple controllers in virtual port mode, the ports from each controller must be equally distributed between the two fabrics. It does not matter which ports are chosen for a fabric if the ports are chosen in a similar manner for all zones in the same fabrics. Adhering to a cabling and zoning standard will create an easily maintainable fabric. Zoning examples in this document use odd SC controller ports for fabric 1, and even SC controller ports for fabric 2.

When soft zoning with multiple controllers in virtual port mode, create the following zones in both fabrics:

- A zone that includes all odd-numbered physical ports from both controllers in one fabric, and a zone that includes all even-numbered physical ports from both controllers in the other fabric. For example: one zone could have ports 1 and 3, the other zone could have ports 2, and 4.
- A zone that includes the virtual ports of all the odd-numbered physical ports from both controllers in one fabric, and a zone that includes the virtual ports of all the even-numbered physical ports from both controllers in the other fabric.
- A zone for each server HBA port that will connect to the SC Series array. The zone must include the WWNs of the HBA port and the multiple SC virtual ports that are on the same FC switch. Virtual ports need to be equally divided between the zones based on their physical ports, (for example, odd ports in one zone, even ports in the other zone). The following example uses two dual-port HBA server adapters as illustrated in Figure 1.
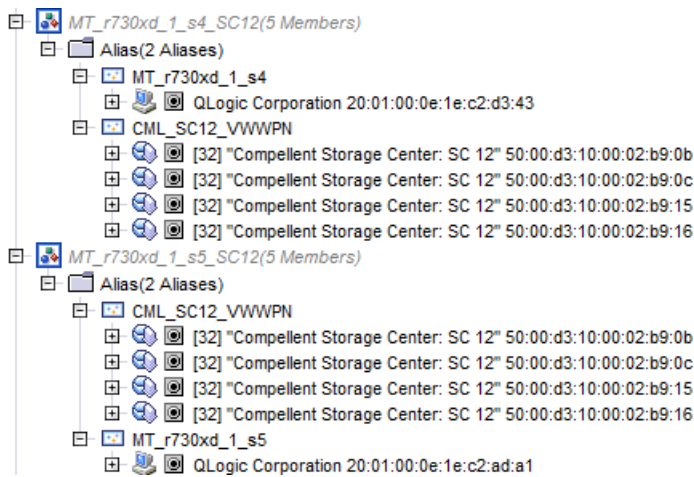
Figure 3   Two zones (one for port 1 from each HBA) for fabric 1

```
☐ 🔧 MT_r730xd_1_s4_SC12(5 Members)
   ☐ 📁 Alias(2 Aliases)
      ☐ 🔲 CML_SC12_VWWPN
         ⊞ 🔵 ◉ [32] "Compellent Storage Center: SC 12" 50:00:d3:10:00:02:b9:09
         ⊞ 🔵 ◉ [32] "Compellent Storage Center: SC 12" 50:00:d3:10:00:02:b9:0a
         ⊞ 🔵 ◉ [32] "Compellent Storage Center: SC 12" 50:00:d3:10:00:02:b9:13
         ⊞ 🔵 ◉ [32] "Compellent Storage Center: SC 12" 50:00:d3:10:00:02:b9:14
      ☐ 🔲 MT_r730xd_1_s4
         ⊞ 🔧 ◉ QLogic Corporation 20:01:00:0e:1e:c2:d3:42
☐ 🔧 MT_r730xd_1_s5_SC12(5 Members)
   ☐ 📁 Alias(2 Aliases)
      ☐ 🔲 CML_SC12_VWWPN
         ⊞ 🔵 ◉ [32] "Compellent Storage Center: SC 12" 50:00:d3:10:00:02:b9:09
         ⊞ 🔵 ◉ [32] "Compellent Storage Center: SC 12" 50:00:d3:10:00:02:b9:0a
         ⊞ 🔵 ◉ [32] "Compellent Storage Center: SC 12" 50:00:d3:10:00:02:b9:13
         ⊞ 🔵 ◉ [32] "Compellent Storage Center: SC 12" 50:00:d3:10:00:02:b9:14
      ☐ 🔲 MT_r730xd_1_s5
         ⊞ 🔧 ◉ QLogic Corporation 20:01:00:0e:1e:c2:ad:a0
```

Figure 4   Two zones (one for port 1 from each HBA) for fabric 2

Table 1   Server HBA ports and SC Series controller front-end ports for zones

FC Zone	Fabric	Server HBA	Server HBA port	SC Series controller	SC Series controller ports
1	1	1	1	1	1, 2 or 1, 3
2	1	2	1	2	1, 2 or 1, 3
3	2	1	2	1	1, 2 or 1, 3
4	2	2	2	2	1, 2 or 1, 3

Table 2   Soft zone example

FC zone	Fabric	WWN	Description
MT_r730xd_1_s4	1	2001000e1ec2d343	Server HBA 1, port 1
		5000d3100002b90b	Controller 1 front-end virtual port 1
		5000d3100002b90c	Controller 1 front-end virtual port 3
		5000d3100002b915	Controller 2 front-end virtual port 1
		5000d3100002b916	Controller 2 front-end virtual port 3

2.1.2   Hard (port) zoning

Hard zoning is based on defining specific ports in the zone. Because the zone is based on ports, if the server or SC Series array is moved to a different port or switch, the fabric will require an update. This can cause issues with the manageability or supportability of the fabric. Therefore, Dell does not recommend hard zoning with SC Series arrays.

2.2 QLogic settings

Configure the server HBAs according to the recommendations in the *Dell Storage Manager Administrator's Guide* available on the [Dell Support website](#) to improve connection speeds between the database server and SC Series arrays.

Note: Both the BIOS settings and OS driver module for QLogic control the performance of an HBA. Settings specified in the driver module take precedence over the BIOS settings.

For new systems, be sure to review the [Dell EMC Storage Compatibility Matrix](#) for a supported QLogic adapter model, driver, boot code version, and firmware version. Then, update the HBA firmware, boot code, and driver to the applicable versions before doing any configuration. For existing systems, Dell EMC recommends verifying the existing configuration against the compatibility matrix for supportability. Then, verify the functionality of the latest firmware, driver, and QLogic settings in a test environment before promoting the changes to a production environment.

2.2.1 Server FC HBA BIOS settings

Using QLogic Fast!UTIL, configure the HBA BIOS during the power on system test (POST) of the server.

1. During the POST, press **[Ctrl] + [Q]** process to start Fast!UTIL.


```
QLE2562 PCI3.0 Fibre Channel ROM BIOS Version 3.24
Copyright (C) QLogic Corporation 1993-2013. All rights reserved.
www.qlogic.com

Press <CTRL-Q> or <ALT-Q> for Fast!UTIL

<CTRL-Q> Detected, Initialization in progress, Please wait...

Firmware Version 7.03.00
```

Figure 5   Starting Fast!UTIL

2. Select an HBA in Fast!UTIL. Repeat steps 2–5 for all HBAs (or initiators) that are zoned to an SC Series array and are displayed by Fast!UTIL.
3. Reset the adapter to factory defaults. Select **Configuration Settings** > **Restore Default Settings**. Press **[ESC]** multiple times to display the **Fast!UTIL Options** menu.
4. Select **Configuration Settings** > **Adapter Settings** and make the changes shown in Table 3. Press **[ESC]** multiple times to display the main Fast!UTIL menu.
5. Select **Scan Fibre Devices**. Press **[ESC]** to display the main Fast!UTIL menu.
6. Select **Configuration Settings** > **Advanced Adapter Settings** and make the changes shown in Table 3.

Table 3   QLogic HBA BIOS Settings

QLogic BIOS menu	QLogic BIOS attribute	Value
Adapter Settings	Host Adapter BIOS	Enable
	Connection Options	1 (for point-to-point only)
Advanced Adapter Settings	Login retry count	60
	Port down retry count	5, or 60[1]
	Link down timeout	30
	Execution Throttle	256
	LUNs per Target	128
	Enable LIP Reset	Yes
Selectable Boot Settings (Each HBA port has two paths to the boot volume. The WWN for each path should be selected, except when installing and configuring Dell EMC PowerPath™. Then, only server initiator port from one HBA should be enabled for boot.)[2]	Selectable Boot [2]	Enable
	Boot Port Name, Lun (nnnnn,0) (This was configured after the FC fabric zones were created) [2]	WWN for the 1st boot volume path, and Lun should correspond to the Lun chosen during the mapping operation in DSM of the volume to the server. For boot from SAN, Dell EMC Linux best practices say Lun/LUN needs to be 0.
	Boot Port Name, Lun (nnnnn,0) (This was configured after the FC fabric zones were created) [2]	WWN for the 2nd boot volume path, and Lun should correspond to the Lun chosen during the mapping operation in DSM of the volume to the server. For boot from SAN, Dell EMC Linux best practices say Lun needs to be 0.

[1] In multipath environments, port down retry count should be set to 5, and in single-path environments, it should be set to 60. For MySQL, it is recommended to use multipath environments.

[2] If business requirements allow, it is recommended to configure boot from SAN. In such cases, apply the recommended configurations as allowed. If using EMC PowerPath for multipathing, see the *EMC Host Connectivity Guide for Linux* and *PowerPath for Linux Installation and Administration Guide* available on the Dell EMC support website for specific instructions for configuring storage connectivity and QLogic selectable boot settings for boot from SAN.

After configuring the QLogic HBA cards, configure the QLogic driver settings.

Note: Record the WWNs of all HBAs identified and enabled in QLogic and zoned to the SC Series array. WWNs are needed when creating a logical server object in Dell Storage Manager (DSM).

2.2.2   Server FC HBA driver settings: timeouts and queue depth

Configure the link down timeout, and if necessary, the queue depth in Linux after backing up the original QLogic adapter configuration file.

The timeout value determines the time a server waits before the server destroys a connection after losing connectivity. For single-path configurations, the timeout should be set to 60 seconds to provide enough time for the WWN of the failed port to transfer to a port on the other controller (assuming a dual controller SC Series configuration is deployed). In a multipath configuration, the timeout value should be set to 5 seconds because the ports will fail over almost immediately. Either Dell EMC PowerPath or native Linux multipathing (Device-Mapper Multipathing) is recommended.

The default queue depth value of 32 may be adequate, but a value of 64 or 128 may work well too. The optimal queue depth value is dependent on a number of parameters, including creating snapshots (Replays) of the MySQL database. Determining the optimal value is out of scope of this document.

Single-path environments:

```
options qla2xxx qlport_down_retry=60
options qla2xxx ql2xmaxqdepth=<value>
```

Multipath environments:

```
options qla2xxx qlport_down_retry=5
options qla2xxx ql2xmaxqdepth=<value>
```

3 SC Series array

Storage administrators have to make complex decisions daily on storage configuration, usage, and planning. For example, when creating a volume, the question may be asked: Will it be sized appropriately? If all volumes are oversized, there is the added issue of over provisioning the array. Also, traditional arrays do not provide an easy way to match performance metrics of disks with the performance characteristics of the data. Another example is with the MySQL InnoDB system tablespace on a traditional SAN. All tables within the tablespace will share the same performance traits of the underlying physical disk regardless of performance expectations of the applications. This could lead into a situation of expensive RAID 10 disk being used for data that only needs RAID 5.

Dell EMC SC Series storage provides solutions to these complex decisions with a robust set of features that provide an easy-to-manage storage solution:

Dynamic Capacity delivers the highest storage utilization possible by eliminating allocated but unused capacity. It completely separates storage allocation from utilization, enabling users to create any size virtual volume upfront, and only consume actual physical capacity when data is written.

Snapshots (replays) provide continuous, space-efficient data protection. A snapshot taken of a volume creates a point-in-time copy (PITC) of the volume by making all written pages read-only. Any further changes to the volume get written to new pages (active data). When the volume is read, SC Series storage seamlessly presents the read-only pages from the snapshot, and any active data. Consistent snapshots in a MySQL environemnt can be most beneifical and effective for database backups, recoveries, and cloning.

Data Progression (DP) is a Dell Fluid Data™ storage or automated tiered storage feature that automatically migrates data to the optimal storage tier based on a set of predefined or custom policies called storage profiles. Data progression eliminates the need to manually classify and migrate data to different storage tiers while reducing the number and cost of drives and reducing the cooling and power costs.

Fast Track technology enhances automated tiered storage by dynamically placing the most frequently accessed data on the fastest, or outer, tracks of each disk drive.

The following subsections provide Dell EMC recommended best practices, key points, configuration examples, and options for SC Series arrays in a MySQL environment.

3.1   Virtual port mode

Dell EMC strongly recommends using SC Series virtual port mode over Legacy port mode to provide redundancy at the port and storage-controller level. If virtual port mode is used, the WWNs of the virtual ports must be used in the zone rather than the physical ports.
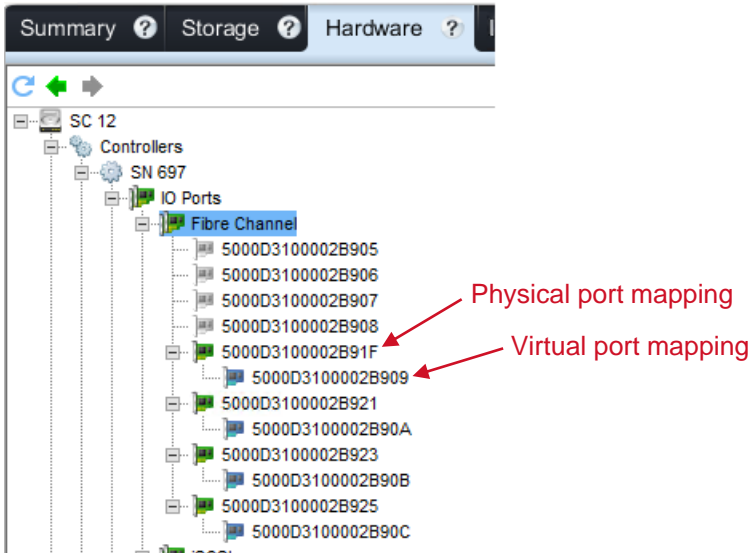
Figure 6   WWNs of physical and virtual ports of an SC8000 controller

In the event of a controller or controller-port failure, the virtual port will fail over to any physical port WWN within the same fault domain. Once the failure is resolved and ports are rebalanced in the SC Series array, the virtual ports return to their preferred physical port. Virtual port mode with FC networks provide benefits over legacy port mode including increased connectivity and bandwidth, and port failover for improved redundancy.

The possible actions taken in virtual port mode when an SC Series controller or port failure occurs are shown in Table 4.

Table 4   Failover mehanics with virtual ports

Scenario	Action taken
Normal operation	All ports pass I/O
A controller fails in a dual-controller array	The virtual ports on the failed controller move to the physical ports on the remaining controller
Port failure	An individual port fails over to another port in the same fault domain

3.2 Fault domains in virtual port mode

Fault domains group front-end ports that belong to the same network and transport media (FC or iSCSI, but not both). There must be a fault domain for each front-end Fibre Channel fabric (see Figure 1). Ports that belongs to the same fault domain can fail over to each other because they have the same connectivity and transport protocol. Dell EMC recommends configuring at least two connections from each controller to each Fibre Channel fabric fault domain.

Name ▽	Controller	Status	Slot	Slot Port	Fault Domain	Purpose
5000D3100002B925	SN 697	✅ Up	4	2	vsan1-2	Front End
5000D3100002B923	SN 697	✅ Up	4	1	vsan1-1	Front End
5000D3100002B921	SN 697	✅ Up	4	4	vsan2-2	Front End
5000D3100002B91F	SN 697	✅ Up	4	3	vsan2-1	Front End

Name ▽	Controller	Status	Slot	Slot Port	Fault Domain	Purpose
5000D3100002B926	SN 698	✅ Up	4	2	vsan1-2	Front End
5000D3100002B924	SN 698	✅ Up	4	1	vsan1-1	Front End
5000D3100002B922	SN 698	✅ Up	4	4	vsan2-2	Front End
5000D3100002B920	SN 698	✅ Up	4	3	vsan2-1	Front End

Figure 7   SC Series fault domains and physical port assignmernt.

To determine the mapping between the physical and virtual ports, see Figure 6.

3.3 Redundancy for SC Series front-end connections

The following types of redundancy are available depending on how the SC Series array is configured.

Storage controller redundancy: The ports on an offline storage controller move to the remaining available storage controller.

Storage controller port redundancy: I/O activity on a failed port moves to another available port in the same fault domain (providing virtual port mode is enabled).

Server path redundancy: Multipathing is configured on the server and multiple paths exist between the SC Series array and server, allowing the server to use multiple paths for I/O. If an I/O path becomes unavailable, the server continues to use the remaining active paths.

Dell EMC strongly recommends that all types of redundancy be provided in all MySQL environments.

3.4   Creating server objects in SC Series arrays

Dell EMC recommends that at least two, dual-port HBAs be installed in the server before it is identified in the SC Series system. Two dual-port HBAs provide redundancy of initiator ports and HBAs. Figure 8 shows a server object with four initiators (two, dual-port HBAs).

Figure 8   HBAs assigned to a server objet in the SC Series system

An important attribute of the server object is the operating system. By setting the operating system to the appropriate type, SC Series storage implements a set of specific OS rules to govern the automated process of mapping volumes to the server object (see section 3.24, Server multipathing, for more information).
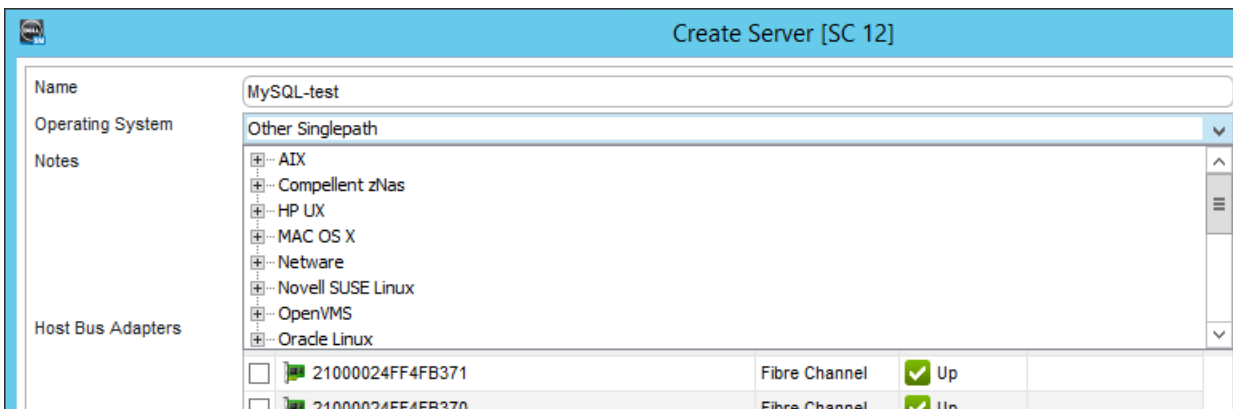

Figure 9   Selecting an OS for a server object in the SC Series system

3.5   Disk pools

In most configurations, SC Series arrays group all disks, regardless of the disk's performance characteristics, into a single folder to create one virtual pool of storage. The virtual pool of storage is referred to as the pagepool and it corresponds to the Assigned disk folder in the SC Series system. There are some cases where multiple disk folders (multiple pagepools) may be needed. In such cases, caution should be used as there are tradeoffs between single and multiple pagepools, the most notable one being performance. If the number of spindles are split between multiple pagepools, there will be a reduction in the maximum performance potential the pagepools can deliver.

Performance potential

>

Pagepool   Pagepool A   Pagepool B
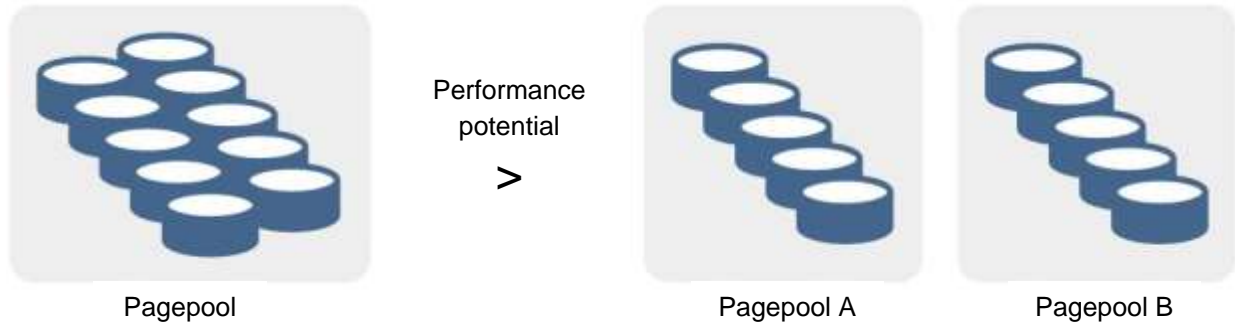
Figure 10   Pagepool performance potential

For this reason, when dealing with high-performance or MySQL database systems in which performance is affected by number of available spindles, Dell EMC strongly recommends that only one pagepool be used. Only in rare circumstances do the benefits of multiple disk folders outweigh the disadvantages. Also, single pagepools are also easier to manage and can accommodate an ever-changing workload. Another reason against using multiple pagepools is that Data Progression does not migrate storage across disk folders, and a second folder would impact the ability of SC Series storage to maximize performance.

When creating the Assigned disk folder, the SC Series system requires a datapage size and redundancy level to be set for the folder. For more information, see Tiered storage and Tier redundancy and RAID.

3.6   Datapage

A datapage is the space that is allocated from the Assigned disk folder to a volume when a request for space is made. Possible datapage sizes are:

512 KB: This datapage size is appropriate for application programs with high performance needs, or environments in which snapshots (replays) are taken frequently under heavy I/O. Selecting this size reduces the amount of space System Manager can present to servers. Flash-optimized storage types use 512 KB by default.

2 MB: The default datapage size. This selection is appropriate for most application program needs.

4 MB: This datapage size is appropriate for configurations that use a large amount of disk space with infrequent snapshots.

Caution: Dell EMC strongly recommends using the default datapage size for MySQL. If there is a need to change the datapage value, consult with Dell Support to assess the performance impact and to ensure system resources remain balanced after changing the datapage size.

When creating a volume, the SC Series system assigns the volume a maximum size as requested. Once the volume is presented to a server, space (datapages) will either be assigned to the volume from the pagepool all at once or when data is written to the volume. For best practices on volume space allocation in MySQL environments, see section 3.22, Preallocating storage.

3.7   Tiered storage

All disk space within the pagepool is allocated into one of three tiers. A tier defines the type of storage media used to save the data. When only two types of disks are used in an SC Series array, the array automatically creates two tiers of storage. The fastest disks are placed in tier 1, and higher capacity, cost-efficient disks with lower performance metrics are assigned to tier 3. When three types of disks are used, the array automatically creates three tiers of storage.
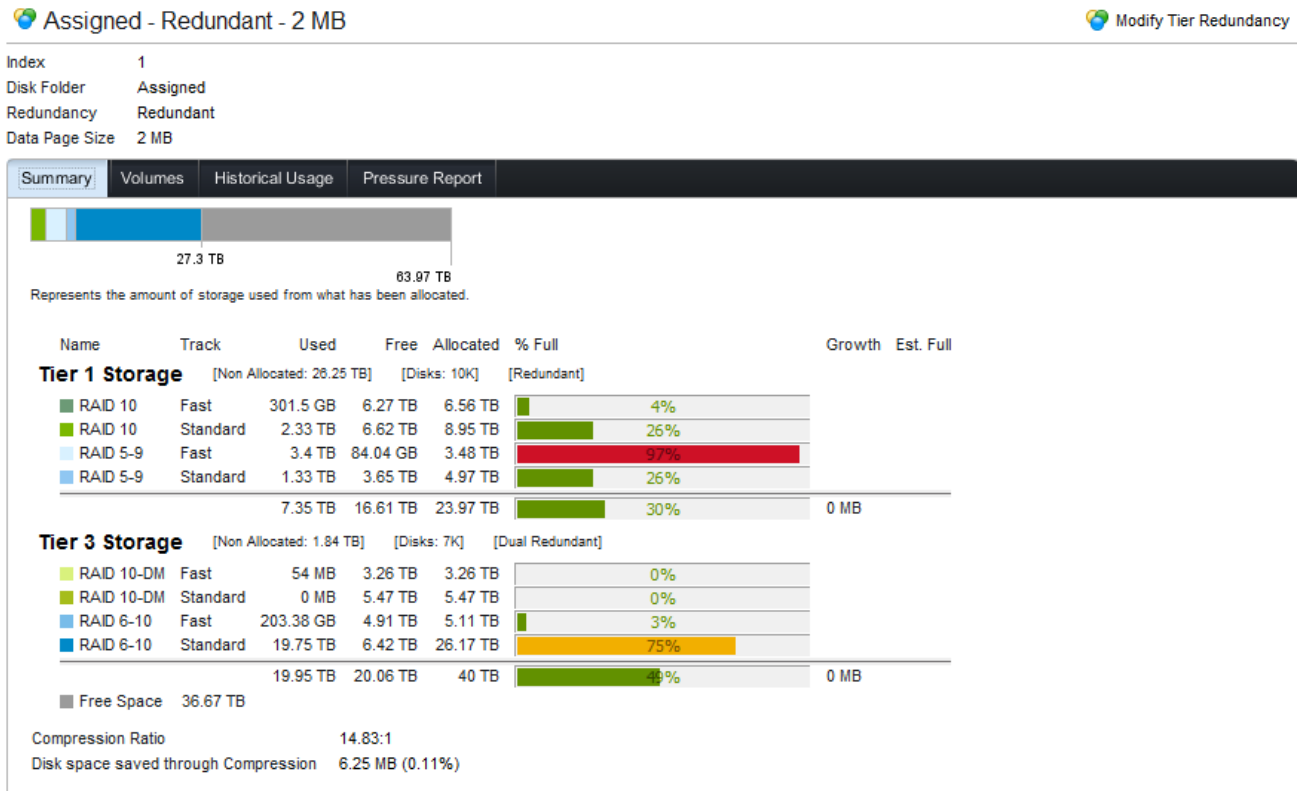
Figure 11   Storage types in SC Series arrays

Frequently accessed data remains on tier 1 (T1), and data that has not be accessed for the last 12 progression cycles is gradually migrated to tier 2 (T2) (should it exist) and then to tier 3 (T3).

3.8 Tier redundancy and RAID

Data within tiers is protected by redundancy through the implementation of RAID technology. RAID requirements for each disk tier are based on the type, size, and number of disks in the tier, and will result in either single or dual redundancy of the data on a volume. In rare cases, redundancy for a tier can be disabled by using RAID 0, but caution should be exercised if using RAID 0 in MySQL deployments (see Table 5 for details).

Table 5   Tier redundancy and RAID types

Tier redundancy	Description
Non-redundant	SC Series arrays will use RAID 0 in all classes, in all tiers. Data is striped but provides no redundancy. If one disk fails, all data is lost. Dell EMC does not recommend using non-redundant (RAID 0) storage unless data has been backed up elsewhere, and then only in specific cases after a thorough evaluation of business requirements. Contact Dell Support for advice.
Single-redundant	Protects against the loss of any one drive. Single-redundant tiers can contain any of the following types of RAID storage: • RAID 10 • RAID 5-5 (requires 6-disk minimum) • RAID 5-9 (requires 10-disk minimum)
Dual-redundant	Protects against the loss of any two drives. Disks larger than 900 GB should use dual redundancy and in some cases it is mandated. Dual-redundant tiers can contain any of the following types of RAID storage. Should the tier have fewer than the required 7 disk minimum for RAID 6, the default tier redundancy is set to single redundancy RAID 5-5 for any disk drive size. • RAID 10 dual mirror (DM) • RAID 6-6 (requires 7-disk minimum) • RAID 6-10 (requires 11-disk minimum)

By default, SC Series storage implements dual redundancy. With RAID 6-10, a minimum of 11 disks are required. With RAID 6-6 a minimum of seven disks are required. With respect to RAID 6, if a tier does not have at least seven disks, the default tier redundancy is set to single redundancy RAID 5-5 for any disk drive size.
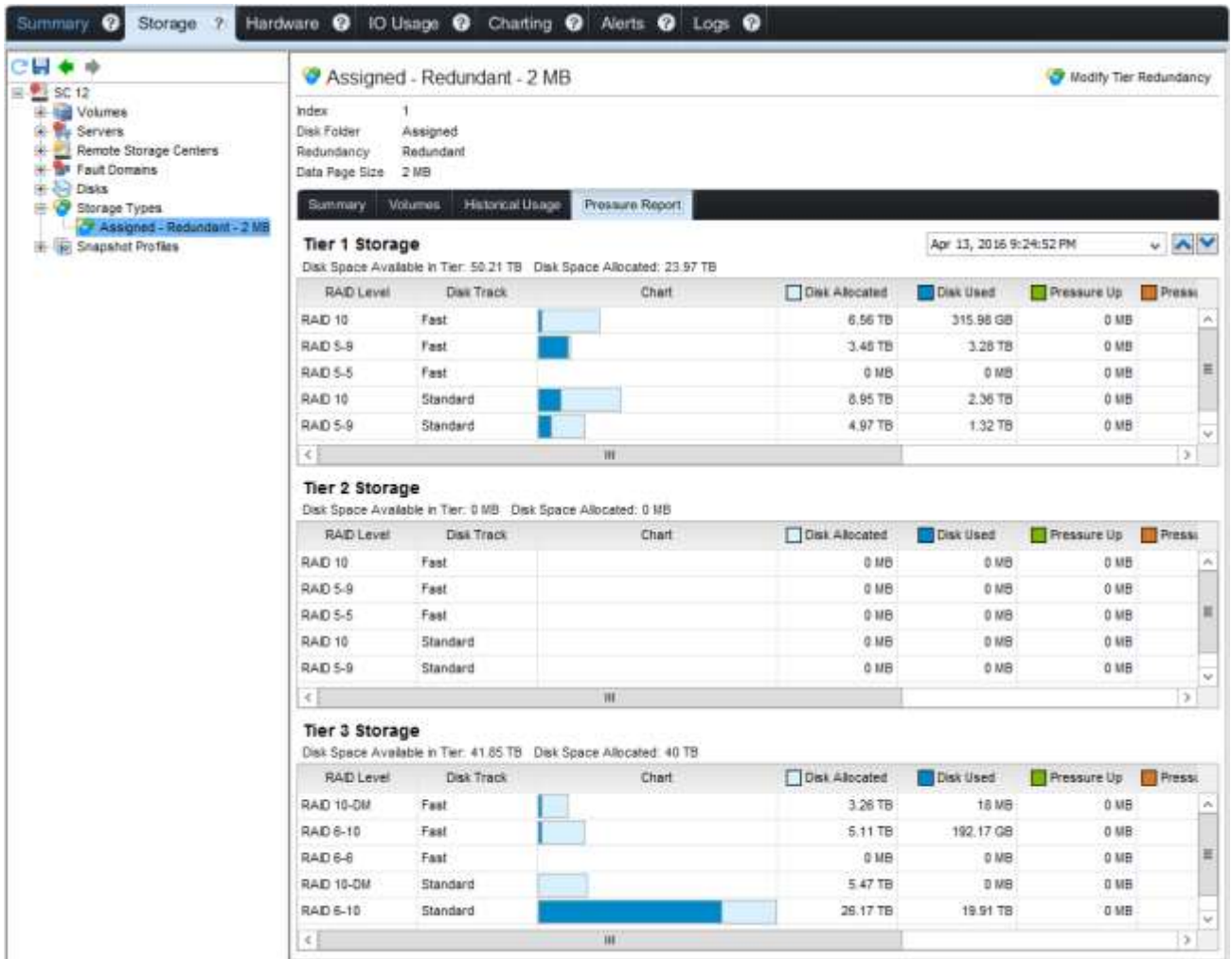

Figure 12   Tier redundancy and RAID types

3.9 Tier redundancy and media type

Media type used within a tier can influence the type of redundancy enforced on the tier.

Table 6   Media type and redundancy requirements

Media type	Redundancy requirements
HDD	By default, disks under 966 GB are set to single redundancy. Dual redundancy is recommended for 966 GB to 1.93 TB disks. Dual redundancy is required when adding 1.93 TB or larger disks to a new disk pool.
SSD	By default, disks under 1.7 TB are set to single redundancy. Dual redundancy is recommended for 1.7 TB to 2.78 TB disks. In an SC Series system with new or existing disk pools, adding 2.79 TB or larger disks requires dual redundancy (single redundancy option is disabled).

Storage profiles define tier redundancy, or the RAID level used to protect data on a volume. See section 3.16 for information on storage profiles and MySQL usage.

In most MySQL environments, using the default tier redundancy provides appropriate levels of data protection, good I/O performance, and storage conservation for all types of database applications. Therefore, Dell EMC strongly recommends using the default tier redundancy and evaluating its suitability before attempting to change it. If changing the default tier redundancy, do so only after evaluating MySQL and application requirements during a RAID rebalance, should one be required. Consult Dell Support in assessing a change.

With high-performance MySQL applications, it is also recommended to evaluate gains in storage efficiencies with dual redundancy against increased latencies caused by the extra RAID penalty of dual redundancy. If the increased latencies are a concern, evaluate the suitability of using single redundancy. Keep in mind that single redundancy will provide greater storage efficiencies, less latency, but will provide less data protection in the event of multiple drive failures.

For more information on redundancy requirements for each disk tier, see the *Storage Center System Manager Administrator's Guide* available on the Knowledge Center at the SC Series Customer Portal (login required).

3.10   RAID stripe width

The default stripe width is 10 wide for RAID 6 (RAID 6-10) storage profiles, and 9 wide for RAID 5 (RAID 5-9). Modifying stripe width updates the corresponding RAID 5 or 6 selections for all predefined storage profiles. User-created storage profiles are not affected. Table 7 shows the stripe widths for all RAID levels in SC Series arrays.

Note: A RAID rebalance should not be performed unless sufficient free disk space is available within the assigned disk folder, and only should be done when most appropriate for application requirements.

Table 7   RAID stripe width in SC Series arrays

RAID level	Stripe width description
RAID 0	Stripes across all drives in the tier with no redundancy
RAID 10	Stripes data along with one copy across all drives in the tier
RAID 10-DM	Stripes data along with two copies across all drives in the tier
RAID 5-5	Distributes parity across five drives (4 data segments, 1 parity segment for each stripe); tier requires at least 6 drives (5 for RAID, and one for spare)
RAID 5-9	Distributes parity across nine drives (8 data segments, 1 parity segment for each stripe); tier requires at least 10 drives (9 for RAID, and one for spare)
RAID 6-6	Distributes parity across six drives (4 data segments, 2 parity segments for each stripe); tier requires at least 7 drives (6 for RAID, and one for spare)
RAID 6-10	Distributes parity across ten drives (8 data segments, 2 parity segments for each stripe); tier requires at least 11 drives (10 for RAID, and one for spare)
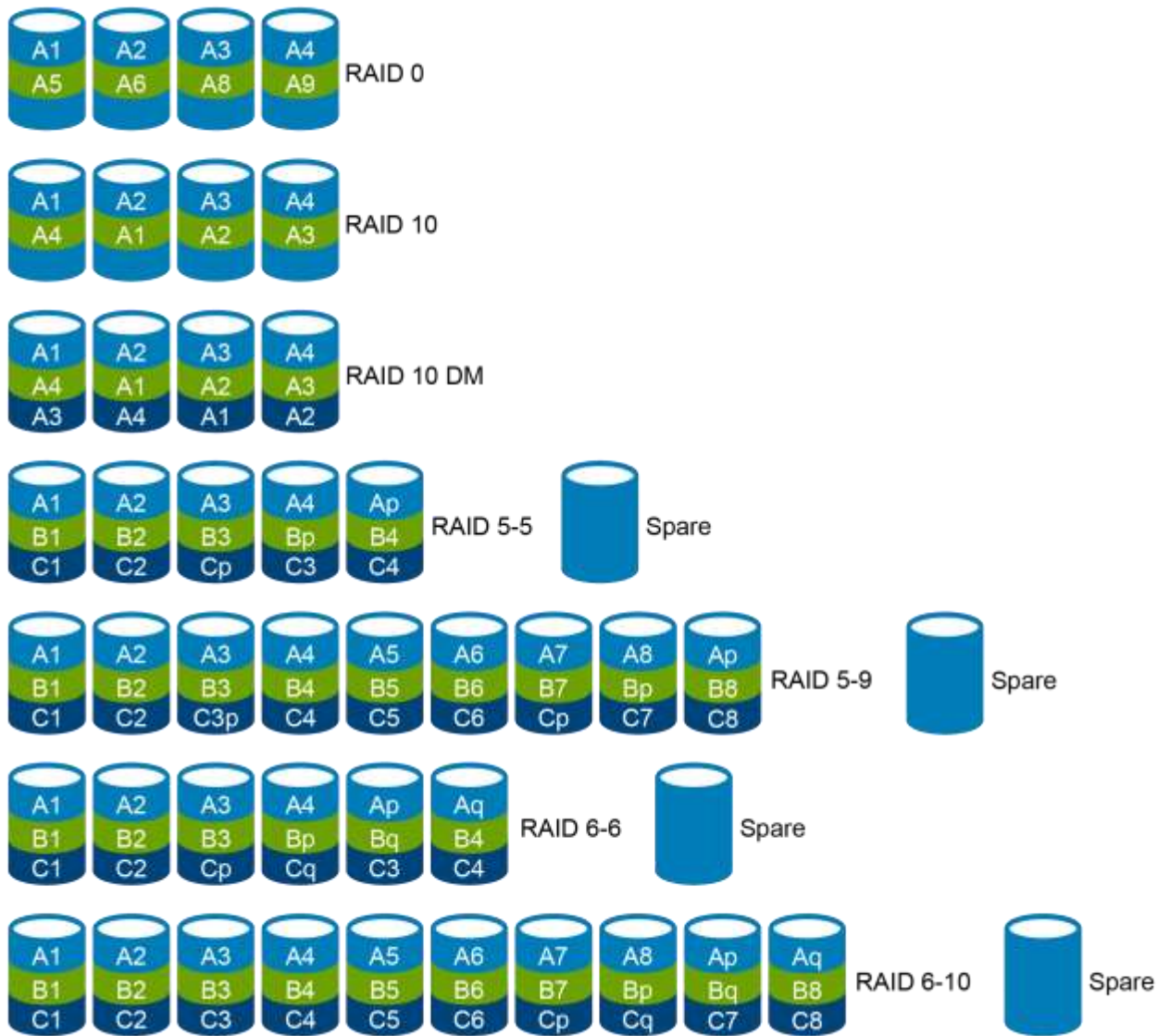
Figure 13   RAID stripes in SC Series arrays

SC Series arrays store the most active data on RAID 10, and the least active data on RAID 5 or RAID 6 (see Tier redundancy and RAID). Distributing data across more drives is marginally less efficient, but decreases vulnerability. Conversely, distributing data across fewer drives is more efficient, but marginally increases vulnerability.

To view RAID stripe widths and efficiencies, in DSM right-click the array, select **Edit Settings**, and click **Storage**.
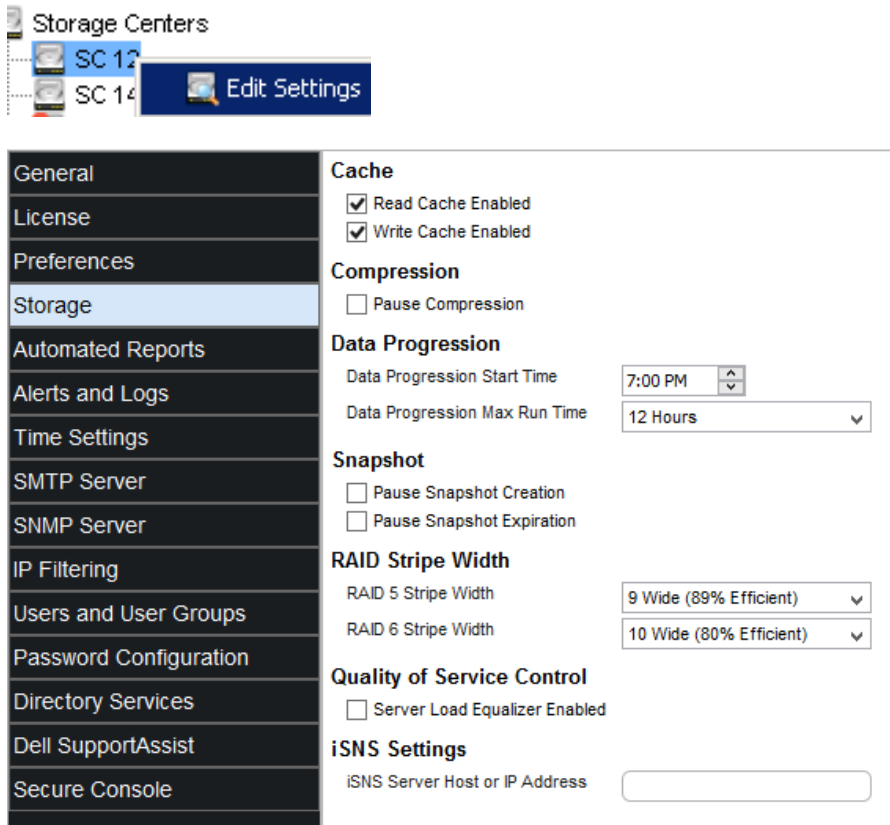
The **RAID Stripe Width** drop-down fields show the available stripe widths for RAID 5 and RAID 6.
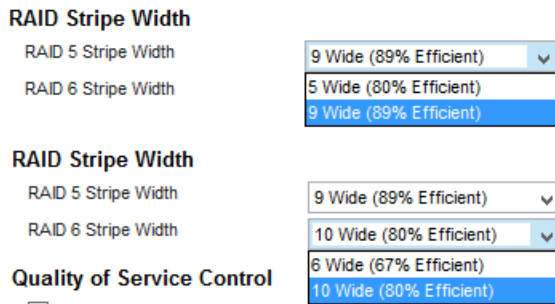
Figure 14   RAID efficiencies of stripe widths

In most MySQL environments, using the default and recommended RAID stripe width provides good I/O performance and storage conservation for all types of database volumes and applications. Dell EMC strongly recommends using the default stripe width and evaluate its suitability before attempting to change it (consult Dell Support in assessing the appropriateness of a change). A RAID rebalance will be required after it has been modified if the tier contains any disks. For more information on stipe size, see the *Storage Center System Manager Administrator's Guide* available on the Knowledge Center at the SC Series Customer Portal.

Note: A RAID rebalance should not be performed unless sufficient free disk space is available within the assigned disk folder, and only should be done when most appropriate for MySQL application requirements. If there is a need to change RAID stripe, consult with Dell Support to assess the performance impact and to ensure system resources remain balanced during the RAID rebalance.

3.11 RAID penalty

Depending on the RAID level implemented, data and parity information may be striped between multiple disks. Before any write operation is considered complete, the parity must be calculated for the data and written to disk. The time waiting for the parity information to be written to disk, or the number of extra I/Os required for parity, is referred to the RAID (or write) penalty. The penalty only comes into play when a write is required. With RAID 0, there is no penalty because there is no parity. When there is no penalty, the write or RAID penalty is expressed as a 1 (the one write required to write the original data to disk).

Table 8 lists the RAID penalty and description for each SC Series RAID level. Calculating the RAID penalty is out of scope of this document.

Table 8   RAID penalty

RAID level	RAID penalty	I/O description
RAID 0	1	1 write
RAID 10	2	2 writes (one for data and one for copy of data)
RAID 10-DM	3	3 writes (one for data and two for copy of data)
RAID 5-5	4	2 reads (one for data and one for parity),
RAID 5-9		2 writes (one for data and one for parity)
RAID 6-6	6	3 reads (one for data and two for parity),
RAID 10-10		3 writes (one for data and two for parity)

Despite the fact that RAID 0 has the lowest RAID penalty, Dell EMC does not recommend using it (see Tier redundancy and RAID). SC Series storage uses RAID 10 by default for every write in all storage profiles, and for most cases, RAID 10 is recommended for all writes.

3.12 Sizing and validating SC Series array design

When sizing an SC Series array with two tiers, if the access pattern taken by MySQL is even across all databases, it may be appropriate to place all MySQL databases on T1 to ensure good performance. In cases like this, Dell EMC recommends that T1 be sized for 100 percent of all the MySQL databases and total expected IOPS. In an environment where the SC Series array is not dedicated to MySQL, T1 sizing needs to also include IOPS for other application requirements. T1 sizing also needs to include the amount of storage required by snapshots retained in T1 when using default storage profiles: Recommended (All Tiers), High Priority (Tier 1), Write Intensive (Tier 1) (see Table 9), or custom storage profiles defined to use T1 for snapshots. If the MySQL access pattern is not even across all database(s), it might be appropriate to size T1 for 75 percent of the IOPS and at least 25 percent of the capacity, and tier 3 for 75 percent of the capacity and at least 25 percent of the IOPS.

In either case, there are several recommendations and points to consider when sizing:

- Dell EMC recommends evaluating tier sizing (including the RAID write penalty and snapshots) before actually implementing the tier in any environment. When sizing an array, assume all I/O will be random. This will yield best results.
- Before releasing any storage system to production, Dell EMC recommends using the Dell Performance Analysis Collection Kit (DPACK) on a simulated production system during at least a 24-

hour period that includes the peak workload. The simulation will help define I/O requirements. It might be possible to use IOMeter to simulate the production system. After go live, repeat the process again.

- Have a good understanding of the MySQL workloads (OLTP, OLAP, or hybrid). The workload will have an impact on how an SC Series array is sized and configured. For OLTP systems, generally SSDs or 15K drives are used to deliver high IOPS with low latencies. For OLAP systems, generally the connectivity of the environment is of concern when sizing the array, as well as choosing between either 15K or 10K drives.

- For most I/O patterns, SSDs provide considerably better performance than spinning media. Understand the tradeoffs between multi- and single-level cell SSDs (MLC, SLC). SLCs are better suited for write-intensive workloads, and MLCs are better suited for read-intensive workloads.

- Have a good understanding of required latency, IOPS and throughput of the entire server (and not just MySQL). The IOPS will help determine the number of disks required in the array, and throughput will help define the paths between the SC Series array and server.

- Accessible datapages on a volume are not eligible to move to a lower tier until they are at least 12 progression cycles old.

- In environments where tables undergo frequent mass updates or reloads, or frequent index rebuilds, datapages may never reach an age of 12 progression cycles. Therefore, these pages will always reside in T1. In cases like this, T1 needs to be sized for 100 percent of the capacity and IOPS of the MySQL database(s).

Dell EMC strongly recommends factoring in the RAID penalty when sizing a SC Series array. If this is not considered, the SC Series array will be undersized. To include IOPS for write penalty, use the following formula:

Total IOPS = Number of IOPS + (Number of IOPS * Write Percentage * (RAID Penalty - 1))

Number of IOPS = Total number of IOPS generated by the server

Write Percentage = Percent of the I/O attributed to writes

RAID Penalty = Number of I/Os required to complete a write

For example, consider a server that is performing a workload of 20,000 IOPS with a mix of 80 percent reads and 20 percent writes to RAID 10. With the previous equation, a total of 24,000 IOPS would be required for T1.

3.13   Data Progression

Data Progression is a separately licensed application feature of the SC Series array that provides a constant balance of performance and cost, leveraging different storage tiers and drive types, as well as among multiple RAID levels within the same tier. Data Progression provides the maximum use of lower-cost drives for stored data, while maintaining high-performance drives for frequently-accessed data. The behavior of Data Progression is determined by the storage profile applied to each volume. To see if Data Progression is licensed, right-click an SC Series array, select **Edit Settings** from the context menu, and select **License**.

Figure 15   Verifying Data Progression license

By default, Data Progression runs every 24 hours at 7 p.m. system time. Should there be a tremendous amount of MySQL activity and demand for SC Series resources during Data Progression cycles, the scheduled start time and a maximum elapsed time of Data Progression can be changed. It is recommended to consult with Dell Support to assess the appropriateness of a change. To update the scheduled time of Data Progression, right-click an SC Series array, select **Edit Settings** from the context menu, select **Storage**, and set the start and maximum elapsed time for Data Progression.

Figure 16   Setting the start time and maximum elapsed run time for Data Progression cycles

SC Series arrays also use Data Progression to move snapshots. When a snapshot is created, the data is frozen and moved immediately to the tier specified by the storage profile to hold snapshots.

SC Series arrays also use Data Progression to move snapshots. When a snapshot is created, the data is frozen and moved immediately to the tier specified by the storage profile to hold snapshots.

Table 9   Snapshot storage classes used by Data Progression

Storage profile	Tier 1	Tier 2	Tier 3
Recommended (All Tiers)	RAID 5-9, RAID 6-10	RAID 5-9, RAID 6-10	RAID 5-9, RAID 6-10
High Priority (Tier 1)	RAID 5-9, RAID 6-10		
Flash Optimized with Progression (Tier 1 to All Tiers)		RAID 5-9, RAID 6-10	RAID 5-9, RAID 6-10
Write Intensive (Tier 1)	RAID 10, RAID 10-DM		
Flash Only with Progression (Tier 1 to Tier 2)		RAID 5-9, RAID 6-10	
Low Priority with Progression (Tier 3 to Tier 2)		RAID 5-9, RAID 6-10	RAID 5-9, RAID 6-10
Medium Priority (Tier 2)		RAID 5-9, RAID 6-10	
Low Priority (Tier 3)			RAID 5-9, RAID 6-10

Snapshots can occur in multiple ways:

- As a scheduled event according to the snapshot profile
- By manual selection through System Manager
- On demand by the SC Series array to move data off T1 in a flash-optimized storage type

3.14 Data Progression pressure reports

A tier can become full through normal usage, by data movement from Data Progression cycles, or from frequent database snapshots with long retention periods. When a tier becomes full, the SC Series array writes data to the next lower tier which can cause performance issues because of the RAID penalty. Therefore, Dell EMC recommends using Data Progression pressure reports to monitor disk usage. Data Progression pressure reports display how space is allocated and consumed across different RAID types and storage tiers for the previous 30 days. This trending information can assist in fine-tuning snapshot profiles and retention periods for MySQL databases and cloning operations

To view the Data Progression pressure report, perform the following:

1. In DSM, select the **Storage** view.
2. In the **Storage** pane, select an SC Series array.
3. Click the **Storage** tab.
4. In the navigation pane, expand **Storage Type.**
5. Select the individual storage type to examine.
6. Select the tab, **Pressure Report**.

Tier 1 Storage
Disk Space Available in Tier: 50.21 TB   Disk Space Allocated: 23.97 TB

RAID Level	Disk Track	Chart	Disk Allocated	Disk Used	Pressure Up	Press
RAID 10	Fast		6.56 TB	315.98 GB	0 MB	
RAID 5-9	Fast		3.48 TB	3.28 TB	0 MB	
RAID 5-5	Fast		0 MB	0 MB	0 MB	
RAID 10	Standard		8.95 TB	2.36 TB	0 MB	
RAID 5-9	Standard		4.97 TB	1.32 TB	0 MB	

Tier 2 Storage
Disk Space Available in Tier: 0 MB   Disk Space Allocated: 0 MB

RAID Level	Disk Track	Chart	Disk Allocated	Disk Used	Pressure Up	Press
RAID 10	Fast		0 MB	0 MB	0 MB	
RAID 5-9	Fast		0 MB	0 MB	0 MB	
RAID 5-5	Fast		0 MB	0 MB	0 MB	
RAID 10	Standard		0 MB	0 MB	0 MB	
RAID 5-9	Standard		0 MB	0 MB	0 MB	

Tier 3 Storage
Disk Space Available in Tier: 41.85 TB   Disk Space Allocated: 40 TB

RAID Level	Disk Track	Chart	Disk Allocated	Disk Used	Pressure Up	Press
RAID 10-DM	Fast		3.26 TB	18 MB	0 MB	
RAID 6-10	Fast		5.11 TB	192.17 GB	0 MB	
RAID 6-6	Fast		0 MB	0 MB	0 MB	
RAID 10-DM	Standard		5.47 TB	0 MB	0 MB	
RAID 6-10	Standard		26.17 TB	19.91 TB	0 MB	

Figure 17   Data Progression pressure report

A drop-down field (shown in Figure 18) can be used to report on a specific date. For more information on Data Progression pressure reports, see the *Storage Center System Manager Administrator's Guide* available on the Knowledge Center at the SC Series Customer Portal.

Figure 18   Specific date period for pressure report

3.15   Volume distribution reports

Dell EMC also recommends using the metrics reported using the subtabs for Volumes, Volume Growth, and Storage Chart. These are located in the main window after selecting **Volumes** from the navigation tree (see Figure 19). The metrics help to determine:

- Growth trends (logical and physical space consumed and allocated space for a volume)
- Volumes which have the largest overhead of snapshot data
- Any necessary snapshot cleanup or snapshot profile adjustments justified
- The amount of logical and physical space consumed and allocated space for a volume
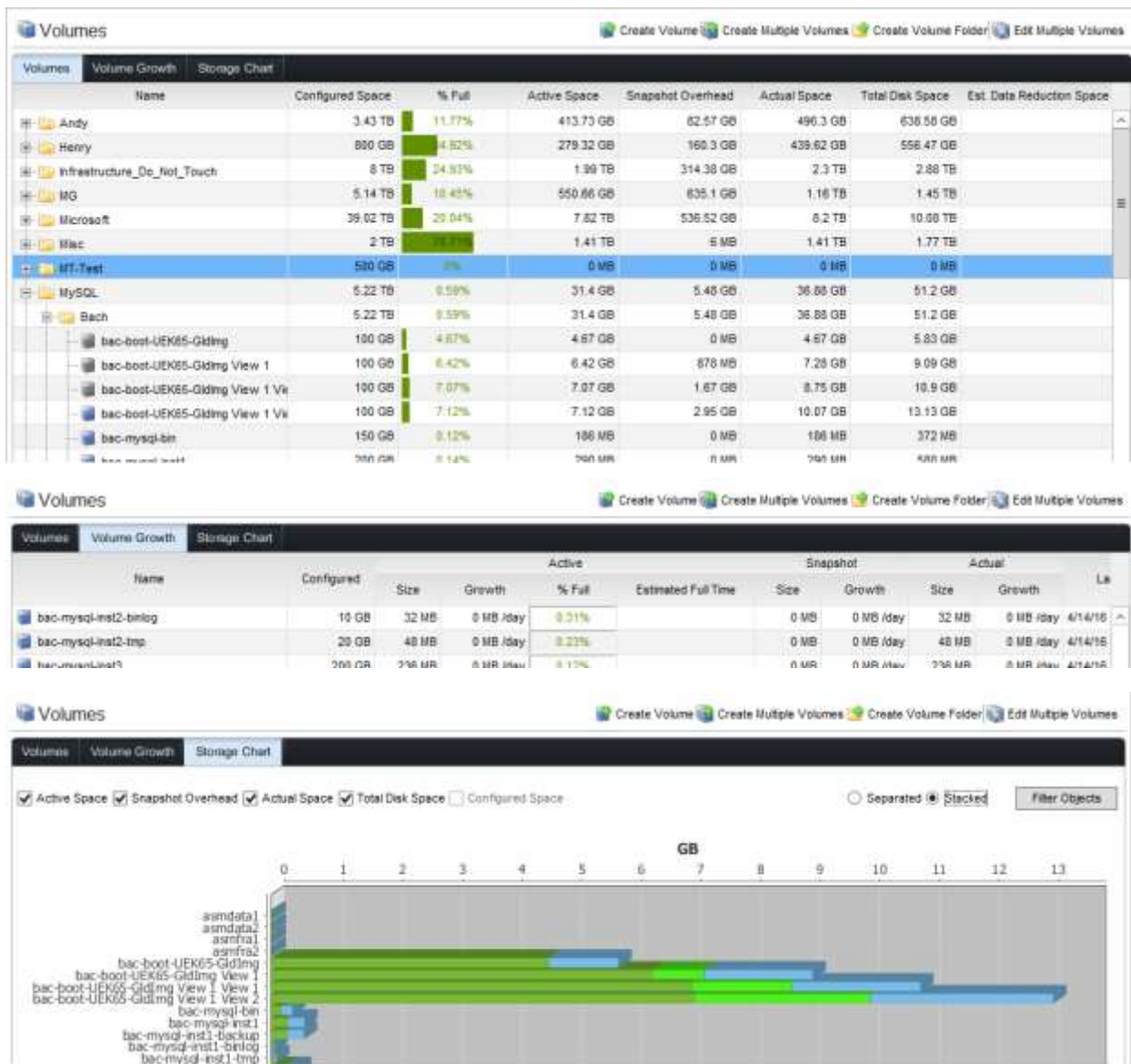
Figure 19   Volume distribution reports

3.16   Storage profiles

Storage profiles are used to accomplish the following:

- Define which tier of disk is used to accept initial writes, or new pages, to a volume
- Determine how to move data between and across tiers
- Set the RAID level used to protect data on a volume
- Apply profiles to each volume, by default

In most MySQL environments, using the recommended default storage profile provides appropriate levels of data protection, good I/O performance for all types of database volumes by ensuring all writes occur to T1 RAID 10, and movement of less-active data to lower storage tiers. Therefore in MySQL environments, Dell EMC recommends using the default storage profile and evaluating its appropriateness before attempting to change or use a different storage profile. If there is a need to pin the entire database (active and less-active

data) to T1, consider the impacts on T1 capacity and Data Progression caused by complete data rebuilds from nightly extract, transform, load (ETL) operations, index rebuilds, and associated snapshots.

Depending on storage media, storage profiles will vary. The default standard storage profiles are described in Table 10.

Table 10   Default standard profiles in SC Series arrays (spinning media or hybrid arrays)

Name	Initial write tier	Tier and RAID levels	Progression
Recommended (All Tiers)	1	Writeable: • T1 RAID 10, RAID 10-DM • T2 RAID 10, RAID 10-DM Snapshots: • T1 RAID 5-9, RAID 6-10 • T2 RAID 5-9, RAID 6-10 • T3 RAID 5-9, RAID 6-10	To all tiers*
High Priority (Tier 1)	1	Writeable: • T1 RAID 10, RAID 10-DM Snapshots: • T1 RAID 5-9, RAID 6-10	No
Medium Priority (Tier 2)	2	Writeable: • T2 RAID 10, RAID 10-DM Snapshots: • T2 RAID 5-9, RAID 6-10	No
Low Priority (Tier 3)	3	Writeable: • T3 RAID 10, RAID 10-DM Snapshots: • T3 RAID 5-9, RAID 6-10	No

* Available only when Data Progression is licensed.

DELLEMC

Recommended (All Tiers): With this storage profile, all new data is written to T1 RAID level 10 storage. Data Progression moves less-active data to T1 RAID 5/RAID 6 or a slower tier based on how frequently the data is accessed. In this way, the most active blocks of data remain on high-performance SSDs or Fibre Channel drives, while less active blocks automatically move to lower-cost, high-capacity SAS drives.

High Priority (Tier 1): This storage profile provides the highest performance by storing written data in RAID 10 or RAID 10 DM on T1, and snapshot data in RAID 5-9/RAID 6-10 on T1. The SC Series array does not migrate data to lower storage tiers unless tier 1 storage becomes full. If Data Progression is not licensed, this is the default storage profile. Without Data Progression, volumes must be configured to use a specific tier of storage, because data will not migrate between tiers.

Medium Priority (Tier 2): This storage profile provides a balance between performance and cost efficiency by storing written data in RAID 10 or RAID 10 DM on T2, and snapshot data in RAID 5-9/RAID 6-10 on T2. The SC Series array does not migrate data to other storage tiers unless T2 becomes full.

Low Priority (Tier 3): This storage profile provides the most cost-efficient storage by storing written data in RAID 10 or RAID 10 DM on T3, and snapshot data in RAID 5-9/RAID 6-10 on T3. The SC Series array does not migrate data to higher tiers of storage unless T3 becomes full.

Table 11   Default standard profiles in SC Series all-flash arrays

Name	Initial write tier	Tier and RAID levels	Progression
Flash Optimized with Progression (Tier 1 to All Tiers)	1	Writeable: • T1 RAID 10, RAID 10-DM Snapshots: • T2 RAID 5-9, RAID 6-10 • T3 RAID 5-9, RAID 6-10	To all tiers
Write Intensive (Tier 1)	1	Writeable: • T1 RAID 10, RAID 10-DM Snapshots: • T1 RAID 10, RAID 10-DM	No
Flash Only with Progression (Tier 1 to Tier 2)	1	Writeable: • T1 RAID 10, RAID 10-DM Snapshots: • T2 RAID 5-9, RAID 6-10	To tier 2 only

DELLEMC

Name	Initial write tier	Tier and RAID levels	Progression
Low Priority with Progression (Tier 3 to Tier 2)	3	Writeable: • T3 RAID 10, RAID 10-DM Snapshots: • T2 RAID 5-9, RAID 6-10 • T3 RAID 5-9, RAID 6-10	To tier 2 only
Low Priority (Tier 3)	3	Writeable: • T3 RAID 10, RAID 10-DM Snapshots: • T3 RAID 5-9, RAID 6-10	No

Flash Optimized with Progression (Tier 1 to All Tiers): This storage profile provides the most efficient storage for an enclosure containing both read-intensive and write-intensive SSDs. All new data is written to write-intensive T1 drives, snapshot data is moved to T2 RAID 5-9/RAID 6-10, and less-active data progresses to T3 RAID 5-9/RAID 6-10. When T1 reaches 95 percent capacity, the SC Series array creates a space management snapshot and moves it immediately to T2 to free up space on T1. The space management snapshot is moved immediately and does not wait for a scheduled Data Progression. Space management snapshots are marked as created on demand and cannot be modified manually or used to create view volumes. Space-management snapshots coalesce into the next scheduled or manual snapshot. The SC Series array creates only one on-demand snapshot per volume at a time.

Write Intensive (Tier 1): This storage profile directs all initial writes to write-intensive SSDs on T1 RAID 10. This data does not progress to any other tier, but remains on T1. This profile is useful for storing MySQL transaction logs and temporary database files.

Flash Only with Progression (Tier 1 to Tier 2): This storage profile performs initial writes to T1 RAID 10/RAID 10 DM on high-performance drives. Less-active data progresses to T2 RAID 5-9/RAID 6-10, but remains on SSDs. This profile is useful for storing volumes with data that requires optimal read performance, such as MySQL database golden images or clones for rapid database deployments.

Low Priority with Progression (Tier 3 to Tier 2): This storage profile directs initial writes to T3 RAID 10/RAID 10 DM on less-expensive drives, and then allows frequently accessed data to progress from T3 RAID 5-9/RAID 6-10 to T2 RAID 5-9/RAID 6-10. This profile is useful for migrating large amounts of data to SC Series for MySQL usage, or infrequent large MySQL data loads without overloading T1 SSDs.

Because SSDs are automatically assigned to T1, profiles that include T1 allow volumes to use SSD storage. If there are volumes that contain data that is not accessed frequently, and do not require the performance of T1 SSDs, use profiles Medium Priority (Tier 2) or Low Priority (Tier 3), or create and apply a new profile that does not include high-performance disks.

In hybrid SC Series arrays (SSDs and HDDs), if there are not enough SSDs to retain the entire database, use custom storage profiles to separate the data between the different media, then monitor the performance to determine a best fit for each media type. When using SLCs with automated tiering, the tier needs to be sized for 100 percent IOPS and percentage of the capacity of the volumes using the SSD tier. If it is not and the tier becomes full, performance can degrade considerably. In cases like this, redistribute the data to more appropriate storage tiers.

Using flash-optimized SC Series arrays provides the benefit of delivering high capacity and performance. SLCs in this configuration should be sized to support the expected IOPS, and MLCs should be sized to support the expected capacity.

MySQL binary transaction logs can be used to help determine if certain tables might be candidates for a dedicated volume assigned to storage profile: Medium Priority (Tier 2), Low Priority (Tier 3), or Low Priority with Progression (Tier 3 to Tier 2). See the section, MySQL binary transaction logs, for more information.

Note: Custom storage profiles are beyond the scope of this document. Dell EMC recommends using the storage profiles provided by the SC Series array for best performance with Data Progression. If a custom profile is needed, engage Dell Support to discuss the performance impact and to ensure that system resources remain balanced with a custom profile.

Note: For best performance with Data Progression, Dell EMC recommends using the storage profiles provided by the SC Series array. If a custom profile is needed, engage Dell Support to discuss the performance impact and to ensure that system resources remain balanced with a custom profile. Custom storage profiles are beyond the scope of this document.
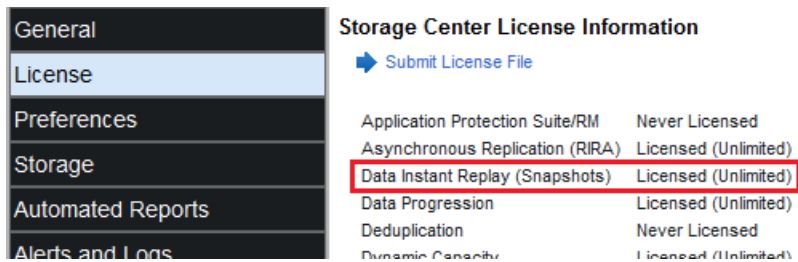
Note: Once the ability to view and create Storage Profiles has been enabled, it cannot be disabled.

3.17 Snapshots (replays)

The snapshot (Data Instant Replay) feature is a licensed technology available on SC Series arrays that provides continuous space-efficient data protection. Snapshots create space-efficient, write-consistent, point-in-time copies (PITC) of one or more volumes based on a collection of rules that exist in a snapshot profile. Snapshots (replays) can be used for immediate recovery from data loss caused by hardware failures or logical errors. SC Series snapshot technology is different from other traditional PITCs because when a snapshot is taken data is not copied, it is only frozen. Should a write request occur to a frozen block, a new RAID 10 or RAID 10 DM block is allocated to the volume and the write will occur there.

Dell EMC strongly recommends that this feature be licensed in a MySQL environment. It dramatically simplifies both the effort and required resources to implement backup and recovery strategies. It also provides an easy methodology for cloning MySQL databases or tables regardless of the MySQL engine or table size.

To see if the **Data Instant Replay (Snapshots)** feature is licensed, right-click an SC Series array, select **Edit Settings** from the context menu, then select **License**.

Figure 20   Verifying Data Instant Replay (Snapshots) license

Some scenarios for using snapshots with MySQL databases are:

- Augment conventional backup strategy and provide immediate recovery from data loss caused by hardware failures or logical errors to the last known state of any unexpired snapshot
- Provide ability to immediately create a copy of a database on a server for development, testing, reporting, and troubleshooting
- Move data from T1 to T2 or T3 using on-demand snapshots, should MySQL required more space on T1 flash-optimized storage when the array is using both spinning and fixed drives
- Provide lower RTO and RPO than conventional backup/recovery strategies
- Provide the ability to replicate a database to a DR location

3.18   Snapshot profiles

A snapshot profile is a policy-based collection of rules describing the type of schedule (once, daily, weekly, monthly), a date/time and the interval to create the snapshot, the volumes, an expiration time for the snapshot, and if the snapshot should be write-consistent across all volumes in the schedule. Write-consistency and data integrity across all volumes in the schedule is governed by defining the schedule as a **Consistent** snapshot profile (see Figure 21)
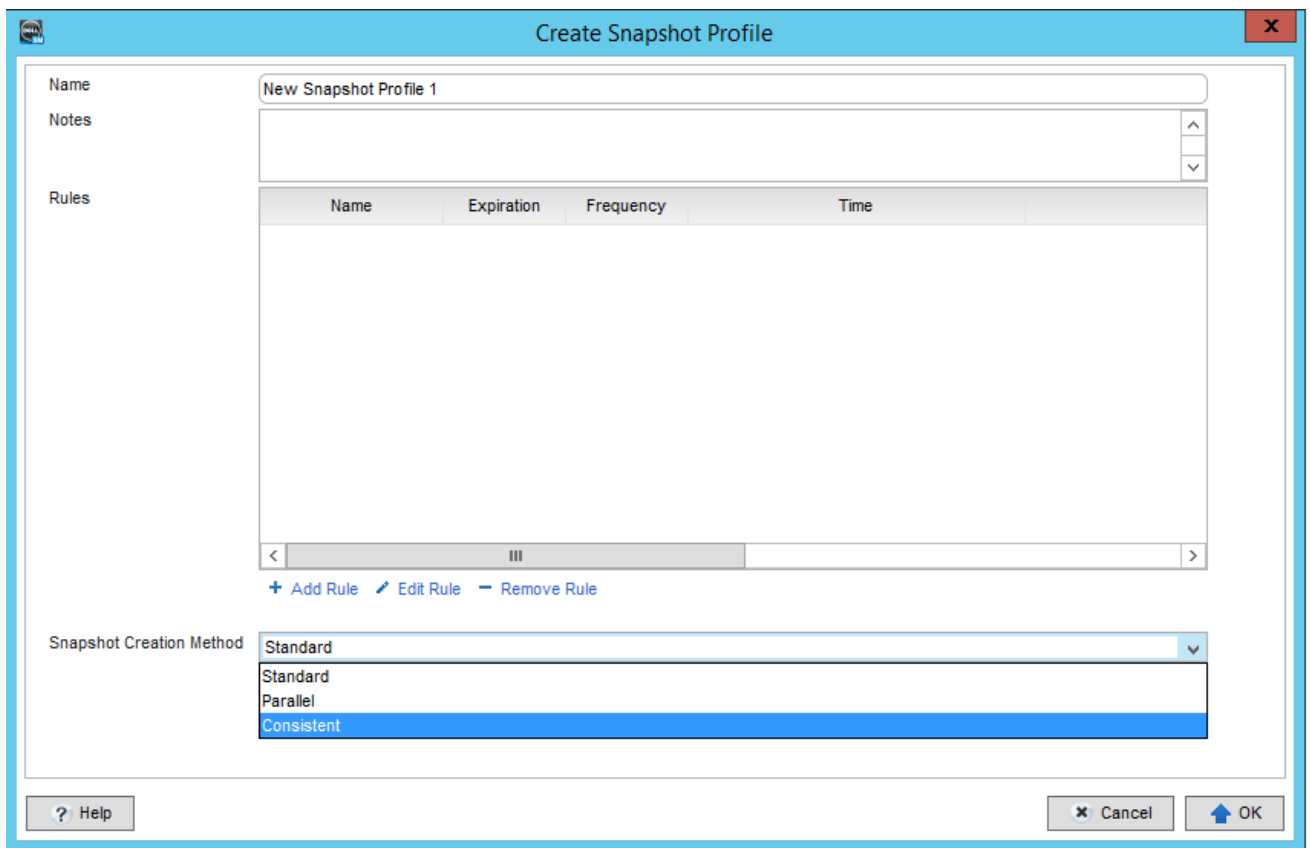
Figure 21   Consistent snapshot profile

Since a snapshot is taken on one or a set of multiple volumes, use care when defining the snapshot profile. For example, if there are multiple databases that reside in a set of one or more volumes, and a consistent snapshot is taken of that volume set, it will contain an image of all the databases. If a snapshot of only one of the databases was needed, disk space will be wasted by the other database snapshots. Also, with respect to flexibility in recovering a database without affecting other databases, if the same snapshot was used to recovery one of the databases, all the other databases in the snapshot would be recovered to the same point in time (PIT). This could leave the other databases in an unusable state. Therefore, Dell EMC recommends that each database be placed on its own set of volumes, and in its own snapshot profile.

When taking snapshots of a MySQL database, if the entire database resides on only one volume, a snapshot profile will be sufficient in providing a write-consistent PITC of the database. However, if a MySQL database spans multiple volumes, a consistent snapshot profile must be used to guarantee write-consistent PITC of the database. To simplify the management of snapshot profiles, Dell EMC recommends using the consistent snapshot profile for all MySQL snapshots, regardless of single or multi-volume databases. That way, should a single volume MySQL database ever expand to multiple volumes, a new snapshot profile does not need to be created. Simply adding the new volume to the existing consistent snapshot profile will suffice.

Note: As mentioned in this section, consistent snapshot profiles provide write-consistency and data integrity across all SC Series volumes in the snapshot. They do not provide write-consistency and data integrity at the application level. For that, a feature of MySQL must be used with consistent snapshot profiles. For more information on best practices with snapshots and MySQL, see sections 4.4.1 and 5.

Dell EMC recommends creating snapshots under these conditions:

- Immediately before the database goes live, before and after an upgrade, major change, repair or any major maintenance operation
- Once per day to allow Data Progression to move age-appropriate data more efficiently and effectively to other storage types and tiers
- On a schedule that satisfies appropriate business requirements for recovery point objective (RPO) and recovery time objective (RTO). The longer the time between snapshots, the greater the RPO becomes. For high-performance, business-critical systems, this may be a concern. To mitigate that concern, simply take more frequent snapshots. If RPO is unknown, a snapshot taken once per day of each MySQL database might be sufficient until RPO is defined.

Expiration times should be set to provide the ability to recovery from older snapshots. When snapshots expire, the SC Series array automatically releases the space used by the system back to the pagepool.

Data Progression moves snapshot data according the storage profile used by a volume. If the storage profile, Recommended (All Tiers), in an HDD system is used, snapshot data is immediately moved to T1 RAID 5-9/RAID 6-10, and then to other tiers, providing Data Progression is licensed. Writes to the volume still occur with RAID 10/RAID 10DM. This allows T1 space to be used more efficiently. See sections, Data progression and Storage profiles, for more information on Data Progression and storage profiles.

Dell EMC also recommends reviewing Data Progression pressure reports periodically to see data space consumption in a tier for the previous 30 days. The report will be useful in fine tuning snapshot profiles and retention periods for MySQL databases and capacity planning of the array. See section 3.14, Data Progression pressure reports, for more information.

Once a snapshot is created, view volumes can be created from it and presented to a server. The server can then use the view volume as a normal volume and perform the desired actions.

To create a snapshot profile:

1. In the **Storage Center** system tree, right-click node **Snapshot Profiles**.
2. Select **Create Snapshot Profile** from the context menu.
3. In drop-down field, **Snapshot Creation Method**, select value **Consistent**.
4. Set **Timeout Snapshot creation after seconds**, and **Expire incomplete Snapshot sets if Snapshot creation timeout is reached**.
5. Provide a name and optional Notes for the volume.
6. Select an on-demand or scheduled snapshot. On-demand snapshot profiles are useful when creating a backup or database clone from within a shell script that executes from a job schedule. If creating a scheduled snapshot, select **Add Rule**.
7. Specify the type of schedule: **Once, Daily, Weekly, Monthly**.
8. Specify the date and time, the frequency, and an expiration time for the snapshot. Click **Continue**.
9. If adding multiple time policies to the snapshot profile, repeat steps 6–8. After all time policies have been added to the schedule, click **OK** and click **Close**.

3.19   SC Series cache settings

The snapshot (Replay) feature is a licensed technology available on SC Series arrays that provides continuous space-efficient data protection. Snapshots create space-efficient, write-consistent, point-in-time copies (PITC) of one or more volumes based on a collection of rules that exist in a snapshot profile. Snapshots can be used for immediate recovery from data loss caused by hardware failures or logical errors. SC Series snapshot technology is different from other traditional PITCs because when a snapshot is taken data is not copied, it is only frozen. Should a write request occur to a frozen block, a new RAID 10 or RAID 10 DM page is allocated to the volume and the write will occur in that new page.

SC Series storage provides both a read and write cache to improve performance of I/O. By using the options on the Cache tab in the SC Series system properties, cache can be enabled or disabled globally for the entire array. Cache settings can also be set at the volume level.

Read cache anticipates the next read and holds it in quick volatile memory, thus improving read performance. Write cache holds written data in volatile memory until it can be safely stored on disk, thereby increasing write performance. By default, both read and write cache are enabled globally.

Because SSDs are memory devices, Dell EMC recommends disabling write cache in an SC Series all-flash array to maximize performance for MySQL applications. Disabling read cache in an SC Series all-flash array may or may not improve performance. In an array of all spinning media, Dell EMC recommends enabling read and write caches. Prior to changing the state of a cache, Dell EMC recommends testing applications to establish a baseline of performance metrics that can be compared against the same application tests with a different cache state. Because a change in either cache setting can have impact on performance, consult Dell Support in assessing the appropriateness of a change.

To view or change the cache state globally, perform the following:

1. In Dell Storage Manager, select the **Storage** view.
2. Right-click an SC Series array in the storage navigation tree.
3. Select **Edit Settings** from the context menu.
4. Select **Preferences**. Dell Storage Manager then displays the global cache properties.
5. Select or clear the **Read Cache** or **Write Cache** check boxes to enable or disable read or write cache.
6. Select **OK**.

For information on setting cache for a volume, see section 3.23, Volume cache settings.

3.20   Creating volumes in Dell Storage Manager

MySQL volumes can have very different performance requirements which can make the management of volumes in traditional storage very involved. There are a few benefits and features of SC Series storage that can help simply volume management for MySQL.

In SC Series arrays, all physical disks belong to a virtual disk pool so that the array can take advantage of all the physical disks in the system. When an SC Series volume is created, pages will be taken from the disk pool and assigned to volume as needed. All I/O against the volume is then spread across all the physical disks, yielding much better performance. The SC Series array also uses RAID levels and storage tiers for improved performance. For the best possible write performance, all writes will be to tier 1 RAID 10. As data ages, the SC Series array automatically migrates it to lower performing storage tiers. Both RAID and storage tiers are configurable for each volume.

Many of the SC Series features and options discussed up to this point can be specified at the time a volume is created. This is possible because of the way in which SC Series arrays virtualize storage. In this section, general guidelines about volume creation in SC Series arrays will be shown. Specific guidelines for volume creation in a MySQL environment are covered in the section, MySQL volume considerations.

To create a volume, perform the following:

1. In Dell Storage Manager, select the **Storage** view.
2. Select an SC Series array from the **Storage** navigation tree.
3. Expand the node **Volumes** from the **Storage Center** navigation tree. Right-click the location within the **Volumes** node where the volume should reside and select **Create Volume** from the context menu.
4. Specify the name, size, and units for the volume.
5. Set the read and write cache for the volume, the snapshot profile, and the destination server, and then select **OK**.

3.21   Volume size

Linux will discover volumes larger than 1 PB, but there are limitations to the partitions and filesystems that can be created on volumes of that capacity. The various Linux filesystems (such as ext3, ext4, xfs, zfs, and btfs) have specifications which vary over time. Consult the appropriate Linux distribution documentation to determine the thresholds and limitations of each filesystem type. The easiest way around some of the limitations is to use the whole volume/disk instead of applying a partition table to the volume. The entire volume/disk can be formatted with the filesystem of choice and mounted accordingly. This is accomplished by running mkfs on the device without applying a partition table. Dell EMC recommends creating fewer larger volumes rather than many small volumes, and making sure that the entire volume is used rather than using a partition. If volumes are oversized, the SC Series array will not waste space in the array as volumes are thin provisioned.

3.22   Preallocating storage

The SC Series array can preallocate storage to a volume when the volume is initially presented to an SC Series server object. This preallocation can be performed if the **Preallocate Storage** option is enabled system-wide in the SC Series array and selected during SC Series volume creation. Preallocating storage causes the SC Series array to take from the pagepool the maximum size specified for the volume, assign it to the volume, and write zero-filled data to the pages. After initial preallocation, should the volume size be increased in the SC Series array, any write that requires a new page be allocated to the volume will cause the array to thinly provision a new non-zero page. Also, should a snapshot be created on the volume, any write to a frozen page will cause the SC Series array to thinly provision a new non-zero page to the volume.

Therefore, expanding the size of a preallocated volume, or creating snapshots on the preallocated volume, negates the benefit of preallocating storage. With respect to snapshots, the reason for this is that when a snapshot of a volume is created, all pages (with zero-filled or non-zero data) within the volume are frozen. Should there be an attempt to write to a frozen page, a new page we be thinly provisioned to the volume and the modification will occur on the new page and not on the preallocated zero-filled frozen page. This negates any of the benefits from preallocation.
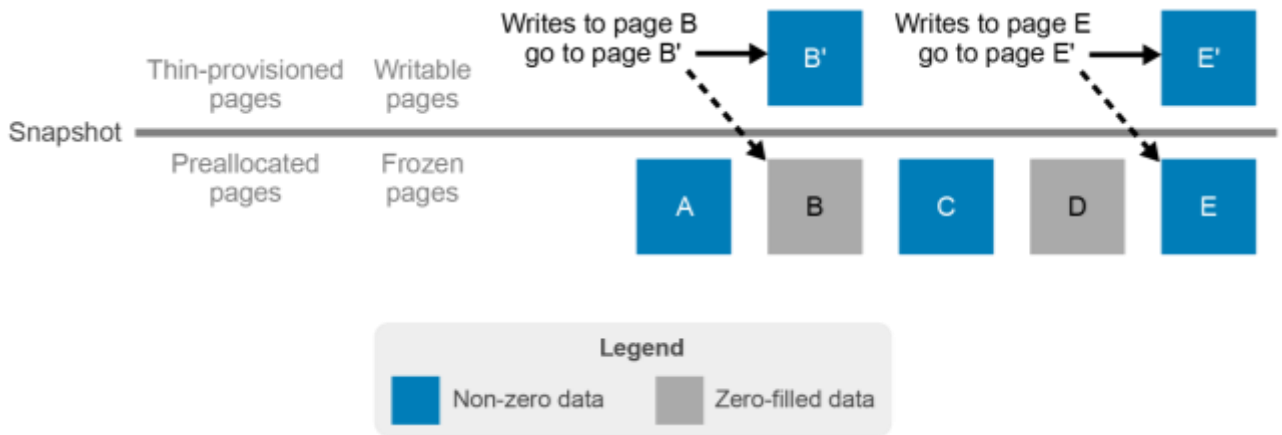
Figure 22   Frozen preallocated pages

Therefore, for high-performance database systems, where performance can be affected by dynamic block allocation, consider the tradeoff between preallocating and not preallocating space, especially if snapshots are a requirement. Also, consider the tradeoff between expanding or not expanding a preallocated volume. Choosing to preallocate upon volume creation means that applying a snapshot profile to the volume or expanding the volume size at any point in the future will undo the benefit of preallocation. For this reason, Dell EMC recommends that snapshots should never be used on preallocated volumes, and that preallocated volumes should never be expanded.

Note: The option to **Preallocate Storage** is only available after the volume is presented to a server, and only if **Allow Preallocate Storage Selection** option has been enabled in DSM and no data has been written to the volume.

To preallocate storage, right-click a volume in the system tree of **Storage Center** and select **Preallocate Storage**.
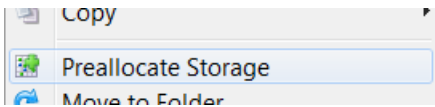
Figure 23   Preallocating storage on a volume

If Preallocate Storage is selected, the SC Series array will take from the pagepool the maximum size specified for the volume. If the pagepool is 20 TB, and the maximum size for a volume is 1 TB, but only 1 MB of data is written to the volume, the volume will be 1 TB in size after preallocation, and the pagepool will have 19 TB of remaining available space. The Preallocate Storage feature runs in the background and takes time depending on the size of the volume being preallocated.

To see the progress of any preallocation, change the view in the SC Series interface to **Background Processes**.

Figure 24   Progress on preallocation process

3.23   Volume cache settings

The snapshot (Replay) feature is a licensed technology available on SC Series arrays that provides continuous space-efficient data protection. Snapshots create space-efficient, write-consistent, point-in-time copies (PITC) of one or more volumes based on a collection of rules that exist in a snapshot profile. Snapshots can be used for immediate recovery from data loss caused by hardware failures or logical errors. SC Series snapshot technology is different from other traditional PITCs because when a snapshot is taken data is not copied, it is only frozen. Should a write request occur to a frozen block, a new RAID 10 or RAID 10 DM page is allocated to the volume and the write will occur in that new page.

Dell EMC recommends disabling write cache for all volumes using SSDs. Disabling read cache for volumes using SSDs is application-specific and may or may not improve performance. In any event, Dell EMC recommends testing applications prior to disabling read cache to establish a baseline of performance metrics can be compared to the same applications tests on a disable read cache.

To view or change cache settings of a volume, use the following steps:

1. In the **Storage Center** navigation tree, select the node **Volumes**. Then locate and right-click the desired volume, and select **Edit Settings** from the context menu.
2. Select or clear the **Read Cache** or **Write Cache** check boxes.
3. Select **OK**.

For information on setting cache globally, see section 3.19.

3.24   Server multipathing

An I/O path generally consists of an initiator port, fabric port, target port, and LUN. Each permutation of this I/O path is considered an independent path. Dynamic multipathing/failover tools aggregate these independent paths into a single logical path. This abstraction provides I/O load balancing across the host bus adapters (HBA), as well as non-disruptive failovers on I/O path failures. The use of dynamic multipath I/O combined with SC Series storage technologies yields the best high availability and performance storage solution for a MySQL database.

To provide multipath redundancy at the path level in Linux, the server must have at least two active and configured FC initiator ports, and multipathing IO (MPIO) must be enabled in the OS. MPIO is provided by enabling Device Mapper Multipath (DM-multipath). To configure server multipathing, file /etc/multipath.conf must be modified. The following includes a snippet of multipath.conf used for the content of this paper:

```
defaults {
        udev_dir                /dev
        polling_interval        5
        path_grouping_policy    failover
        getuid_callout          "/lib/udev/scsi_id --whitelisted --
device=/dev/%n"
        prio                    const
        path_checker            directio
        rr_min_io               1000
        rr_weight               uniform
        failback                manual
        no_path_retry           fail
        user_friendly_names     yes
}
```

```
devices {
        device {
                vendor "COMPELNT"
                product "Compellent Vol"
                features "0"
                hardware_handler "0"
                no_path_retry fail
        }
}
blacklist {
        devnode "^(ram|raw|loop|fd|md|dm-|sr|scd|st)[0-9]*"
        devnode "^hd[a-z]*"
        devnode "^cciss!c[0-9]d[0-9]*[p[0-9]*]"
}

multipaths {
        multipath {
                wwid "36000d3100002b9000000000000081c82"
                alias mysql1
        }
<snippit>
```

For additional information on multipath and configuring multipathing, refer to the best practices document *Dell EMC SC Series with Red Hat Enterprise Linux 6x* and other references in appendix A. In addition to a properly configured /etc/multipath.conf file, the server object in the SC Series array must also be configured for multipathing (see section 3.4).

Once a server object is defined to support multipathing, a volume can be configured to use multipathing when it is presented to that server. To configure multipathing on a volume in the SC Series array, execute the following steps:

1. Right-click the volume from the **Storage Center** navigation tree in Dell Storage Manager.
2. Select **Map Volume to Server** from the context menu.
3. Locate and select the server for volume presentation.
4. Select **Advanced Options**.
5. Select the check box **Map to All Available Server Ports**, or unselect the check box and select the desired HBAs to use for multipathing (see section, Soft (WWW) zoning). By default, all appropriate mappings should be selected.
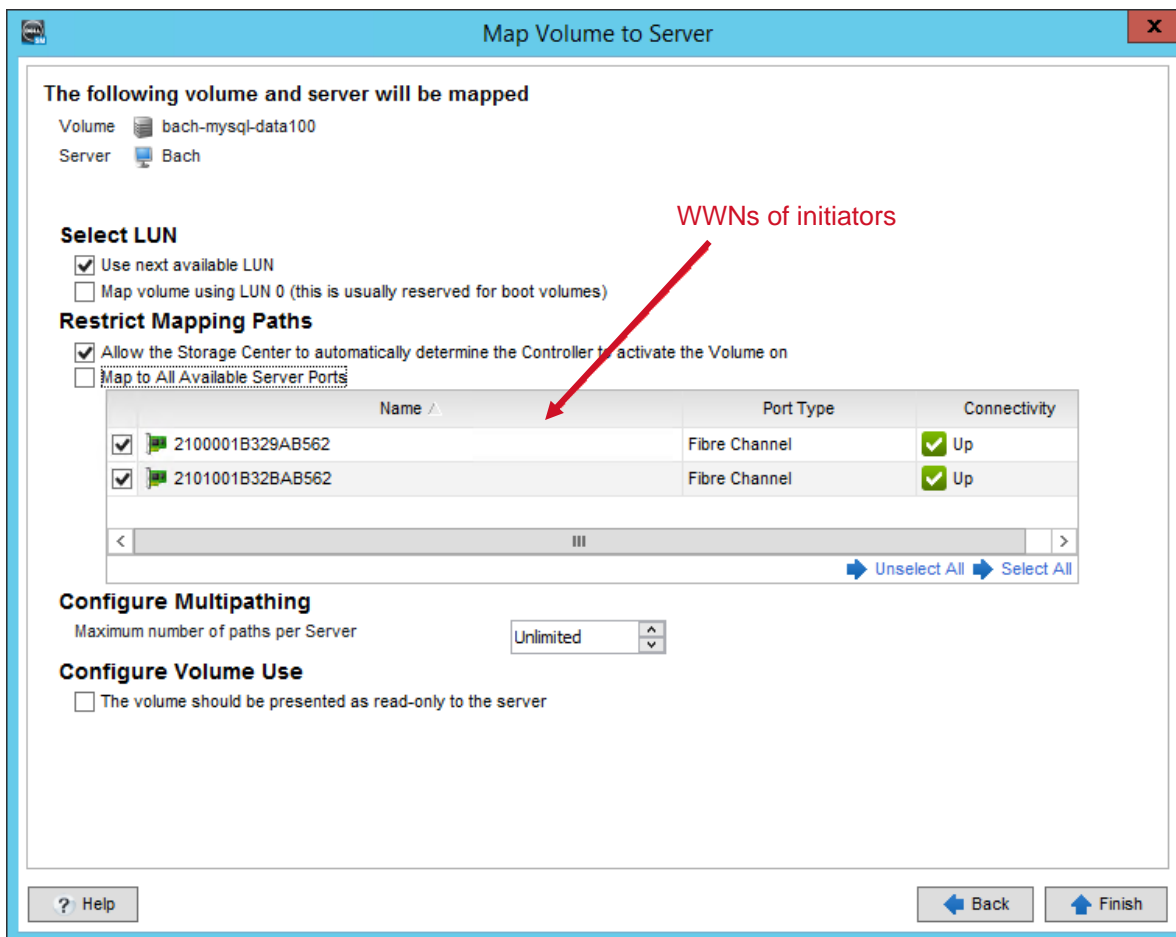
Figure 25   Defining mapping paths for volume presentation

Leave the check-boxes for **Map to All Available Server Ports** and **Maximum number of paths per Server** default to take advantage of the maximum number of paths provided by the OS. I/O requests for a volume are processed by the controller that owns the volume, and a volume can only be owned by one controller at a time. By default, all available paths are selected. This allows the SC Series array to automatically switch controller ownership of a volume as needed to evenly distribute I/Os across both controllers. Therefore, in a MySQL environment that has high performance requirements, Dell EMC recommends leaving the mapping default, so that the SC Series array can balance the volume mappings between both controllers.

6.   Select **Continue**, and **Create Now**.

4 MySQL

MySQL is available in several editions.

MySQL Community Edition is available on over 20 platforms, is free for download, and is made available under GPL licensing. Support is provided by a rather large active community of open source developers. Some product features include:

- Pluggable storage engine architecture
- Multiple storage engines: InnoDB, MyISAM
- Stored procedures

Commercial editions include the following:

- MySQL Standard Edition
- MySQL Enterprise Edition
- MySQL Cluster Carrier Grade Edition

MySQL Standard Edition delivers scalable, reliable, and high-performance online transaction processing (OLTP) solutions. Some product features include:

- InnoDB storage engine
- ACID compliant
- Low total cost of ownership (TCO)
- MySQL Workbench

MySQL Enterprise Edition is the most comprehensive and integrated set of advanced features, tools, and technical support. Some product features include:

- All the features of MySQL Standard Edition
- Low risk and TCO in developing, deploying, and managing business applications
- MySQL Enterprise Backup which reduces the risk of data loss by providing hot backups; compression; full, incremental and partial backups; and point-in-time recovery
- Highly available database infrastructures, including replication, and Oracle® Clusterware
- Scalability to sustain existing performance or scalability with increasing query and data loads

The content of this document was developed using MySQL 5.7 Community Edition. For additional information regarding available MySQL editions, refer to https://www.mysql.com/products/.

4.1 MySQL database and application design

Database and application design can greatly impact the performance of an environment. The database architect needs to look at the most efficient way to design all database objects (schemas, tables, indexes, and columns) while minimizing I/O. The application developers need to ensure query and data manipulation language (DML) statements are constructed in a way to minimize the amount of data selected or updated. This will make it easier to maintain the applications and database as they evolve.

For more information on MySQL database design and query optimization, see the MySQL articles, Optimizing Database Structure, and Optimizing SELECT Statements.

4.2　The ACID model and MySQL

ACID (Atomicity, Consistency, Isolation, and Durability) is a concept or set of design principles in relational database theory that guarantees database transactions are processed reliably and can play an important part in determining other business application design strategies. Any database management system that does not adhere to ACID properties cannot be considered reliable.

Atomicity: The all-or-nothing approach to database updates for anyone beyond the user performing the operation. A single logical update, known as a transaction, is either committed or aborted in its entirety.

Consistency: Changes to database values are consistent with changes to other values in the same database (for example, value changes must be valid according to all defined triggers, constraints, rules, cascades). Some related MySQL features include InnoDB double write buffer and crash recovery.

Isolation: The relational database management system (RDBMS) controls that prevent conflicts between concurrent transactions. If multiple transactions are updating data, one transaction is not allowed to overwrite the changes being made by another transaction. The resulting database state should be the same as if the transactions were executed serially.

Durability: Committed updates must never be lost, and can be used to recover a system should it fail.

MySQL includes components such as the InnoDB storage engine that adhere closely to the ACID model. Although MySQL InnoDB follows the ACID properties, it does not guarantee that all databases will be recoverable in the event of a server crash. For example, data definition language (DDL) operations are not recoverable should a server crash occur. It can occur where CREATE TABLE can construct half-open tables where InnoDB and the filesystem are inconsistent with each other. This can result in a situation that requires a complete dump and reload. There are ways to address this, but they are out of scope of this document.

Other MySQL storage engines do not adhere to the ACID properties and therefore should not be used where ACID compliance and space reclamation are required. Also, the lack of ACID compliance may affect strategies for making write-consistent database backups regardless of the backend storage being used.

4.3　MySQL storage engines

Storage engines manage SQL operations on tables. Their availability depends on the edition and version of MySQL. Understanding the different storage engines is vital to knowing when and how to create write-consistent SC Series snapshots (replays) of a database or table for backup and cloning purposes, and when thin reclamation can be realized in SC Series arrays. Of the available storage engines in MySQL, the two most popular are InnoDB and MyISAM.

The show engines command displays the details of the storage engines:

```
mysql> show engines;
+-------------------+--------+------- … ----+-------------+------+---------
--+
| Engine            | Support | Comment …     | Transactions | XA   |
Savepoints |
+-------------------+--------+------- … ----+-------------+------+---------
--+
| MRG_MYISAM        | YES     | Collect …     | NO           | NO   | NO
|
| CSV               | YES     | CSV sto …     | NO           | NO   | NO
|
| PERFORMANCE_SCHEMA | YES    | Perform …     | NO           | NO   | NO
|
| BLACKHOLE         | YES     | /dev/nu … rs) | NO           | NO   | NO
|
| InnoDB            | DEFAULT | Support …     | YES          | YES  | YES
|
| MyISAM            | YES     | MyISAM  …     | NO           | NO   | NO
|
| ARCHIVE           | YES     | Archive …     | NO           | NO   | NO
|
| MEMORY            | YES     | Hash ba …     | NO           | NO   | NO
|
| FEDERATED         | NO      | Federat …     | NULL         | NULL | NULL
|
+-------------------+--------+------- … ----+-------------+------+---------
--+
9 rows in set (0.00 sec)
```

The default storage engine (table type) is not the same in all versions and editions of MySQL, but as shown in the storage engine details, InnoDB is the default. InnoDB is also the most general-purpose engine, and for most cases it is the recommended engine. For additional information on table types, see section, MySQL table types.

4.4 MySQL table types

MySQL supports a mix of different storage engines, also known as table engines or table types, within the same database. Table types can impact the ability to create a completely recoverable, write-consistent database backup or clone that is ACID compliant regardless of the underlying storage technology. Because of this, it is important to understand which table types will yield write-consistent database snapshots, and the location used by the storage engines to store table data, index, metadata, and binary transaction logs. This section briefly discusses the two most common table types: MyISAM and InnoDB.

Before beginning a discussion of MyISAM and InnoDB, it is important to understand how a table type or table engine is set for a table, which can help you take advantage of some SC Series features. The type is an attribute of a MySQL table, which specifies which storage engine will be used to govern the operations on the table. It is specified when a table is created or altered.

```
CREATE TABLE <TableName> (<ColName> <Coltype>) [ENGINE = MYISAM];
```

```
ALTER TABLE <TableName> [ENGINE=INNODB];
```

If the ENGINE clause is not specified when creating the table, the table type is set to InnoDB.

```
mysql> create table mytest (c1 char);
Query OK, 0 rows affected (0.05 sec)

mysql> select table_name, engine
    -> from information_schema.tables
    -> where table_schema='mysql' and table_name='mytest';
+------------+--------+
| table_name | engine |
+------------+--------+
| mytest     | InnoDB |
+------------+--------+
1 row in set (0.00 sec)
```

There are several ways to determine the table type of existing tables, but should the table be damaged or corrupted, not all methods will work. However, the following select command should work in most cases once the database has been selected:

```
mysql> select table_name, engine
    -> from information_schema.tables
    -> where table_schema='mysql';
+---------------------------+--------+
| table_name                | engine |
+---------------------------+--------+
<snippet>
| general_log               | CSV    |
| innodb_index_stats        | InnoDB |
| innodb_table_stats        | InnoDB |
| ndb_binlog_index          | MyISAM |
<snippet>
| user                      | MyISAM |
+---------------------------+--------+
31 rows in set (0.00 sec)
```

4.4.1 InnoDB table type, tablespaces, and SC Series snapshots and space reclamation

InnoDB is the only table type that supports savepoints, transactions, and distributed transactions (XA). Because of this, Dell EMC recommends using InnoDB tables whenever possible. InnoDB can be most beneficial under the following conditions or requirements:

- Consistent image snapshots when used with SC Series storage arrays
- ACID compliance
- Commit, rollback, and crash-recovery
- Row-level locking with no escalating to coarser granularity locks
- Increased multi-user concurrency and performance
- Primary keys or clustered indexes
- Foreign keys to support referential-integrity constraints
- Transactional database with a mix of writes and reads that do not require full text indexing

- Applications requiring frequent table lock escalations under heavy but slow reads for long periods of time
- If a read is slow or has not completed, and a read-write is waiting on the first read to finish
- Separate buffer pools for caching data and indexes
- Tables not limited to OS with 2 GB file size limit
- Benefits of MySQL file-per-table tablespaces can be realized

The following MySQL statements will set a table's type to InnoDB:

```
CREATE TABLE <TableName> (<ColName> <Coltype>);
CREATE TABLE <TableName> (<ColName> <Coltype>) ENGINE = INNODB;
ALTER TABLE <TableName> ENGINE=INNODB;
```

InnoDB uses the concept of a tablespace to store its tables and indexes. By default, each InnoDB table (and its indexes) is stored in its own tablespace (a set of files). This storage method is called file-per-table tablespace, and it is enabled by the configuration option innodb_file_per_table. Alternatively, all InnoDB tables (and their indexes) can be stored in the system tablespace which may consist of multiple files or raw disk partitions. To determine the current setting of option innodb_file_per_table, execute the following command while logged in to the desired database.

```
mysql> show variables where Variable_name = 'innodb_file_per_table';
+-----------------------+-------+
| Variable_name         | Value |
+-----------------------+-------+
| innodb_file_per_table | ON    |
+-----------------------+-------+
1 row in set (0.00 sec)
```

Note: When taking SC Series snapshots of InnoDB tables, if the corresponding tablespace spans multiple volumes, make sure that all volumes are placed within a SC consistent snapshot profile, and then create the snapshot using that schedule after executing FLUSH TABLES WITH READ LOCK in MySQL. This guarantees that a write-consistent image of the database and LUNs are made at the same point in time, providing the ability for crash recovery. See sections, Snapshot profile and MySQL backups and SC Series snapshots, for more information.

4.4.1.1 file-per-table tablespaces

There are two files per each file_per_table tablespace and they reside in the storage directory of the database. One file contains the table format, the other contains the table data and associated indexes. The following shows the files in the file_per_table tablespace of table engine_cost:

```
# pwd
/var/lib/mysql/mysql
 # ls -ltr engine_cost*
-rw-r-----. 1 mysql mysql  8780 Feb 22 14:01 engine_cost.frm
-rw-r-----. 1 mysql mysql 98304 Feb 22 14:01 engine_cost.ibd
```

Table 12   File-per-table tablespace files

InnoDB table file	File description
<TableName>.frm	Table format
<TableName>.ibd	Table data and indexes

Disk space can be reclaimed in SC Series arrays when truncating or dropping a MySQL InnoDB table that uses file-per-table tablespaces, assuming the filesystem supports the SCSI UNMAP command and the volume was mounted with option DISCARD. See section, Ext4 file systems and MySQL, for more information on mounting ext4 filesystems.

Some advantages of file-per-table tablespaces include:

- SC Series space reclamation can be realized
- Faster truncate table operations and easier movement (including tablespace transports) of tables
- OPTIMIZE TABLE can be used to compact and recreate a file-per-table tablespace and return the space to the system for space reclamation
- Improved, fine-grained I/O optimization, space management, and backups because each table can exist on its own separate storage device
- MySQL compressed and dynamic row formants can be used
- More efficient storage for BLOB or TEXT columns
- Improved chances for successful recovery
- Each file-per-table tablespace has a 64 TB size limit

When an innodb_file_per_table tablespace file becomes full, MySQL will auto extend the file by small amounts, after which 4 MB extensions will be used. The extension size cannot be altered, and since it is a multiple of the default SC datapage size (2 MB), there is no need for adjustments.

Dell EMC recommends the use of file-per-table over system tablespaces. However, since each business requirement is unique, it is advised to examine the benefits of both, and use the tablespace type that best fits the need.

4.4.1.2   System tablespace

System tablespace files (ibdata<n>) and associated system tablespace logs reside in the storage directory for the mysql installation (default is /var/lib/mysql). The following examples show one file in the system tablespace, its two redo logs, and the type of data stored in the file:

```
# pwd
/var/lib/mysql
# ls -ltr ibdata* ib_logfile*
-rw-r-----. 1 mysql mysql 50331648 Mar  4 14:14 ib_logfile1
-rw-r-----. 1 mysql mysql 50331648 Mar  7 10:20 ib_logfile0
-rw-r-----. 1 mysql mysql 12582912 Mar  7 10:20 ibdata1
```

Should a table that exists in the system tablespace be dropped or truncated, the free space can only be used for new InnoDB tables.

Some advantages of using the system tablespace are:

- Less wasted space compared to file-per-table tablespace because all unused space resides in one tablespace
- Less fsync operations
- Fewer open file handles per table
- Backward compatibility with MySQL 5.5 or 5.1
- Raw disk partitions can be used for system tablespaces

When a system tablespace file becomes full, MySQL will auto extend the file in increments of MB defined by option innodb_autoextend_increment. The default size is 64. For best performance and efficient space management, this value should be the same as or a multiple of the SC Series datapage size (512 KB, 2 MB, or 4 MB).

Use the following MySQL query to verify the increment size.

```
mysql> show variables where Variable_name = 'innodb_autoextend_increment';
+-----------------------------+-------+
| Variable_name               | Value |
+-----------------------------+-------+
| innodb_autoextend_increment | 64    |
+-----------------------------+-------+
1 row in set (0.00 sec)
```

To determine the current datapage size in SC Series arrays, log in to Dell Storage Manager (DSM) and click **Storage Management** > **Disk** > **Configure Storage**.

The **Storage Type** displays the current datapage size of the **Assigned disk** folder.
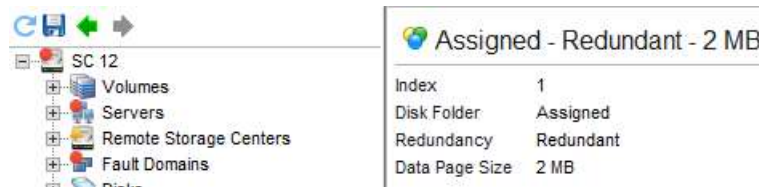
Figure 26   SC Series data page size of volume extents

If it is necessary to create more granular SC Series snapshots for just the system tablespace, or to use a dedicated SC Series volume for the system tablespace, create the system tablespace at a different location by specifying the following two parameters in section [mysqld] of the mysql configuration file (default is /etc/my.cnf). Then create an SC Series consistent snapshot profile that includes only the LUNs for the system tablespace:

```
innodb_data_home_dir=<base directory for the system tablespace>
innodb_data_file_path=<datafile_spec1[;datafile_spec2[;…datafile_specn]…]>
```

Where:

```
datafile_spec<1..n-1>::file_name:file_size
datafile_spec<n>      ::file_name:file_size[:autoextend[:max:max_file_size]]
```

MySQL concatenates the values of the two parameters to determine the name and location of the tablespace:

```
<innodb_data_home_dir>/<innodb_data_file_path>
```

System tablespace logs can also be located to different locations and reconfigured as necessary per business requirements. See the MySQL article, InnoDB Redo Log, for more information.

innodb_data_home_dir is optional. If not specified, MySQL uses the data directory (the ./ directory). If an empty string is specified, absolute file paths can be used in innodb_data_file_path.

file_size is specified in units of M or G for megabytes or gigabytes.

autoextend directs MySQL to extend the file in increments of 8 MB when it runs out of free space in the tablespace.

max:max_file_size should be used when **autoextend** is specified and there is a possibility of the file size exceeding the file size limit of the filesystem. Since InnoDB is unaware of the filesystem maximum file size, if the maximum size is not specified and autoextend pushes the file beyond the filesystem maximum, fatal errors and possible crashing of the system could occur.

If **autoextend** is used, Dell EMC recommends using **max** and set **max_file_size** to the maximum file size (max_file_size is specified in units of M or G for megabytes or gigabytes). If **max** and **max_file_size** are not used, Dell EMC strongly recommends taking more frequent snapshots with longer expiration periods of the MySQL environment to allow for more recovery options, should autoextend push a file beyond filesystem limits.

4.4.2   MyISAM table type

MyISAM is based on the predecessor ISAM storage engine, and can be most beneficial under certain conditions or requirements:

- ACID compliance not required
- Full-text index searches
- Application requires a disproportionate, heavy read-only over writes
- Application will have infrequent table lock escalations
- Simpler to design and create then InnoDB
- Better for novices
- No foreign key relationships
- Proposed to be better for read-intensive tables than InnoDB
- No need for cascading deletes and updates
- Rollback capabilities not needed

The following MySQL statements will set a table's type to MyISAM:

```
CREATE TABLE <TableName> (<ColName> <Coltype>) ENGINE = MYISAM;
ALTER TABLE <TableName> ENGINE=MYISAM;
```

MyISAM stores each of its tables in three separate files, where the file extension indicates its function. The following shows the three MyISAM files that belong to table tables_priv in database mysql. Although the following example shows all three files in the same directory, the data file and index file can reside in different directories on different physical devices to improve performance with the DATA DIRECTORY and INDEX DIRECTORY table options of the CREATE TABLE command:

```
# pwd
/var/lib/mysql/mysql
# ls -ltr tables_priv.*
-rw-r-----. 1 mysql mysql 8955 Feb 22 14:01 tables_priv.frm
-rw-r-----. 1 mysql mysql  899 Feb 22 14:01 tables_priv.MYD
-rw-r-----. 1 mysql mysql 8192 Feb 22 14:01 tables_priv.MYI
#
```

Table 13   MyISAM table files

MyISAM table file	File description
<TableName>.frm	Table format
<TableName>.MYD	Contains the table data (My Data = MYD)
<TableName>.MYI	Contains the table indexes (My Index = MYI)

4.4.3   Converting from MyISAM to InnoDB

If a table type needs to be changed from MyISAM to InnoDB, it can be done so dynamically by executing the following MySQL command, providing the table is not using full-text indexing. MyISAM tables that are used with full-text indexing cannot be converted.

```
ALTER TABLE <TableName> ENGINE=INNODB;
```

4.5   MySQL volume considerations

A MySQL database is made up of many different files having different performance and snapshot requirements. With a standard default installation, MySQL databases, encryption keys for tablespace encryption, mysqldump backups and miscellaneous sql and txt files are stored in /var/lib/mysql.

```
# ls -ltr /var/lib  | grep mysql
drwxr-x---. 2 mysql    mysql    4096 Feb 22 14:01 mysql-keyring
drwxr-x--x. 6 mysql    mysql    4096 Feb 29 12:23 mysql
drwxr-x---. 3 mysql    mysql    4096 Feb 29 12:41 mysql-files
```

Should an SC Series snapshot be required of this LUN, the snapshot would contain the entire boot volume. In some environments, this single volume deployment in the boot LUN might be sufficient. However, Dell EMC recommends using at least one dedicated (non-boot) volume per MySQL instance, per logical function as needed. This provides the ability to:

- Expand the options for backup and recovery strategies
- Expand the options for cloning databases without wasting space from other databases in the snapshot clone
- Take advantage of the SC Series ability to distribute the IO requests of the volumes evenly between the controllers thereby leveraging the resources on both controllers
- Have more options to isolate multiple databases on the same server especially when running multiple instances of MySQL
- Have more options to fine tune QLogic settings for specific MySQL requirements
- Have greater flexibility, manageability, and control over moving a database to a different tier of storage for improved performance and greater efficiencies on disk usage.

Table 14 provides one example of multiple dedicated volumes for multiple MySQL instances on a server, along with a possible generic performance ranking of either high or low, and generic snapshot ranking of either frequent or infrequent as required by MySQL. The performance ranking will be helpful in determining which storage profile could be used for the volume. Frequency of snapshots will be helpful in determining which snapshot profile to use for the volume.

Table 14   Fine-grained volume configuration in MySQL

Volume name	ext4 mount point	Volume description	Performance	Snapshot frequency
<server>-boot	/	Boot partition	Low	Infrequent
<server>-mysql1-data	/mysql1-data	Data for MySQL instance 1	High	Frequent
<server>-mysql1-stbs	/mysql1-innodb-stbs	InnoDB system tablespace instance 1	High *	Frequent
<server>-mysql1-binlog	/mysql1-redo	Binary log MySQL instance 1 redo	High *	Frequent
<server>-mysql1-undo	/mysql1-undo	InnoDB undo for MySQL instance 1	High*	Frequent
<server>-mysql1-backup	/mysql1-backup	MySQL instance 1 backups	Low	Optional
<server>-mysql1-audit	/mysql1-aud	Audit for MySQL instance 1	Low*	Frequent
<server>-mysql1-tmp	/mysql1-tmp	Tmp files MySQL instance 1	High*	Frequent
<server>-mysql2-data	/mysql2-data	Data for MySQL instance 2	High	Frequent
<server>-mysql2-stbs	/mysql2-innodb-stbs	InnoDB system tablespace instance 2	High**	Frequent
<server>-mysql2-binlog	/mysql2-redo	Binary log MySQL instance 2 redo	High**	Frequent
<server>-mysql2-undo	/mysql2-undo	InnoDB undo for MySQL instance 1	High**	Frequent
<server>-mysql2-backup	/mysql2-backup	MySQL instance 2 backups	Low	Optional
<server>-mysql2-audit	/mysql2-aud	Audit for MySQL instance 2	Low**	Frequent
<server>-mysql2-tmp	/mysql2-tmp	Tmp files MySQL instance 2	High**	Frequent

*Same performance and snapshot requirements as Data for MySQL instance 1.
**Same performance and snapshot requirements as Data for MySQL instance 2.

If Table 14 contains too many dedicated LUNs for the business requirement, it can be simplified. For example, if log files, data directory, and PID file can exist on a single volume for each MySQL instance, then option --basedir can be used to direct MySQL to use that one location for certain files. Table 15 provides an example of a simplified volume layout:

Table 15   Moderate volume configuration in MySQL

Volume name	ext4 mount point	Volume description	Performance	Snapshot frequency
<server>-boot	/	Boot partition	Low	Infrequent
<server>-mysql1	/mysql1	Data, index, logs for MySQL instance 1	High	Frequent
<server>-mysql1-backup	/mysql1-backup	MySQL instance 1 backups	Low	Optional
<server>-mysql1-tmp	/mysql1-tmp	Tmp files MySQL instance 1	High	Frequent
<server>-mysql2	/mysql2	Data, index, logs for MySQL instance 2	High	Frequent
<server>-mysql2-backup	/mysql2-backup	MySQL instance 2 backups	Low	Optional
<server>-mysql2-tmp	/mysql2-tmp	Tmp files MySQL instance 2	High	Frequent

With respect to MySQL temporary tables, they are only active for the duration of the MySQL connection that requires them, and generally have a short live span. As such, there is a strong possibility that they will not require data progression. But they will be required in the snapshots. The storage class selected for temporary tables should be in line with the performance requirements of the temporary data.

The number of volumes can be reduced even further. Backups could be in --basedir location, and /tmp, /var/tmp, or --basedir could be used for MySQL temporary files. When looking at both the MySQL and SC Series volume considerations, Dell EMC recommends starting with at least the following minimal volume configuration for MySQL and adjusting it for business requirements where finer grain control over volume content, backup control, temporary usage per MySQL instance are required.

Table 16   Minimal volume configuration in MySQL

Volume name	ext4 mount point	Volume description	Performance	Snapshot frequency
<server>-boot	/	Boot partition	Low	Infrequent
<server>-mysql1	/mysql1	Data, index, logs for MySQL instance 1	High	Frequent
<server>-mysqln	/mysqln	Data, index, logs for MySQL instance n	High	Frequent

In the previous examples, the root directory of the ext4 file systems (mount point) was used for the MySQL data directory, but that does not have to be the case. If it is, Dell EMC strongly recommends using the following MySQL option and setting when calling MySQL:

```
--ignore-db-dir=lost+found
```

If it is not, MySQL will experience an issue when initializing its environment because it expects the base directory to be void of all entries. There are several ways to declare and set **--ignore-db-dir:**

- **--Ignore-db-dir** is not a dynamic MySQL option, so add and set it in section **[mysqld]** of section in **/etc/my.cnf**. Once it is added, start the MySQL service.

```
# cat /etc/my.cnf
<snippet>
[mysqld]
<snippet>
ignore-db-dir=lost+found
# service mysqld start
Starting mysqld:                                        [  OK  ]
```

- MySQL service script **/etc/init.d/mysqld** supports passing additional arguments:

```
service <script> <command> [options]
```

Therefore, if **--Ignore-db-dir** is not added to **/etc/my.cnf**, it can be specified when starting the MySQL service:

```
# service mysqld start --ignore-db-dir=lost+found
Starting mysqld:                                        [  OK  ]
```

To change the default location for MySQL, amend the following steps necessary.

1. Before making any changes, stop mysql service.

```
service mysqld stop
```

2. Create an SC Series snapshot of the entire MySQL environment.
3. Create a new SC Series volume for the data directory and mount it. Then copying the contents of the default location (var/lib/mysql) to the new location as per requirements.

```
cp -rap /var/lib/mysql /mysql1
```

4. Set the ownership and privileges of the new location for MySQL.

```
chown mysql.mysql /mysql1
```

5. Edit the MySQL configuration file /etc/my.cnf and update values of datadir and socket variable.

   Old:

```
datadir=/var/lib/mysql
socket=/var/lib/mysql/mysql.sock
```

   New:

```
datadir=/mysql1
socket=/mysql1/mysql.sock
```

6. Adjust access control security policies accordingly in SELinux.

Note: By default, MySQL is installed to use SELinux. See section 4.10 for more information.

---

7. Restart MySQL.

```
service mysqld start
```

4.6   Ext4 file systems and MySQL

Because ext4 supports the SCSI UNMAP command, Dell EMC recommends using ext4 file systems for at least the datadir location. SCSI UNMAP allows space to be reclaimed in SC Series arrays (version 5.4 or newer). If a MySQL MyISAM or InnoDB file-per-table tablespace is truncated or deleted, the filesystem will release the space and SC Series arrays will return the space back to it pagepool. MySQL supports other file systems too, but if space reclamation is important, then it is recommended to use ext4. To enable space reclamation functionality, the discard flag must be issued when mounting the ext4 filesystem or placed in file **/etc/fstab**.

```
# mkfs -t ext4 /dev/mapper/mysql-inst1
# mount -o discard /dev/mapper/mysql-inst1 /mysql1
```

From **/etc/fstab,** it might look like this:

```
/dev/mapper/mysql-inst1          /mysql1      ext4     defaults,discard    0 0
```

Dell EMC strongly recommends that ext4 file systems be created on the entire disk, and not on a logical or primary partition. This is due to the amount of careful planning and consideration, and to the level of advanced knowledge of Linux systems that would be required to expand the underlying volume. Expanding a file system that resides directly on a physical disk is much less complicated and risky.

For more information see the article, Native Free Space Recovery in Red Hat Linux, and additional material from the subscription-based Red Hat Customer Portal.

4.7   Expanding SC Series volumes

Expanding file systems that reside on a logical or primary partition is not recommended. However, expanding file systems that reside on the entire physical disk is. As with any partition or filesystem operation, there is some risk of data loss. Dell EMC recommends creating a snapshot of the entire database using a consistent snapshot profile prior to expanding a volume. Creating the snapshot will require the use of **FLUSH TABLES WITH READ LOCK** in MySQL before the snapshot is created.

For more information on volume expansion, refer to:

- *Dell EMC SC Series with Red Hat Enterprise Linux 6x* best practices
- Section 4.4.1, InnoDB table type, tablespaces, and SC Series snapshots and space reclamation
- Section 5, MySQL backups and SC Series snapshots

4.8   Deleting MySQL volumes from Linux

Should data in MySQL out live its business purpose and retention period, it could be considered a candidate for cleanup. If the underlying volume only contained this data, it too could be considered a candidate for cleanup. However, removing the underlying volume from Linux OS is a little more involved. To remove the associated LUNs that were used to store the MySQL data, Dell EMC recommends using the following steps to remove a MySQL volume from Linux:

1. In MySQL, truncate the tables that reside in the LUN, then stop all application access to the volume.
2. Make sure no pids are active on the mount point. If there are, stop the pids.

    ```
    fuser /<MySQL vol>
    ```

3. Unmount the volume.

    ```
    umount /<MySQL vol> –l
    ```

4. Remove the volume from /etc/fstab.

    ```
    vi /etc/fstab
    ```

5. Get a listing of the device as reported by multipath.

    ```
    multipath –l
    ```

6. Remove the device from /etc/multipath.conf and from multipath.

    ```
    vi /etc/multipath.conf
    service multipathd reload
    multipath -f ${MPIO_ALIAS of MySQL vol}
    ```

7. Remove the device from the binding and wwids files.

    ```
    (echo "/^${MPIO_ALIAS of MySQL vol} /d
    wq") | ex -s /etc/multipath/bindings
    (echo "/^${WWID of MySQL vol} /d
    wq") | ex -s /etc/multipath/wwids
    ```

8. Commit to disk all data in the kernel filesystem buffers.

    ```
    sync
    ```

9. Free pagecache, dentries, inodes.

    ```
    echo 3 > /proc/sys/vm/drop_caches
    ```

10. Call block device and flush buffers for all multipath block devices for the volume.

    ```
    blockdev --flushbufs /dev/<device>
    ```

11. Flush the on-drive write-cache buffer for all multipath block devices for the volume.

    ```
    hdparm -F /dev/<multipath sd[<device>] device>
    ```

12. Remove the SCSI device from the system for all multipath block devices for the volume.

    ```
    echo 1 > /sys/block/<device>/device/delete
    ```

13. Unmap and delete the volume from the Dell Storage Center.

Dell EMC strongly recommends taking a cold backup of the entire database prior to and after the any cleanup operation, and then ensuring only reads be executed on the database during the cleanup operation. For more information on volume expansion, refer to the *Dell EMC SC Series with Red Hat Enterprise Linux 6x* best practices document.

4.9 MySQL data directories and SC Series arrays

As mentioned previously, Dell EMC recommends using at least one dedicated (non-boot) volume per MySQL instance, per logical function as needed, and if multiple MySQL instances are required on a server, each MySQL instance needs to use its own data directory. This provides the ability to create write-consistent snapshots of each set of databases managed by a specific MySQL instance, and provides greater flexibility in defining backup and recovery requirements and replay snapshot profiles.

MySQL data directory locations are defined by the datadir configuration option in **/etc/my.cnf** or by the **--datadir=<directory>** option when calling a MySQL instances. By default, it is set **to /var/lib/mysql**.

Several methods exist for moving or copying an existing datadir location, and creating a new MySQL data directory. Some of the methods and how they apply to SC Series arrays are discussed in the following sections.

4.9.1 Move MySQL data directory to a dedicated LUN

Dell EMC recommends moving the default data directory to a dedicated ext4 files system mounted with the discard option to enable thin space reclamation in SC Series arrays. This provides several benefits:

- Reclaims space as it is released from a MySQL database and returns it to the SC Series pagepool
- Provides fine-grained snapshot content — only MySQL database content is snapped
- Reduces the complexity of restoring or recovering a database in a snapshot of the boot LUN
- Provides greater flexibility with snapshot profiles and content and easily provides a way to adhere to specific backup and retention requirements for only the data that needs to meet the requirements and timelines
- Contains information in the snapshot only pertaining to MySQL and not the OS
- Expands the available options for backup and recovery strategies, including a disaster recovery scenario, by mounting the binary logging volume to a new master MySQL host if replication was behind at the time of the disaster
- Expands the options to isolate multiple databases on the same server when running multiple instances of MySQL

The following example shows how to move the default data directory (/var/lib/mysql) on a server running a single instance of MySQL to a dedicated SC Series volume. The example assumes the LUN is ext4 formatted and mounted with the discard option and SELinux is enabled with default policies for MySQL. See sections, 4.6 and 4.10 for more information. The following example needs to be adjusted accordingly for any system, and it assumes semanage is installed.

1. Shut down all database applications, and use Linux user root to stop the MySQL service.

```
# service mysqld stop
Stopping mysqld:                                          [  OK  ]
```

2. Create an SC Series snapshot of the entire MySQL environment.

3. Create a new volume in the SC Series array for the new data directory and present it to the server.
4. Scan Linux for the new device, set up multipath, and mount the device.
5. Verify that the dedicated LUN for MySQL data is mounted.

```
# df -k
Filesystem                     1K-blocks     Used Available Use% Mounted on
<snippet>
/dev/mapper/mysql-inst1        206293688    60684 195747244   1% /mysql1
<snippet>
```

6. Copy or move the original data files to the new location.

```
# cp -rap /var/lib/mysql/* /mysql1
```

7. Change the ownership and privileges of the new mount point to the user running the MySQL service.

```
# chown mysql:mysql /mysql1
```

8. Change the MySQL configuration file to use the new location. Create a backup of the original MySQL option file and modify it so that any pathnames referring to the original data directory refer to the new data directory and new socket file. Also, modify any other options that must be unique per instance, such as the TCP/IP port number and the log files.

```
# cp -p /etc/my.cnf /etc/my.cnf.`date +%Y%m%d_%H%M`
# vi /etc/my.cnf
```

Replace datadir and socket values. For example:

Old values:

```
datadir=/var/lib/mysql
socket=/var/lib/mysql/mysql.sock
```

New values:

```
datadir=/mysql1
socket=/mysql1/mysql.sock
```

Add the following to the end of the file:

```
[client]
socket=/mysql1/mysql.sock
```

9. Set the access control security policies in SELinux for the new location and socket file (see section MySQL and SELinux for instructions).
10. Restart MySQL and associated applications.

4.10   MySQL and SELinux

SELinux (Security-Enhanced Linux) provides access control security policies and governs resources from accessing objects they should not. By default, in Linux 6 when installing MySQL, SELinux is configured with a set of policies that expects MySQL databases to reside in /var/lib/mysql. If MySQL attempted to write to databases in some other directory, SELinux would prevent the writes unless additional access control policies were added to SELinux to allow the access.

To circumnavigate the access control security policies, SELinux could be disabled, but that would remove all security policies that might have been required or intended to run on a server. For that reason, Dell EMC does not recommend disabling SELinux to simplify changing the default location of MySQL writes, or for configuring multiple MySQL services to run from the same server, unless business requirements do not need the level or protection provided by SELinux. The Dell EMC recommendation is to modify or create new SELinux access control security policies when changing the default MySQL data directory.

Several commands can be used to change SELinux policies: semanage, restorecon, and chcon. Each have their own purpose and may not be interchangeable. The following shows how semange was installed on the system for this paper:

```
# yum whatprovides /usr/sbin/semanage |grep SELinux |head -1
policycoreutils-python-2.0.83-19.18.el6.x86_64
# yum -y install policycoreutils-python
Loaded plugins: security
Setting up Install Process
Resolving Dependencies
--> Running transaction check
---> Package policycoreutils-python.x86_64 0:2.0.83-24.0.1.el6 will be installed
<snippet>
Complete!
```

The following example shows the required changes to SELinux when changing the default data directory from /var/lib/mysql to /mysql1. The process would have to be modified if multiple MySQL data locations were introduced, or if multiple MySQL services were going to run from the same server — both of which are out of scope of this document.

1. In a default installation, MySQL runs with SELinux, so verify that SELinux is running. If it is not running, start it.

   ```
   # getenforce
   Enforcing
   ```

2. Change the ownership of the mount point so that MySQL has access to it.

   ```
   # chown mysql:mysql /mysql1
   ```

3. Get the current SELinux context for the existing storage location and socket file. This is done because if additional security contexts were added and needed, they can be carried over to the new location and socket file.

   ```
   # semanage fcontext -l | grep '\/var\/lib\/mysq*'
   /var/lib/mysql(/.*)?      all files  system_u:object_r:mysqld_db_t:s0
   /var/lib/mysql/mysql\.sock socket    system_u:object_r:mysqld_var_run_t:s0
   ```

4. Add the corresponding SELinux context mappings for the new location and socket file.

```
semanage fcontext -a -t mysqld_db_t "/mysql1(/.*)?"
semanage fcontext -a -t mysqld_var_run_t -f -s "/mysql1/mysql\.sock"
```

5. Verify the new context mappings exist.

```
# semanage fcontext -l | grep '^\/mysql1'
/mysql1(/.*)?              all files system_u:object_r:mysqld_db_t:s0
/mysql1/mysql\.sock       socket     system_u:object_r:mysqld_var_run_t:s0
```

6. Apply the new SELinux context to the running system and make the changes persistent.

```
# restorecon -Rv /mysql1
restorecon reset /mysql context system_u:object_r:file_t:s0-
>system_u:object_r:mysqld_db_t:s0
<snippet>
```

7. In the event that datadir is the root directory of a dedicated file system, datadir will contain the directory lost+found. Prior to MySQL 5.5.28, this will cause a warning. Starting with 5.5.28, the warning can be mitigated on MySQL initialization by ignoring any content residing in datadir. Using the root user, start the MySQL service and ignore directory in datadir that existed prior to the move. (See section 4.5 for more information).

```
# service mysqld start --ignore-db-dir=lost+found
```

Note: See section 4.5 for additional information on other required changes to MySQL that are needed to use the new mount point.

5 MySQL backups and SC Series snapshots

Using SC Series snapshots (replays) as backups can be an efficient backup strategy in terms of performance, effort, and resources required to create the backup and meet the RTOs. This is in contrast to using mysqldump to create backups which will take up the entire space of the database on disk for each backup. Incremental MySQL backup tools can be very space efficient too, however they can be complicated to use and can be resource intensive to maintain in any environment.

One important concept of using SC Series snapshots with MySQL as part of a backup strategy is the use of consistency groups. A consistency group is a set of one or more volumes that are logically related. When snapshots are taken using a consistency group, the snapshot for each volume in the consistency group is made at exactly the same time. Consistency groups are defined by creating a consistent snapshot profile (see section, Snapshot profile). When snapshots are combined with MySQL binary transaction logs for point-in-time recovery and mysqldump for long-term disaster protection, a schedule can be created that most efficiently meets requirements.

The following subsections concentrate on using SC snapshots for backups. Refer to appendix A for additional information on MySQL backups and recovery and MySQL Enterprise Backup.

5.1 Full and incremental (point-in-time) backups

There are several different types of backups that can be taken of a MySQL environment, and SC Series consistent snapshots can be used for either backup type:

Full backup of a MySQL environment includes all the databases which are under the control of a specific MySQL instance.

Incremental backup contains only data changes made from one point-in-time to another point-in-time. These changes are recorded in MySQL binary transaction logs.

5.2 MySQL binary transaction logs

Database changes made from one point in time to another are recorded in binary transaction logs only if MySQL is started with option --log-bin=<value>. The default location of binary logs is the data directory, but it can be changed. When using SC Series snapshots as backups of binary transaction logs, ensure a consistent snapshot profile includes the volumes used for datadir and log-bin. This guarantees a consistent image of the databases as well as any binary transaction logs.

```
# cat /etc/my.cnf
<snippit>
[mysqld]
datadir=/mysql1
log-bin=server-bin
server-id=0
<snippit>
# ls -lr /mysql1/server-bin*
-rw-r-----. 1 mysql mysql    60 Mar 31 10:14 /mysql1/server-bin.index
-rw-r-----. 1 mysql mysql 36166 Mar 31 11:06 /mysql1/server-bin.000003
-rw-r-----. 1 mysql mysql   177 Mar 31 10:13 /mysql1/server-bin.000002
-rw-r-----. 1 mysql mysql   177 Mar 31 10:07 /mysql1/server-bin.000001
```

MySQL command **show binary logs**; displays all binary log files, and **show master status**; displays the name of the current binary log.

```
mysql> show binary logs;
+-------------------+-----------+
| Log_name          | File_size |
+-------------------+-----------+
| server-bin.000001 |       177 |
| server-bin.000002 |       177 |
| server-bin.000003 |     36166 |
+-------------------+-----------+
3 rows in set (0.00 sec)

mysql> show master status;
+-------------------+----------+--------------+------------------+-------------------
-----+
| File              | Position | Binlog_Do_DB | Binlog_Ignore_DB |
Executed_Gtid_Set |
+-------------------+----------+--------------+------------------+-------------------
-----+
| server-bin.000003 |    36166 |              |                  |
|
+-------------------+----------+--------------+------------------+-------------------
-----+
1 row in set (0.00 sec)
```

The utility **mysqlbinlog** converts the contents of binary logs from binary to text, and it has options for processing specific sections of the logs based on times or events within the log.

```
# mysqlbinlog server-bin.000003 | more
```

MySQL binary logs can also assist in determining if certain tables are candidates for dedicated volumes pinned to storage profiles: **Medium Priority (Tier 2), Low Priority (Tier 3)**, **Low Priority with Progression (Tier 3 to Tier 2)**, or a custom profile to a lower tier of storage. In verbose mode, **mysqlbinlog** displays the actual DML statements that were executed. A shell script can parse the information and display the total number and type of DML operations against a table. If a table has a low number of DML operations over an extended period of time, it might be a candidate to be pinned to a lower tier of storage, especially if writes to RAID 10/RAIN 10 DM and automatic data tiering with data progression are not necessary.

```
-bash-4.1$ ksh ./num_DML_ops.sh
/mysql1/server-bin.000001
/mysql1/server-bin.000002
/mysql1/server-bin.000003
    101 ### insert into mysql.emp
      2 ### update mysql.emp
      1 ### delete from mysql.emp
-bash-4.1$
```

5.3   Cold backups with SC Series snapshots

The most complete, reliable, and consistent backup is a full, cold database backup of all the MySQL databases. This type of backup may not be appropriate for all implementations because it requires all database activity and the mysql service to be stopped, but it is by far the simplest to implement and guarantees consistent backups. With a minimal volume configuration (see Table 16), this backup could be created with a consistent snapshot profile that included volume /mysql1. Since mysql needs to be down during the creation of the snapshot, it is recommended that the snapshot profile be set for manual snapshots. Since full, cold backups could be disruptive to the environment, one solution is to create the snapshot from a replicated slave server that could be taken offline without harming application availability.

5.4   Hot and warm MySQL backups

Hot and warm backups are methods for backing up a MySQL database while applications are reading and writing information in the database. These backups are much less intrusive to clients than full, cold backups. However, care must be exercised to enforce suitable locking to guarantee data changes do not compromise backup integrity. For this reason, hot backups are typically targeted for large, mature, highly-available systems where a complete outage on the environment is prohibited.

Hot backups are taken without placing the tables in read-only mode. Any data that was inserted or changed while the backup was being made, must be included in the backup, and any data that was deleted, must be excluded from the backup. Uncommitted transactions must also be excluded from the backup.

Warm backups restrict database write operations while the backup is made.

5.5   Creating full backups with SC Series snapshots

To create full backups of MySQL databases by using SC snapshot technology, execute the MySQL command FLUSH TABLES WITH READ LOCK prior to creating the snapshot.

FLUSH TABLES WITH READ LOCK obtains a lock across all databases, which restricts incoming writes. Once the lock is in place, it flushes active cache to disk, assuring on-disk integrity.

A basic backup procedure would look similar to the following:

1. FLUSH TABLES WITH READ LOCK
2. FLUSH LOGS
3. Create an SC Series consistent snapshot using the correct consistent snapshot profile
4. UNLOCK TABLES

This method is intended to be used when manually creating backups, and not from a script. If backups need to be created from a script, the FLUSH TABLE command and snapshot creation must be executed from the same MySQL session. This is due to how MySQL manages locks created by FLUSH TABLE. For a script, the solution might look like the following (the example is one line of code, not two):

```
echo "FLUSH TABLES WITH READ LOCK; FLUSH LOGS; SYSTEM /mysqlbkp.sh; UNLOCK
TABLES;" | mysql -u <user> -p<password>
```

Where **mysqlbkp.sh** is a user-created script that makes the appropriate CLI call to the SC Series array to create the consistent snapshot. For information on the CLI interface (CompCU), see the *Storage Center Command Utility Reference Guide* on the Knowledge Center at the SC Series Customer Portal (login required).

The procedures described in this section can be applied to both MyISAM and InnoDB backups.

5.5.1 MyISAM

When creating a snapshot of a MyISAM database, amend the procedure in section 5.5 as necessary. For example, to get a consistent backup, acquire read locks and flush the tables, or stop the MySQL server. The flush is needed to ensure that all active index pages are written to disk. The read locks ensure that other clients can continue to read data while a snapshot is created.

5.5.2 InnoDB

When creating a snapshot of a MyISAM database, amend the procedure in section 5.5 as necessary. Also, to guarantee ACID compliance of all transactions, MySQL option innodb_flush_log_at_trx_commit=1 must declared. It instructs MySQL to manage its transactional commits to stable storage in an ACID compliant way.

Note: FLUSH TABLES WITH READ LOCK with InnoDB will not lock all internal data tables like it will with MyISAM. When using FLUSH TABLES WITH READ LOCK with InnoDB tables, it is solely intended to be a tool to settle I/O.

5.6 Creating incremental backups with SC Series snapshots

Ensure that the MySQL service is started with option **--log-bin**, and create an initial full backup of the environment (or the last incremental backup). Then when the incremental backup needs to be created, rotate the binary logs using MySQL command FLUSH LOGS. After flushing the logs, a snapshot of the binary log can be created. If binary transaction logs have been enabled, then the binary logs should also be rotated immediately prior to the creating full backups.

To create backups of MySQL databases using SC Series snapshot technology, execute the MySQL command **FLUSH TABLES WITH READ LOCK** prior to creating the snapshot.

A Additional resources

A.1 Technical support and resources

Dell.com/support is focused on meeting customer needs with proven services and support.

Storage Solutions Technical Documents provide expertise that helps to ensure customer success on Dell EMC storage platforms.

A.2 Referenced or recommended publications

Referenced or recommended Dell publications:

- Dell EMC SC Series with Red Hat Enterprise Linux 6x

Referenced or recommended Red Hat publications:

- Red Hat Enterprise Linux 6 DM Multipath, DM Multipath Configuration and Administration
- SELinux Contexts - Labeling Files
- Persistent Changes: semange fcontxt
- The file_t and default_t Types

Referenced or recommended MySQL publications:

- MySQL 5.7 Reference Manual
- mysqldump – A Database Backup Program
- MySQL Enterprise Backup Overview
- MySQL Enterprise Backup User's Guide (version 4.0.0)
- SELinux and MySQL

Other recommended publications:

- Standard RAID levels