



NETAPP TECHNICAL REPORT

# MySQL Backup and Restore Using Zmanda Recovery Manager and NetApp Snapshot Technology

Karthikeyan Nagalingam, Product and Partner Engineer, NetApp

K. K. George, Vice President, India Development Operations, Zmanda, Inc.

March 2008 | TR-3656

## TABLE OF CONTENTS

<b>1</b>	<b>INTRODUCTION</b>	<b>3</b>
1.1	INTENDED AUDIENCE	3
1.2	PURPOSE	3
<b>2</b>	<b>BACKUP AND RECOVERY SOLUTION FOR MYSQL OVERVIEW</b>	<b>3</b>
2.1	CONSIDERATIONS AND BEST PRACTICES FOR MYSQL BACKUP SOLUTION	5
2.2	ZMANDA RECOVERY MANAGER FOR MYSQL	6
2.3	ZMANDA MANAGEMENT CONSOLE (ZMC) FOR MYSQL BACKUP	8
2.4	ZMANDA RECOVERY MANAGER FOR MYSQL CLUSTER EDITION	9
2.5	USING SNAPSHOT TECHNOLOGY FOR BACKUP	9
2.6	ZRM WITH NETAPP PLUG-IN	10
2.7	PREREQUISITES FOR NETAPP SNAPSHOT COPIES	11
2.8	NETWORK SETTINGS	12
2.9	ETHERNET—GIGABIT ETHERNET, AUTONEGOTIATION, AND FULL DUPLEX	12
2.10	NETAPP SOFTWARE PRODUCTS USED IN THIS SOLUTION	13
<b>3</b>	<b>HOT AND WARM BACKUP</b>	<b>13</b>
<b>4</b>	<b>BACKUP LEVELS</b>	<b>13</b>
<b>5</b>	<b>DATABASE ENGINES</b>	<b>14</b>
<b>6</b>	<b>BACKUP METHODS USED BY ZRM</b>	<b>14</b>
<b>7</b>	<b>ONLINE MYSQL BACKUP: ZRM AND NETAPP SNAPSHOT TECHNOLOGY</b>	<b>15</b>
7.1	SOLUTION OVERVIEW	15
7.2	SETUP OVERVIEW	15
<b>8</b>	<b>APPENDIX: BACKUP AND RESTORE SETUP—ZRM WITH NETAPP SNAPSHOT TECHNOLOGY</b>	<b>16</b>
8.1	BACKUP	16
8.2	SUMMARY REPORT OVERVIEW	25
8.3	RESTORE	26
8.4	COMMANDS AND FILES USED FOR BACKUP	27
<b>9</b>	<b>REFERENCES</b>	<b>29</b>
<b>10</b>	<b>ACKNOWLEDGEMENTS</b>	<b>29</b>
<b>11</b>	<b>DISCLAIMER</b>	<b>29</b>

# 1 INTRODUCTION

Today, organizations have operations across the globe and they are required to keep their mission-critical applications running throughout the day. They also expect application performance to be maintained during routine maintenance such as backup and recovery, regardless of the data growth rate, which can sometimes be very high. Since backup windows are shrinking and the amount of the data that needs to be backed up is ever increasing, it is a complex task to define a point in time at which a backup can be taken with minimal impact on system performance and availability.

The MySQL™ database is the world's most popular open source database. Many data centers deploy MySQL because of its low database licensing cost compared to that of other databases. Zmanda and NetApp have worked together to deliver a backup and recovery solution for MySQL users through Zmanda Recovery Manager (ZRM) and NetApp® Snapshot™ technology.

Zmanda Recovery Manager is a comprehensive backup and recovery product for MySQL that works with every storage engine and supports local and remote backup of MySQL. ZRM is certified by MySQL and NetApp. For hot and scalable backup of MySQL, ZRM has a plug-in that enables backup using NetApp Snapshot technology. Zmanda provides two editions for the MySQL database backup:

- Zmanda Recovery Manager for MySQL—Enterprise Edition
- Zmanda Recovery Manager for MySQL—Cluster Edition

## 1.1 INTENDED AUDIENCE

This technical report is intended for MySQL database administrators (DBAs) and storage administrators, and for storage architects who are familiar with MySQL, Snapshot technology, SnapMirror®, and volume clones. We assume that you already acquired a copy of ZRM for MySQL and NetApp Snapshot option from Zmanda

## 1.2 PURPOSE

This document describes how NetApp Snapshot and Zmanda Recovery Manager can be used to back up and restore a MySQL database for NetApp storage systems. Specifically, this report covers the following topics:

- Infrastructure required integrating Zmanda Recovery Manager with a NetApp storage system
- Backing up a MySQL database using Zmanda Recovery Manager using a NetApp Snapshot plug-in
- Restoring a MySQL database using Zmanda Recovery Manager

# 2 BACKUP AND RECOVERY SOLUTION FOR MYSQL OVERVIEW

The MySQL database server has been the heart of data-driven applications in the open source environment that serves a growing and intensely demanding customer base. The “M” in the SAMP/LAMP/WAMP stack (Solaris®/Linux®/Windows®, Apache™, MySQL, and PHP/Perl/Python®), MySQL has been tested by heavy transaction processing applications, terabyte-sized data warehouses, as well as high-traffic Web sites and a proven open source database technology. With its fast adoption rate, MySQL has evolved to become an enterprise-class database complete with must-have features and a supporting system of must-have services and production support that successful organizations demand. MySQL Enterprise is the solution provided by MySQL AB for these data centers.

Traditional MySQL backup and recovery methods present the following challenges:

- **System performance impact:** MySQL database backup operations typically significantly impact the performance of a production system because they place a high load on the database server, the storage system, and the underlying network during the backup process.
- **Shrinking backup windows:** Enterprises expect maximum availability of their applications. Defining an appropriate window for creating backup images can present a challenge when the database needs to be accessible all day.

- **Rapid data growth:** Enterprise data is growing exponentially. That means longer windows are needed to back up the data. Organizations that can no longer afford an extended window for backups are forced to heavily invest in the backup infrastructure to keep the backup window short. Growing databases also require more tape media or disk storage space for backup images. Incremental backups can address these issues, but longer restore times make them unacceptable.
- **Decreasing mean time to recover (MTTR):** The mean time to recover is the time needed to recover from a database failure. The MTTR can be divided into two parts: the time that is necessary to restore the database from a backup image and the time that is necessary to perform roll-forward recovery of the database. The roll-forward recovery time depends on the number of archives and active logs that need to be reapplied to the database after it has been restored.

NetApp and Zmanda provide unique functionalities that address these challenges.

NetApp Snapshot technology enables DBAs to create point-in-time copies of file systems, which can be used to protect data—from a single file to a complete disaster recovery solution. Snapshot technology offers the following key features:

- **Stability:** A NetApp Snapshot copy is a read-only, completely static, and incorruptible copy. As such, it enables organizations to perform consistent backups from a NetApp storage system while applications are running.
- **Performance:** Storing a NetApp Snapshot copy on a NetApp system has no performance impact. In addition, creating and deleting a Snapshot copy has virtually no performance impact.
- **Minimum storage requirement:** Two Snapshot copies taken in sequence differ from one another by the blocks added or changed in the time interval between when the two were taken. This block-incremental behavior minimizes the amount of storage space consumed.
- **Fast backup and recovery:** A Snapshot copy of a database can be created in a matter of seconds, regardless of the size of the database or the level of activity on the NetApp storage system. This dramatically reduces the database backup window from hours to seconds and helps database administrators to schedule frequent low-impact database backups. Using the Data ONTAP® SnapRestore® command, an entire database can be restored in a matter of seconds from a Snapshot backup. Because no data copying is involved, an incredible amount of time is saved when a database is returned to the state it was in at the time the Snapshot copy was created. Additionally, because Snapshot copies can be taken quickly and a large number of Snapshot copies can be retained, the amount of time needed to perform a roll-forward recovery operation against a database can be greatly reduced.
- **High reliability:** The RAID architecture used for NetApp storage systems is unique and provides greater reliability than direct-attached storage. If a RAID member disk fails, it is automatically reconstructed (using parity disk data) without any user intervention. NetApp supports single parity as well as double parity RAID. NetApp's double parity RAID, known as RAID-DP™, is considered approximately 10,000 times more reliable than traditional RAID. For more details on RAID-DP, refer to the technical report "Double Parity RAID for Enhanced Data Protection with RAID-DP" (<http://www.netapp.com/library/tr/3298.pdf>) located on the NetApp Web site (<http://www.netapp.com>).

Zmanda Recovery Manager for MySQL offers an easy-to-use yet flexible and robust backup and recovery solution for MySQL servers for database administrators. ZRM is certified by MySQL and NetApp. ZRM has a plug-in that enables backup using NetApp Snapshot technology for hot and scalable backup of MySQL. ZRM for MySQL allows database administrators to:

- Schedule full and incremental backups of a MySQL database.
- Start immediate backup or postpone scheduled backups based on thresholds defined.
- Choose to do more flexible logical or faster raw backups of their database.
- Perform backup that is the best match for their storage engine and their MySQL configuration.
- Backup their remote MySQL database through a firewall.
- Configure on-the-fly compression and/or encryption of MySQL backups to meet storage and security needs.
- Get e-mail notification of the status of backups and receive MySQL backup reports via RSS feed.
- Monitor and browse backups.
- Define retention policies and delete backups that have expired.
- Recover a database easily to any point in time or to any particular transaction, for example, just before a user made an error.

- Parse binary logs to search and filter MySQL logs for operational and security reasons.
- Take Snapshot copies of live MySQL with Linux LVM, Windows VSS, Solaris ZFS, NetApp SnapManager®, and Veritas™ VxFS to minimize locking on the database. This means that DBAs can back up mission-critical MySQL production environments without interrupting online data access.
- Easily manage MySQL databases from a single Web-based Zmanda Management Console.

## 2.1 CONSIDERATIONS AND BEST PRACTICES FOR MYSQL BACKUP SOLUTION

### CONSIDERATIONS

- How fast and how easy do you want the MySQL recovery process to be?
- What will be the impact of the MySQL backup process on your application?
- What will your backup configuration look like? This is the “what, where, when, and how” of MySQL backup.
- How will you manage your backup process and backed up data?
- What kind of tracking, reporting, and compliance requirements does your business have for your MySQL backup implementation?

### BEST PRACTICE

- **Cross engine and platform support:** MySQL backups should work with every MySQL storage engine, including MyISAM, InnoDB, NDB, and others. The solution should also back up multiple versions of MySQL running on Linux, Windows, Solaris, Mac® OS X, and other UNIX® platforms. This enables MySQL DBAs to use the optimal storage engine and platform for each application, without worrying about implication on backup.
- **Scale up and out:** Your backup solution has to scale up and be able to back up terabytes of data stored in a single database. It should also scale out from backup of a single MySQL server to a backup of multiple MySQL servers that could be geographically dispersed and quite large.
- **Reduced or zero application downtime:** To accommodate data growth the backup solution should scale up with the size of the database without increasing lock times, thus minimizing the impact on users and applications. A backup window is sometimes defined as the amount of time it takes to complete a backup. A better definition for a backup window is the amount of time that the application is unavailable or degraded during backup. This is an important distinction, because some backup scenarios take the application offline for just a few moments, but can then spend hours moving the resultant backup to media. One such example is using Snapshot technologies combined with traditional backup.
- **Relational integrity protection:** A successful backup solution maintains the referential integrity of the database; any databases and tables that reference each other must be backed up together to ensure a consistent state after restoration.
- **Location of backup data:** Storing backup data on the same disk or server as the original data is risky. Backup data should optimally reside in a physically separate location, preferably many miles from the production data.
- **Recovery objective requirements:** Many DBAs focus too much on optimizing the backup process, thus ignoring an essential feature—the ability to recover data as quickly as possible after various mishaps. The recovery time objective (RTO) is the amount of time elapsed between the disaster and restoration of business functions. You should carefully consider how long your MySQL applications can be offline when designing your backup and recovery solution. For applications that cannot afford to be down for very long, you should consider MySQL replication with backup. The recovery point objective (RPO) is the time (prior to the disaster or error) to which you plan to recover your data. The most desirable RPO allows restoring transactions to the point just prior to the mishap. ZRM for MySQL can recover data to any point in time between two successful backups.
- **Ease of recovery:** Real-life recoveries are done under stressful conditions. That is why simplified and bullet-proof recovery procedures make a big difference when you are the one responsible for recovery.
- **Scheduling and automation:** A backup solution should provide flexible, rule-based scheduling. A key feature is the ability to postpone and cancel backups based on administrator-defined

thresholds. For example, you might want to postpone backup runs when the MySQL database is under heavy load. Automation is also essential. Relying on human intervention to perform backups usually results in inconsistent execution. You should design your solution to work with as little human intervention as possible. Automated backups must be monitored, preferably via customizable reports and automatic notifications. Backups are not very useful if you can't find the desired data quickly. At a minimum, your backup and recovery solution should list all backups performed. This list should contain what was backed up, when it was backed up, and the location of the backup media.

- **Backup security:** Access to backups must be secure. In addition to allowing encryption, the backup/recovery solution should control access to the tool itself, for example, allow only authorized personnel to perform recovery.

## 2.2 ZMANDA RECOVERY MANAGER FOR MYSQL

The Zmanda Recovery Manager for MySQL (ZRM for MySQL) provides an intelligent and easy-to-configure framework for backing up and restoring MySQL databases.

### ZRM ARCHITECTURE

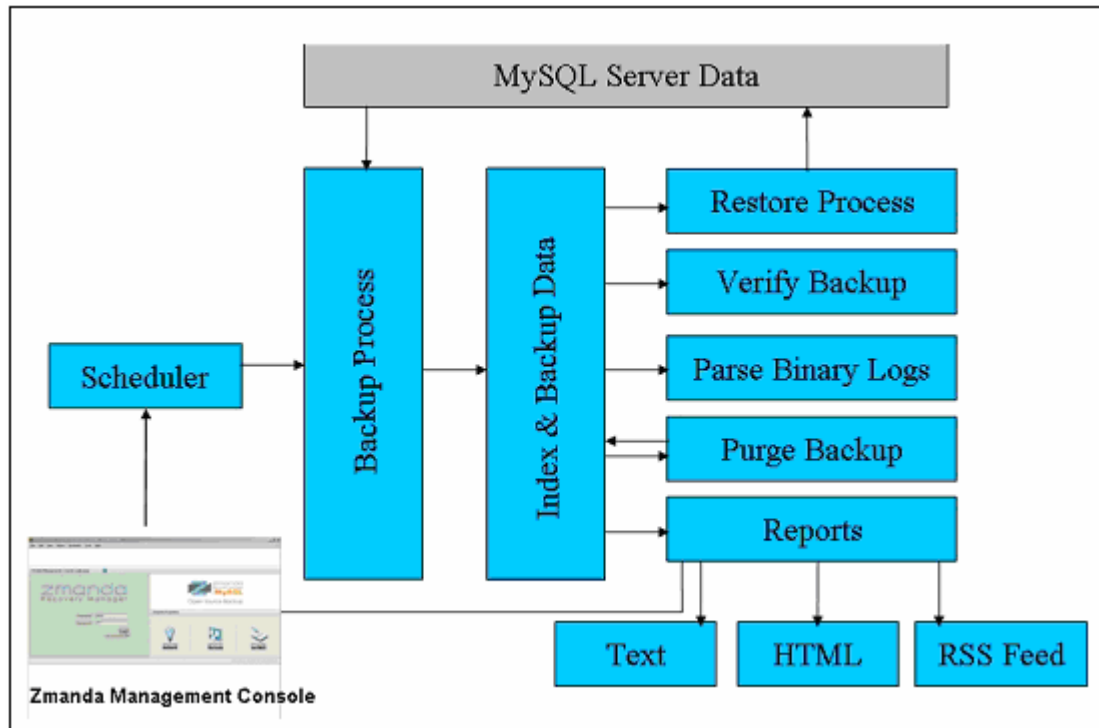


Figure 1) ZRM for MySQL architecture.

[Figure 1](#) depicts the ZRM architecture, which is modular and pluggable, which simplifies customization and addition of new features.

The ZRM scheduler starts all backup processes. Depending on the MySQL server, storage engine, and underlying storage technology (for example, whether Snapshot copies are available), ZRM intelligently decides what kind of backup to perform. The primary consideration is impact on the database. ZRM bases its scheduling decisions on the following criteria:

- Are Snapshot copies possible?
- Is a raw backup possible?
- Is replication available?
- Does the backup set include a MySQL cluster database?

If other options are not feasible, or, for example, if Snapshot fails because there is not enough space on the reserved area, the ZRM performs a logical backup.

You can always override the ZRM recommendations. For example, instead of using a Snapshot copy you can force ZRM to perform a logical backup.

ZRM will index all backups to track the metadata required for a successful restore. Everyday, ZRM checks for backups that are older than the retention policy (the number of days or months that backups should be kept). The expired backups are purged from the ZRM backup index and removed from the disk. A reporting engine provides customizable reports in either text or HTML format. Reports can be distributed by e-mail or by RSS feed.

### **OPTIMIZE BACKUP AND RESTORE CAPABILITIES**

ZRM for MySQL optimizes backup and restore operations on MySQL databases. It provides full flexibility to individually leverage native MySQL/OS backup tools, levels, scheduling, and so on. It generates logs that assist in optimizing these capabilities. It also provides filters to easily locate anomalous database events.

#### **Backup Features**

- Creates backup of multiple MySQL databases managed by one or more MySQL servers.
- Creates backup of tables in a single database.
- Supports hot backup.
- Supports multiple backup methods, depending on the storage engine used by MySQL tables.
- Provides full and incremental database backups.
- Supports use of mysqldump, mysqlhohopy, all common Snapshot technologies, and MySQL replication to execute backups.
- Creates consistent backups of the database regardless of the storage engine used by database tables.
- Supports SSL authentication between the local ZRM for MySQL and the remote MySQL server to allow secure backups over the Internet and across firewalls.
- Verifies backed up data images.
- Backup images can be compressed as well as encrypted using standard tools such as gzip, GPG, and so on.
- Backup runs can be cancelled by users with administrative privileges.

#### **Recovery Features**

- Optionally maintains a backup index that stores information about each backup run.
- Includes a report utility that can be used to browse indexes.
- Supports recovery of full and incremental database backups.
- Performs selective, incremental restores based on binary log position or by a given point in time, thus protecting the database from operator errors.
- Binary log filtering helps you decide what to restore and what to discard.
- Depending on the type of backups selected, data can be recovered to the same machine or to a different machine.

#### **Reporting and Scheduling Capabilities**

- Can schedule backup runs immediately or in daily/weekly/monthly intervals.
- Automatically generates backup reports.
- Has predefined backup reports as well as user-defined backup reports.
- Reports any backup statistic or combination thereof.
- Can format reports as HTML or text.
- Sends e-mail notification about the backup run status.
- Provides backup reports as RSS feed.

## Plug-ins

ZRM for MySQL allows plug-in operations to extend ZRM for MySQL capabilities. Plug-ins optimize the backup process for your environment. The following plug-in operations are supported (for more information, refer to the Appendix under Section [Backup How](#)):

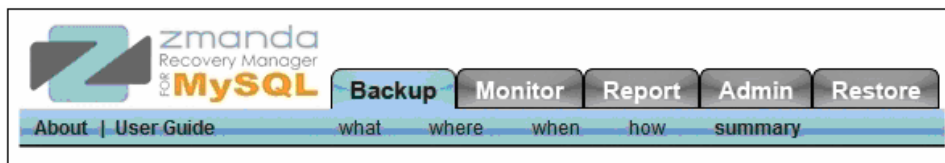
## Pre-scheduling

- Pre-backup
- Post-backup
- Copy plug-ins for Linux/UNIX and Windows
- Binary log parser plug-in
- Snapshot plug-in

## 2.3 ZMANDA MANAGEMENT CONSOLE (ZMC) FOR MYSQL BACKUP

In addition to its command line and configuration file interface, ZRM for MySQL also includes the Zmanda Management Console for MySQL. The ZMC is a Web-based management console that delivers all the automation, management, monitoring, and recovery capabilities of ZRM for MySQL in an intuitive graphical user interface.

ZMC is integrated with the Zmanda Network, which provides certified ZRM binaries, technical support, and a knowledge base to help you deploy your MySQL backup solution. ZMC provides context-sensitive help and the most current information about ZRM from the Zmanda Network.



Just as the MySQL Enterprise Monitor allows you to manage MySQL server operation, ZMC lets you manage backup and restore operations for MySQL.

Figure 2 shows the local server running ZMC backing up two MySQL with multiple MySQL databases. These are backed up by configuring at least two backup sets, that is, at least one backup set per MySQL server.

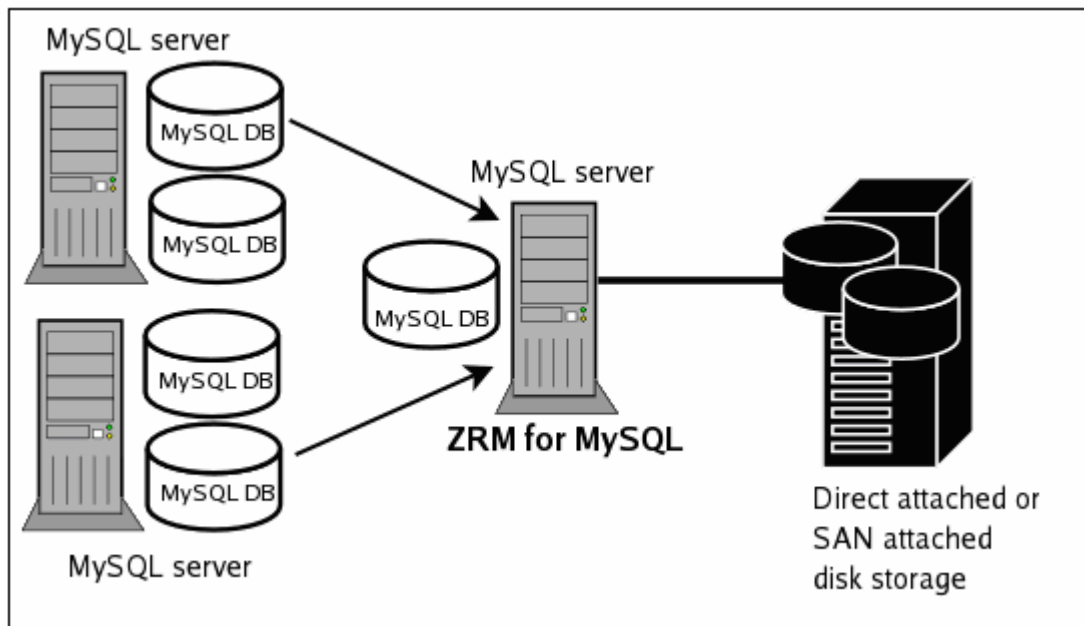


Figure 2) Zmanda Management Console.



## BACKUP SETS

Most ZMC operations work on backup sets. A backup set defines the "what," "where," "when," and "how" of data that is to be backed up. ZMC users can have two roles: administrator or operator. All backups and reports previously created using community versions of ZRM can be managed using ZMC.

## ZMC COMPONENTS

ZMC consists of the following components:

- Apache Web server
- MySQL database server
- Perl and Perl modules

All ZMC components are installed in a separate directory (under `/opt/zmanda/zrm`), and therefore do not impact existing installations of AMP stacks on the backup server. ZMC runs on the ZRM server and can be accessed from any Web browser. ZMC is supported on Internet Explorer 6 and 7 on Windows, and Firefox (1.5 or higher) on Windows and Linux.

## ZMC AND THE ZMANDA NETWORK

The ZMC is closely integrated with the Zmanda Network, which provides documentation as well as context-sensitive help on messages and error resolution. The Zmanda Network is continuously updated with the latest Zmanda information. This allows the ZMC to provide up-to-date suggestions and resolutions for specific issues faced by a backup administrator.

## 2.4 ZMANDA RECOVERY MANAGER FOR MYSQL CLUSTER EDITION

This document has been prepared for Zmanda Recovery Manager for MySQL Cluster Edition 2.1. ZRM for MySQL Cluster Edition provides all the functionality supported by ZRM for MySQL Enterprise Edition 2.1 in addition to support for MySQL clusters (NDB storage engine).

ZRM for MySQL Cluster Edition supports backup and recovery for MySQL Cluster version 5.0 and 5.1 (including MySQL Cluster 5.1 Carrier Grade Edition).

## 2.5 USING SNAPSHOT TECHNOLOGY FOR BACKUP

To perform backups with the least impact on database applications, consider backing up from Snapshot copies. A Snapshot copy provides a logical copy of a file system at a particular point in time. This logical copy can be used as a backup source, allowing recovery of the database to the point in time when the Snapshot copy was taken.

When correctly configured, Snapshot copies reduce the read lock time to a few seconds or even milliseconds. To ensure consistency of data for MyISAM and other storage engines, ZRM issues a read lock for a moment (in most cases a second or two), which is acceptable in most application scenarios. Note that the lock is enforced only during Snapshot copy creation; there is no lock during the subsequent data transfer when the Snapshot copy is mounted and read. You can continue to run queries on your tables even during the momentary lock time.

ZRM performs the following steps to ensure recoverable and consistent backups:

- Identify database volumes
- Lock databases
- Flush the memory buffers for logical consistency of data on disk (you can not take a Snapshot copy of memory—this technology works only for a disk)
- Create Snapshot copies
- Unlock databases
- Mount Snapshot copies
- Copy data from Snapshot copies
- Unmount Snapshot copies
- Delete Snapshot copies

ZRM for MySQL allows you to choose where to store backups; you can back up to any disk that is writeable by ZRM. For example, you can back up to a local disk, a disk array, or to a NAS device. You should never keep your backup data on the same spindles as the original data, not even for a short period of time unless you copy it onto another server or medium and keep a version of the backup on the same disk for faster recovery.

## 2.6 ZRM WITH NETAPP PLUG-IN

ZRM uses the NetApp Snapshot technology to take the point-in-time image of the underlying NFS exported volume, which is mounted from the NetApp storage system to store the MySQL databases, binary logs, and update logs. The point-in-time image (Snapshot copy) is stored in the same volume. Using Snapshot the ZRM will back up the MySQL database to the backup server.

NetApp storage systems include NetApp SnapManager software, which facilitates near-instantaneous hot backups and rapid restores.

ZRM for MySQL includes an optional Snapshot plug-in that integrates with SnapManager software to create consistent MySQL backups. It creates temporary Snapshot copies of the NetApp volumes to back up. The Snapshot copies are removed when the backup run is completed. ZRM for MySQL using SnapManager performs backups with minimal impact on MySQL applications.

This section describes the command-line configuration of NetApp Snapshot backups, including requirements for the MySQL database.

### MYSQL CONFIGURATION REQUIREMENTS

- Both ZRM and MySQL servers require configuration of `sudo` privileges (see Section [2.7](#)). On Solaris platforms, this indicates that the `SMC sudo` package must be installed.
- If backing up a remote MySQL server, the destination directory specified on the **Backup Where** page must exist on the MySQL server as well as the ZRM server. The MySQL backup user (OS level) must have read/write permission to access this directory.

All MySQL data and logs must reside on NetApp volumes. Some of the possible configurations are as follows:

- All MySQL data is on a single NetApp volume.
- Specific database directories are on different volumes.
- For databases containing InnoDB-based tables, the Snapshot copy can only be used if the database directory, the InnoDB data files, and the InnoDB logs are all on NetApp volumes.
  - The InnoDB shared data files are on a separate NetApp volume.
  - The InnoDB logs are on a separate NetApp volume.

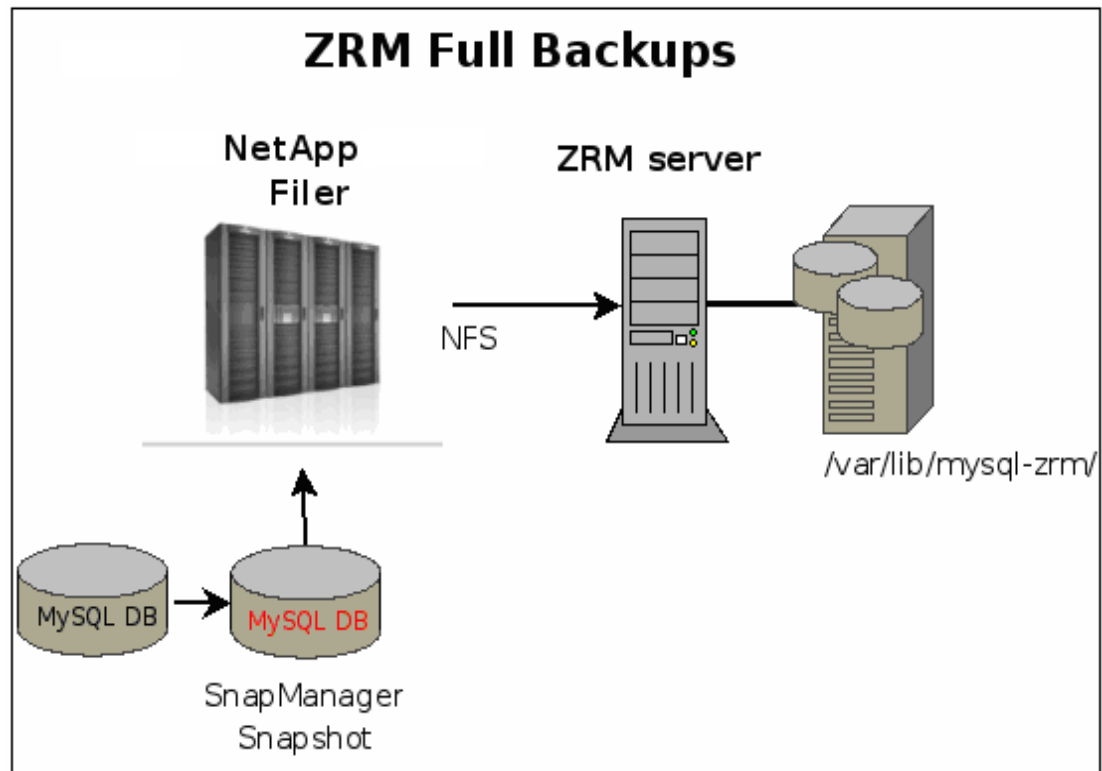


Figure 3) ZRM full backups.

The NetApp volumes are mounted on the ZRM server using NFS. ZRM user "mysql" should have permission to read and write to the volumes.

## 2.7 PREREQUISITES FOR NETAPP SNAPSHOT COPIES

- The MySQL backup user must be granted `sudo` privileges to execute NetApp commands on the MySQL server. To do this, do one of the following:

- Add a line similar to the following example to `/usr/local/etc/sudoers` on the MySQL server:

```
mysql
Zmanda.btcppe.netapp.com=NOPASSWD: /bin/mount, NOPASSWD: /bin/umount,
NOPASSWD: /bin/df
```

Where:

`ZRMserver.mycompany.com` is the fully qualified domain name for the ZRM server.

Note that if the `mount`, `umount` and `df` commands are installed in nonstandard locations, the above example would not work without editing it to reflect the different paths.

- Provide full access to the `mysql` user as below for testing purposes in `/etc/sudoers`. Then provide `df`, `umount`, and `mount` access to the `mysql` user.

```
%mysql ALL=(ALL) NOPASSWD: ALL
```

**Note:** If we use RHEL5 – U1 and above, ensure that the `Default requiretty` line is not present in the `/etc/sudoers` file.

- Enable the `ssh-key auth` between the MySQL server and the ZRM server for `mysql` user because the backup happens as `mysql` user.
- If ZRM is remote to the MySQL server, the `ssh-keys` should be shared between the ZRM and MySQL servers.
- If ZRM is remote to the MySQL server, the UID and GID of the `mysql` user should be the same on both the ZRM and MySQL machines.
- The snap reserve for the NetApp volume must be adequate to accommodate changed data. The NetApp default of 20% may be low or high for your site, depending on data change rates.
- To determine the snap reserve, use the NetApp `snap reserve volumename` command.
- All MySQL database files (data, log, indexes) must be stored in NetApp volumes to ensure consistency.
- If any of the files are not on NetApp volumes, a raw backup using `mysqlhotcopy` or a logical backup using `mysqldump` will be taken based on the storage engines of the tables in each of the databases.

## 2.8 NETWORK SETTINGS

When configuring network interfaces for new systems, it's best to run the `setup` command to automatically bring up the interfaces and update the `/etc/rc` file and `/etc/hosts` file. The `setup` command requires a reboot to take effect.

However, if a system is in production and cannot be rebooted, the network interfaces can be configured using the `ifconfig` command. If an NIC is currently online and needs to be reconfigured, it must first be brought down. To minimize downtime on that interface, a series of commands can be entered on a single command line separated by a semicolon (;).

For example:

```
filer>ifconfig e0 down;ifconfig e0 'hostname'-e0 mediatype auto netmask
255.255.255.0 partner e0
```

**Note:** When configuring or reconfiguring NICs or VIFs in a cluster, it is essential that you include the appropriate partner <interface> name or VIF name in the configuration of the cluster partner's NIC or VIF, to ensure fault tolerance in the event of cluster takeover. Consult your NetApp support representative for assistance. A NIC or VIF used by a database should not be reconfigured while the database is active. Doing so can result in a database crash.

## 2.9 ETHERNET—GIGABIT ETHERNET, AUTONEGOTIATION, AND FULL DUPLEX

Any database using NetApp storage should utilize Gigabit Ethernet on both the NetApp storage system and the database server.

NetApp Gigabit II, III, and IV cards are designed to autonegotiate interface configurations and are able to intelligently self-configure themselves if the autonegotiation process fails. For this reason, NetApp recommends that Gigabit Ethernet links on clients, switches, and NetApp systems be left in their default autonegotiation state. The links should be changed only if no link is established, performance is poor, or other conditions arise that might warrant further troubleshooting.

Flow control should, by default, be set to "full" on the appliance in the `/etc/rc` file by including the following entry (assuming the Ethernet interface is `e5`):

```
ifconfig e5 flowcontrol full
```

If the output of the `ifstat -a` command does not show full flow control, then you will also have to configure the switch port to support it. (The `ifconfig` command on the appliance will always show the requested setting; `ifstat` shows what flow control was actually negotiated with the switch.)

## 2.10 NETAPP SOFTWARE PRODUCTS USED IN THIS SOLUTION

### SNAPSHOT

NetApp Snapshot technology delivers more stability, scalability, recoverability, and performance than competing snapshot technologies. NetApp has leveraged its superior Snapshot technology as a foundation for developing a family of data protection solutions; these products incorporate and extend the advantages of Snapshot technology to deliver enterprise data protection.

A Snapshot copy is a locally retained point-in-time image of data. NetApp Snapshot technology is a feature of the WAFL® (Write Anywhere File Layout) storage virtualization technology that is part of Data ONTAP, the microkernel that ships with every NetApp storage system. A NetApp Snapshot copy is a "frozen," read-only view of a WAFL volume that provides easy access to old versions of files, directory hierarchies, and/or LUNs (logical unit numbers).

The high performance of NetApp Snapshot also makes it highly scalable. A NetApp Snapshot copy takes only a few seconds to create—typically less than one second, regardless of the size of the volume or the level of activity on the NetApp storage system. After a Snapshot copy has been created, changes to data objects are reflected in updates to the current version of the objects, as if Snapshot copies did not exist. Meanwhile, the Snapshot version of the data remains completely stable. A NetApp Snapshot copy incurs no performance overhead; users can comfortably store up to 255 Snapshot copies per WAFL volume, all of which are accessible as read-only and online versions of the data.

Snapshot technology creates read-only versions of a FlexVol® volume. Snapshot images are used for quick recovery of deleted or corrupted data. System administrators use Snapshot images to facilitate backups of files, directory hierarchies, LUNs, and/or application data.

System administrators use Snapshot copies to facilitate frequent, low-impact, user-recoverable backups of files, directory hierarchies, LUNs, and/or application data. Snapshot copies vastly improve the frequency and reliability of backups, since they incur minimal performance overhead and can be safely created on a running system.

Snapshot copies provide near-instantaneous, secure, user-managed restores. Users can directly access Snapshot copies to recover from accidental deletions, corruptions, or modifications of their data. Since the security of the file is retained in the Snapshot copy, the restoration is both secure and simple.

## 3 HOT AND WARM BACKUP

Both raw and logical backups provide warm backup, which indicates that you don't have to shut down MySQL server for backup, but all tables are locked for writes during backup. That is why you should consider using the ZRM scheduling plug-in that allows delaying backups based on thresholds you define. For example, you can postpone a backup for an hour if more than 50 users are accessing the database at that moment.

Hot backup means you can back up a database without any impact on users. ZRM provides hot backup by using Snapshot copies. Also, ZRM takes no lock on the database while performing incremental backups.

## 4 BACKUP LEVELS

ZRM supports two backup levels: full (level 0) and incremental (level 1).

A full backup is a complete copy of the specified databases or tables. The copy can be logical, raw, or Snapshot based.

An incremental backup stores data changed since the last (full or incremental) backup by leveraging MySQL binary logs. When ZRM for MySQL performs an incremental backup, it flushes and saves the MySQL binary logs. Because logs can be flushed and saved without a read lock, it is a "hot backup," having no impact whatsoever on the live database. The binary log saves all database events, thus providing a very fine granularity of recovery points. ZRM makes it easy to recover data from incremental backups even when multiple incremental restores are required to achieve the recovery point. Although incremental backups require that MySQL binary logs are enabled, in most cases this results in a performance hit of less than 1%.

## 5 DATABASE ENGINES

ZRM provides logical, raw, and Snapshot-based backup for all database engines including NDB clusters. Note that some engines do not support all backup methods:

Table 1) Database engines.

Storage Engine	Logical	Raw	Snapshot
MyISAM	Yes	Yes	Yes
InnoDB	Yes	No	Yes
NDB (Cluster)	No	Yes	No
Archive	Yes	Yes	Yes
Falcon	Yes	Yes	Yes

You can mix storage engines (for example, InnoDB and MyISAM tables) within a single backup set. In such cases, ZRM will account for all storage engines in the backup set before deciding on the optimal backup mechanism.

## 6 BACKUP METHODS USED BY ZRM

The table below summarizes the technologies that ZRM supports for different operating systems:

### MYSQL SERVER

Table 2) Backup.

Linux	Solaris	Windows	OS X and Other UNIX
Logical—warm Raw—warm LVM—hot VxFS—hot NetApp—hot	Logical—warm Raw—warm ZFS—hot VxFS—hot NetApp—hot	Logical—warm VSS—hot	Logical—warm

## 7 ONLINE MYSQL BACKUP: ZRM AND NETAPP SNAPSHOT TECHNOLOGY

### 7.1 SOLUTION OVERVIEW

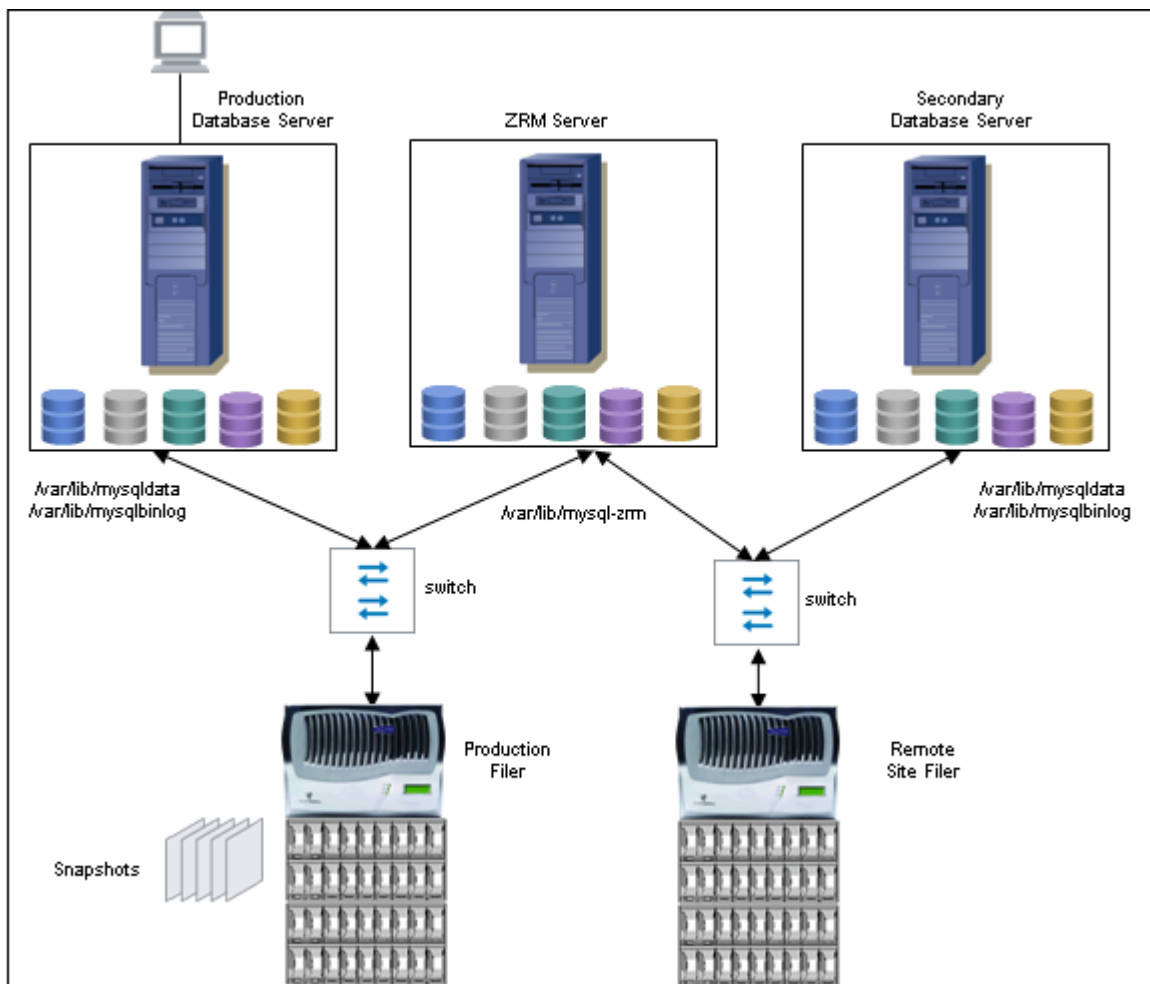


Figure 4) Online MySQL backup using ZRM and NetApp Snapshot technology—solution overview.

### 7.2 SETUP OVERVIEW

1. The primary database server responds to the clients to access data using client programs like PHP, Perl, and so on.
2. The primary database server accesses the two volumes from the primary NetApp storage system.
3. Each volume serves a separate task such as one for MySQL datadir (`/var/lib/mysqldata`), another one for MySQL binlog (`/var/lib/mysqlbinlog`).
4. Install the ZRM software in the ZRM server.
5. Create and configure a backup set for each MySQL database server in the ZRM server.
6. ZRM creates the point-in-time Snapshot copy for each volume using NetApp Data ONTAP API.

7. Mount the Snapshot copy on the ZRM server and copy the MySQL database files and folders to `/var/lib/mysql-zrm`, which is mounted from the remote site NetApp storage system or separate NetApp storage system; this will prevent data loss if the production or remote site system goes down.
8. The secondary database server needs to be up and running when the production database server does not respond to the client. For that, we need to restore the MySQL database files and folders from the particular backup set from the ZRM server. Then the client will continue to access the database through the secondary database server.

## 8 APPENDIX: BACKUP AND RESTORE SETUP—ZRM WITH NETAPP SNAPSHOT TECHNOLOGY

This section describes the detailed installation and configuration of the MySQL backup and restore procedure using ZRM with NetApp Snapshot technology.

### 8.1 BACKUP

#### ZMANDA CLIENT INSTALLATION

1. Download the **MySQL-zrm-enterprise-client-2.1-1.noarch.rpm** from <http://network.Zmanda.com/download.php#Zrm>. Install it in the MySQL server being backed up. This is essential for Snapshot-based backup.

2. Type:

```
rpm -i MySQL-zrm-enterprise-client-2.1-1.noarch.rpm [ for Linux]
```

#### ZRM COMPONENTS INSTALLATION

1. Download the ZRM enterprise edition from <http://network.Zmanda.com/download.php#Zrm>.
2. Log in as `root` user.

```
chmod +x ZRM-enterprise-2.1-installer.bin
```

3. Download and copy the ZRM license to `/etc/Zmanda/Zmanda_license`. This license file can be downloaded from <http://network.Zmanda.com/download.php#Zrm>.

```
./ZRM-enterprise-2.1-installer.bin
```

The rapid installer then starts. Follow the on-screen instructions.

#### MYSQL SERVER CONFIGURATION

The following configuration changes are required before you start the backup:

1. Check that the `datadir` parameter value points to the NetApp storage volume mounted folder.
2. Check that the binary log and update log are enabled.
3. Check MySQL basic functionality like login, create database/table, insert, delete, and update records.
4. Check the size of the database to back up.



## CREATING A NEW BACKUP SET

Create a new backup set

A backup set consists of the what, where, when and how you would like to backup your data. Backup servers can manage multiple backups sets and backup multiple MySQL servers, however, a backup set can backup only one MySQL server.

**Create New Backup Set**

Backup Set Owner: admin

Backup set name: NetAppMySQLZmanda

Comments: This backup set is used to Manage the backup for the MySQL server

Create Backup Set

Figure 5) Create a backup set.

To create a new backup set, from the **Create a new backup set** window, do the following:

1. From the **Backup Set Owner** drop-down list, select the ZRM for the MySQL user who will own the backup set. The only available user will be Admin until you configure more users.
2. Under **Backup set name**, type a unique and descriptive name for the backup set, for example, **NetAppMySQLZmanda**.
3. Under **Comments**, enter the description of the backup set.
4. Click **Create Backup Set**.

The **Backup Summary** page appears allowing you to set further options for the backup set ([Figure 9](#)).

## BACKUP—WHAT, WHERE, HOW SUMMARY

The screenshot shows the Zmanda Recovery Manager for MySQL web interface. At the top, there is a navigation bar with tabs for 'Backup', 'Monitor', 'Report', 'Admin', and 'Restore'. The 'Backup' tab is active. Below the navigation bar, there is a breadcrumb trail: 'About | User Guide' followed by 'what', 'where', 'when', 'how', and 'summary'. The 'Backup Set' dropdown menu is set to 'NetAppMySQLZma'. The main content area is titled 'What would you like to backup?' and contains a green message: 'Configuration parameters have been saved for this backup set.' Below this message, there are two main sections: 'MySQL Server Parameters' and 'MySQL User Parameters'. The 'MySQL Server Parameters' section includes: 'Server Type\*' (MySQL Server), 'Connection Type' (Port and Socket File radio buttons), 'Socket File Path' (empty text box), 'Host' (10.73.69.119), and 'MySQL Client Utilities Path' (usr/bin). The 'MySQL User Parameters' section includes: 'Username' (root), 'Password' (masked with asterisks), and 'SSL Options' (empty text box). At the bottom, there is a 'What to backup' section with 'Backup Type' set to 'All Databases' and a 'Go' button.

Figure 6) Parameters to back up.

### MySQL Server Parameters

The **Server Parameters** option allows you to specify the connection details for the MySQL server to back up.

1. In the **Server Type** drop-down list, the **MySQL Server** option is set by default.
2. Under **Connection Type**, select one of the following options:
  - Select **Port** to enter a port number for communicating with the MySQL server.
  - Select **Socket File** to enter a path to the socket file.

The default path is `/tmp/mysql.sock`.

3. Under **Host**, enter the host name or IP address of the MySQL server being backed up, or **localhost** if the MySQL server and the ZRM server are on the same machine.

In our example the MySQL server host IP is `10.73.69.119`.

4. Under **MySQL Client Utilities Path**, if you have installed the same versions of MySQL client commands on the ZRM server and the MySQL server (typically in `/usr/bin`), use the factory (or site settings) defaults for all backup sets. If you installed different versions of MySQL on the ZRM server and the MySQL server (that is, the backup client), enter the full path to where MySQL binary commands are installed on the backup client. For example: `/opt/lampp/bin`.

### MySQL User Parameters

1. Under **Username**, enter the user name of the MySQL user on the MySQL server.
2. Under **Password**, enter the password for the MySQL user.
3. Under **SSL Options**, if the connection from the ZMC server to the remote server is via SSL, specify the `MySQL ssl` parameters. If you are backing up databases on the local host, leave this field empty.

**Note:** The MySQL user should have the following permissions:

**Backup user:** LOCK TABLES, SELECT, FILE, RELOAD, SUPER

**Restore user:** CREATE, DROP, INDEX, SHUTDOWN, INSERT, ALTER, SUPER, REPLICATION CLIENT, CREATE ROUTINE, ALTER ROUTINE, RELOAD ON

If you are restoring from logical backups, additional privileges will be required for the restore user. For example, if there are stored procedures in the logical backup image being restored, the restore user must have `CREATE ROUTINE` and `ALTER ROUTINE` privileges. If you are not sure of the privileges that are required for restoration, temporarily grant the restore user all privileges for the databases and/or tables being restored.

### **What to Back Up**

Choose an option from the drop-down menu. Note that because the database(s) and table(s) shown on this page are updated dynamically based on information retrieved from the MySQL server, these may not match what has been set on the **Site Settings** page.

The **Go** button allows you to refresh the connection to the MySQL server(s) to determine what databases and tables are currently available for backup. Available options are:

**All Database(s):** Select this option to include all databases and all tables for backup.

**Specific Database(s):** Select this option to display a list of databases available for backup on the MySQL server. Select all the databases that you want to add to this backup set.

**Specific Table(s):** Select this option to display a drop-down menu that lets you select a database from which to select tables to back up, followed by the list of tables in the selected database; check all the tables that you want to add to this backup set.

### **Save**

Click **save** to commit changes you have made to the currently selected backup set.

The following message appears:

Configuration parameters saved for this backup set

You can now proceed to the **Backup Where** page ([Figure 7](#)) to select a target device (or directory) to store the backup image.

### **Backup Where**



The screenshot shows the Zmanda Recovery Manager for MySQL interface. At the top, there is a navigation bar with tabs for 'Backup', 'Monitor', and 'Restore'. Below the navigation bar, there is a message: 'Configuration parameters have been saved for this backup set.' The main section is titled 'Backup Where Parameters' and contains three input fields: 'Destination Directory' with the value '/var/lib/mysql-zrm', 'Temporary Directory' with the value '/tmp', and 'Retention Policy' with the value '60' and a dropdown menu set to 'days'.

Figure 7) Location to back up.

Using this page, you can set the location and the retention policies for the backup, as follows:

- **Destination Directory**

Backups are stored under the backup directory of the ZMC machine. The default directory is `/var/lib/mysql-zrm`. The backup user should have permissions to create directories in the Destination Directory.

The recommended practice is to mount a file system at `/var/lib/mysql-zrm`. In our setup we use NFS-mounted storage from the NetApp storage system for the MySQL backup data.

For example: If backup set "daily" requires 150GB of disk space for uncompressed backups and backup images upon compression are 100GB, the backup directory file system should have at least 225GB (1.5 \* 150GB) for space. After the backup run is completed, only 100GB of disk space will be used.

- **Temporary Directory**

ZMC creates temporary files and directories during backup. This directory should have sufficient space to hold at least one full backup on the MySQL server during backup and on the backup host during restore. The default is to use the OS-specified temporary directory (`$TMPDIR` on Linux).

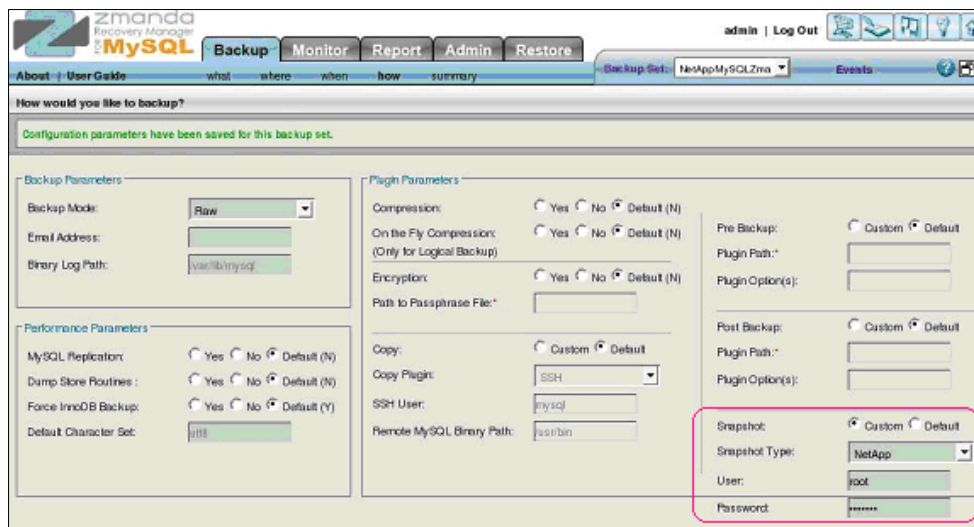
- **Retention Policy**

ZMC creates all the backups on a hard disk or external storage. When storage is exhausted, backup runs will fail. A retention policy sets the limit of the period for which the backup set will be retained on the hard disk or the storage. Images older than the retention policy specified, for example, 60 days, are automatically purged.

- **Save Button**

Click **save** to save the parameters. Error/Warning messages are displayed in the message box.

## Backup How



The screenshot shows the Zmanda Recovery Manager MySQL Backup preferences window. The window is titled "zmanda Recovery Manager MySQL" and has a navigation bar with "Backup", "Monitor", "Report", "Admin", and "Restore" tabs. The "Backup" tab is selected. The window displays configuration parameters for a backup set named "NetAppMySQLZma".

The configuration parameters are organized into several sections:

- Backup Parameters:** Backup Mode (Raw), Email Address, Binary Log Path (/var/lib/mysql).
- Performance Parameters:** MySQL Replicator (Yes/No/Default (N)), Dump Store Routines (Yes/No/Default (N)), Force InnoDB Backup (Yes/No/Default (Y)), Default Character Set (utf8).
- Plugin Parameters:** Compression (Yes/No/Default (N)), On the Fly Compression (Only for Logical Backup) (Yes/No/Default (N)), Encryption (Yes/No/Default (N)), Path to Passphrase File, Copy (Custom/Default), Copy Plugin (SSH), SSH User (mysql), Remote MySQL Binary Path (/usr/bin).
- Pre Backup:** Custom/Default, Plugin Path, Plugin Option(s).
- Post Backup:** Custom/Default, Plugin Path, Plugin Option(s).
- Snapshot:** Custom/Default, Snapshot Type (NetApp), User (root), Password (xxxxxxx).

The Snapshot section is highlighted with a red box.

Figure 8) Backup preferences.

- Select the Snapshot type as **NetApp** and enter the user name **root** and the password **xxxxxxx** to connect to the NetApp storage system.

## Prerequisites for Using NetApp Snapshot Copies

- The MySQL backup user must be granted `sudo` privileges to execute NetApp commands on the MySQL server. To do this, do one of the following:
  - Add a line similar to the following example to `/usr/local/etc/sudoers` on the MySQL server:

```
mysql
Zmanda.btcppe.netapp.com=NOPASSWD:/bin/mount,NOPASSWD:/bin/umount,NO
PASSWD:/bin/df
```

Where:

ZRMserver.mycompany.com is the fully qualified domain name for the ZRM server.

Note that if the `mount`, `umount`, and `df` commands are installed in nonstandard locations, the above example would not work without editing it to reflect the different paths.

- o Provide full access to the `mysql` user as below for testing purposes in `/etc/sudoers`. Then provide `df`, `umount`, and `mount` access to the `mysql` user.

```
%mysql ALL=(ALL) NOPASSWD: ALL
```

**Note:** If we use RHEL5 – U1 and above, make sure the “`Default requiretty`” line is not present in the `/etc/sudoers` file.

- Enable the `ssh-key` auth between the MySQL server and the ZRM server for the `mysql` user because the backup happens as `mysql` user. If the ZRM is remote to the MySQL server, the `ssh-keys` should be shared between the ZRM and MySQL servers.
- If the ZRM is remote to the MySQL server, the UID and GID of “`mysql`” user should be the same on both the ZRM and MySQL machines.
- The `snap reserve` for the NetApp volume must be adequate to accommodate changed data. The NetApp default of 20% may be low or high for your site, depending on data change rates. For testing purposes we use 5%.
- To determine the `snap reserve`, use a NetApp **snap reserve** `volume name` command such as `snap reserve mysqldata`.
- All MySQL database files (data, log, indexes) must be stored in NetApp volumes to ensure consistency. `/var/lib/mysqldata` is the `datadir` that is mounted from the `mysqldata` volume from the NetApp storage system.

If any of the files are not on NetApp volumes, a raw backup using `mysqlhotcopy`, or a logical backup using `mysqldump`, will be taken based on the storage engines of the tables in each of the databases.

## Backup Summary

The Backup Summary page (Figure 9) lists all relevant settings for the backup set in one place, showing each option and where the option was set (that is, either factory, site, or in the backup set itself).

The screenshot shows the Zmanda Recovery Manager interface. At the top, there are navigation tabs: Backup, Monitor, Report, Admin, and Restore. The current backup set is 'NetAppMySQLZma'. Below the navigation, there is a section titled 'Summary of backup set parameters' with a message: 'Click any link in the Backup Sets Parameter Page to go to where that value is set (Backup Set or Site Settings).' On the left, there is a 'Set Site Defaults' button and a message: 'Backup set parameters verified successfully. Click the "Backup Now" button to run an unscheduled backup..'. At the bottom left, there are buttons for 'Full Backup' and 'Backup Now'. The main part of the page is a table titled 'Backup Set Parameters' with columns for Name, Value, and Set In.

Name	Value	Set In
<b>What</b>		
Backup Type	All databases	Factory Defaults
Host	10.73.69.119	Backup Set
MySQL Client Utilities Path	/usr/bin	Backup Set
Password	*****	Backup Set
Port Number		
SSL Options		
Socket File Path		
Username	root	Backup Set
<b>Where</b>		
Destination Directory	/var/lib/mysql-zrm	Factory Defaults
Retention Policy	60D	Backup Set
Temporary Directory	/tmp	Factory Defaults
<b>How</b>		
Backup Mode	raw	Factory Defaults
Binary Log Path	/var/lib/mysql	Factory Defaults

Figure 9) Summary of backup set parameters.

1. Select **Backup set parameters verified successfully.**

2. To start the backup, click **Backup Now.**

Once the backup starts, the Monitor tab automatically opens and shows the status of the MySQL server backup.

3. You can also start backup from the CLI with the corresponding backup set configuration:

```
mysql-zrm --action backup --backup-set NetAppMySQLZmanda -verbose
```

**Note:** When the backup fails or stops in the middle of backup, before starting the backup of the same backup set configuration remove the concerned backupset .mysql-zrm.pid file from the backup set folder.

## MONITOR AND SUMMARY REPORT

The Monitor page shows the progress of backups as they are executed (Figure 10).

The screenshot shows the Zmanda Recovery Manager interface. At the top, there are navigation tabs: Backup, Monitor (selected), Report, Admin, and Restore. The user is logged in as 'admin'. The 'Backup Set' is 'NetAppMySQLZma'. The main content area is divided into two panes:

- Backup Run Summary:** Shows a green checkmark status. Parameters include: Date: 2008-03-14, Time: 19:25:49, Destination: /var/lib/mysql-zm, Host: 10.73.69.119, Level: Full, Mode: raw, Read Locks Time: 00:00:00, Time Taken: 03:28:51, Size: 443.07 GB, Replication: No, Compressed: No, Encrypted: No.
- Backup Run Details:** Shows a table of phases with their durations and messages. All phases are marked with a green checkmark.

Phase	Time Taken	Messages
Initialization	00:00:00	Location of Backup: /var/lib/mysql-zm/NetAppMySQLZmanda/20080314192549
Running pre backup plugin	00:00:00	
Flushing logs	00:00:00	
Creating snapshot based backup	03:28:00	Databases Backed Up: db94 db76 db96 db51 db24 db12 db50 db117 db91 db26 db60 db37 db19 db102 db93 db95 db9 db8 db34 db89 db111 one db80 db78 db101 db15 db57 db47 db109 db99 db46 db62 db84 db64 db54 db75 db71 db6 db49 db18 db55 db32++ db105 db106 db113 db107 db53 db28 db52 db118 db59 db97 db68 db45 db98 db14 test bin db2 db40 db83 db21 db41 db61 db58 db31 db74 db42 db82 db110 db30 db120 db72 db119 db16 db44 db56 db114 db3 db23 db63 db10 db1_copy db108 db38 db67 db66 db81 db79 db43 db48 db103 db7 db33 db36 db4 db29 db70 db17 db115 db104 db90 db85 db73 db5 db27 db1 db88 db13 db25 db32 db87 db77 db69 db39 db92 db112 db22 mysql db65 db86 db116 db100 db35 db20
Find table type	00:00:00	
Calculating backup size & checksums	00:00:51	
Running post backup plugin	00:00:00	
Cleanup	00:00:00	

Figure 10) Monitor window.

The Monitor window is divided into two panes:

- The left pane (Backup Run Summary) shows a statistical view of backup processes.
- The right pane (Backup Run Details) displays a step-by-step view of processes.

### Status

Both panes use these symbols to indicate the status of the backup:

- The yellow exclamation point (!) indicates an informational message.
- The green checkmark symbol indicates that a task was executed successfully.
- The red octagon ("stop sign") symbol indicates that the process stopped with fatal errors.

### Backup Run Summary Pane

The Backup Run Summary shows a status icon (explained below), followed by the parameters in effect for the backup set.

### Backup Run Details

Backup Run Details shows step by step how the previous and current backup runs progressed. In our example we use Snapshot-based backup with the following processes involved:

- Initializing
- Running pre-backup plug-in
- Flushing logs
- Creating Snapshot-based backup
- Finding table type
- Calculating backup size and checksums
- Running post-backup plug-in
- Cleaning up

The time needed to take the Snapshot copy on the NetApp storage system is just seconds. The duration of the Snapshot copy process does not depend on the size of the volume.

- **Phase**

Phase information in the first column contains a symbol and the name of the process that has been executed.

- **Time Taken**

The second column (Time Taken) shows the duration of each process, displayed in hh:mm:ss format. In our example the time taken for the 443.07GB MySQL database is 3 hours, 28 minutes, and 51 seconds, which includes taking the Snapshot copy, checking the checksums, and copying the backup files and folders to the ZRM server from the Snapshot copy.

- **Messages**

Messages contain a wealth of information about the backup process, including creating the full path of directories where data is being stored. Messages may also contain warnings or errors found, including which process generated the message. In our example we use 126 databases; each database has 431 tables and each table has 13,645 records of it.

### **Summary Report**

ZRM for MySQL automatically generates backup reports after the backup run is completed (or if it fails for some reason). This backup summary report includes:

- Status of the current/last backup run
- Backup statistics
- Location of the backup image
- Backup level

If the machine on which the ZMC is running is configured to send mail, reports can be automatically e-mailed after every backup is run.



## 8.2 SUMMARY REPORT OVERVIEW

The Summary Report page displays the progress and final status of previous backup runs of a given backup set.

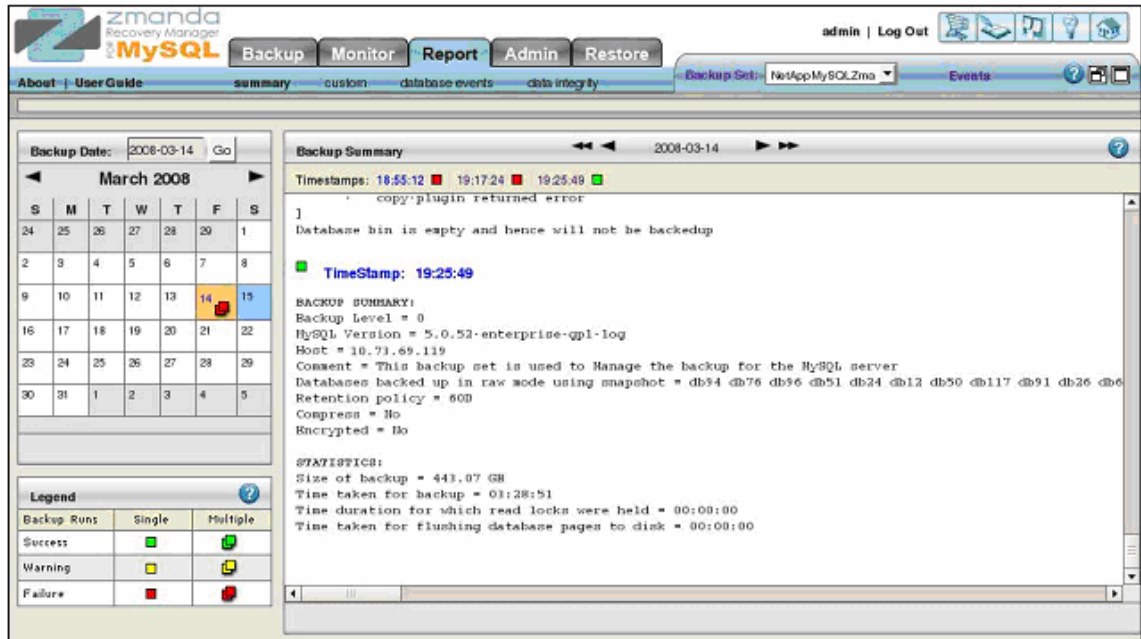


Figure 11) Backup summary.

The Summary Report page consists of two panes:

- The left pane shows a calendar control from which you can select report dates, and a legend that explains the report icons.
- The right pane displays the report for the date selected on the calendar.
- To select reports, do one of the following:
  - Browse buttons at the top of the report.
  - Enter a date on the left pane and click **Go**.
  - Select a date on the calendar.

### SUMMARY REPORT PAGE MAIN FUNCTIONALITY

The Summary Report page displays the progress and final status of previous backup runs of a given backup set.

#### DATE AND LEGEND PANE

The Summary Report page opens with the calendar set to the current month.

#### Selecting a Report Date

Do one of the following:

- Enter the date (mm/dd/yy[yy]) and click **Go**.
- Click on any icon in the calendar.

The report showing summary data (if any) for the selected date appears.

## **Calendar Control**

The calendar control offers a graphical way to choose a report date. Dates that include backup data have a status icon; click the icon to display the given report (you cannot select dates with no backups to report on).

When you click a report icon, it changes color to show that it has been visited:

Date Selected & Visited

The example above shows that March 14 was selected and visited.

## **LEGEND**

The Legend pane displays the differences between six outcomes of a backup run. These are:

- Normal Backup Single or Multiple Backup Runs on the same day.
- Backup with Warning(s), Single or Multiple Backups on the same day.
- Backup with Error(s), Single or Multiple Backups on the same day.

In all cases, there will be only one legend in the calendar control for that date. However, the summary will contain details of all the backups that were run that day for that backup set.

## **BACKUP SUMMARY**

The Backup Summary consists of:

- **Backup Summary:** the what, when, where, and how of the backup run
- **Statistics:** the amount of data backed up and the time it took

The Backup Summary has the following navigational aids:

## **Date Browse Button**

The **Date Browse** button allows you to conveniently move back and forth one day (using < or >) or one week (using << or >>) at a time.

## **Timestamp Links**

Timestamp links below the Browse buttons display the time at which the backup run was initiated. Clicking the **Timestamp** link takes you to the **Restore What** page with the date and time automatically populated.

**Note:** Similar links to the **Restore What** page are featured in predefined reports for successful backup runs. They are also included in the **Database Events** page.

## **8.3 RESTORE**

It is possible to perform a restoration through the ZMC as you do with backup. In the example described we use the CLI for restoration:

```
mysql-zrm --action restore --host 192.168.1.62 --source-directory  
/var/lib/mysql-zrm/NetAppMySQLZmanda/20080227171032 --backup-set  
NetAppMySQLZmanda --verbose --ssh-user=mysql
```

Where:

host 192.168.1.62: the location to restore

source-directory /var/lib/mysql-zrm/NetAppMySQLZmanda/20080227171032: backup data used to restore

backup-set NetAppMySQLZmanda: the backup set used to restore

ssh-user=mysql: the user who does the restoration using the ssh protocol

## 8.4 COMMANDS AND FILES USED FOR BACKUP

```
[root@mysql 20080314192549]# df -HT /var/lib/mysql*
```

Filesystem	Type	Size	Used	Avail	Use%	Mounted on
/dev/sda3	ext3	53G	6.4G	43G	13%	/
10.173.68.23:/vol/binlog	nfs	36G	3.2M	36G	1%	/var/lib/mysqlbinlog
10.173.68.23:/vol/mysqlatabase	nfs	541G	479G	63G	89%	/var/lib/mysqldata

```
[root@Zmanda 20080314192549]# df -HT /var/lib/mysql*
```

Filesystem	Type	Size	Used	Avail	Use%	Mounted on
10.73.68.22:/vol/zmandaback	nfs	585G	136G	449G	24%	/var/lib/mysql-zrm

### MySQL server basic configuration for Snapshot copy backup

```
[root@mysql 20080314192549]# cat /etc/my.cnf
```

```
[client]
port                = 3306
socket              = /var/lib/mysql/mysql.sock

[mysqld]
port                = 3306
socket              = /var/lib/mysql/mysql.sock
skip-locking
key_buffer          = 384M
max_allowed_packet = 1M
table_cache         = 512
sort_buffer_size    = 2M
read_buffer_size    = 2M
read_rnd_buffer_size = 8M
myisam_sort_buffer_size = 64M
thread_cache_size   = 8
query_cache_size    = 32M
thread_concurrency  = 8
datadir=/var/lib/mysqldata
set-variable=max_binlog_size=1M

log-bin=/var/lib/mysql/bin/mysql-bin
server-id           = 1

tmpdir              = /tmp/
log-update          = /path-to-dedicated-directory/hostname
sync_binlog         = 1

[mysqldump]
quick
```

```
max_allowed_packet = 16M
```

```
[mysql]  
no-auto-rehash
```

```
[isamchk]  
key_buffer = 256M  
sort_buffer_size = 256M  
read_buffer = 2M  
write_buffer = 2M
```

```
[myisamchk]  
key_buffer = 256M  
sort_buffer_size = 256M  
read_buffer = 2M  
write_buffer = 2M
```

```
[mysqlhotcopy]  
interactive-timeout
```

```
[root@mysql mysqldata]# /etc/init.d/mysql restart
```

```
Shutting down MySQL [ OK ]  
Starting MySQL [ OK ]
```

```
[root@Zmanda mysql-zrm]# /etc/init.d/zmc_zrm_init restart
```

```
Stopping /etc/init.d/zmc_zrm_init:  
Stopping ZRM services  
prg: zmc_dblogger. pid: 6252  
kill 6252: No such process  
prg: rss2event. pid: 6253  
kill 6253: No such process  
prg: zmc_eventserver. pid: 6237  
kill 6237: No such process  
/usr/sbin/setenforce: SELinux is disabled  
/opt/Zmanda/zrm/zmandactl.sh : httpd stopped  
/opt/Zmanda/zrm/zmandactl.sh : mysql stopped  
/usr/sbin/setenforce: SELinux is disabled  
Starting /etc/init.d/zmc_zrm_init:  
/usr/sbin/setenforce: SELinux is disabled  
/opt/Zmanda/zrm/zmandactl.sh : mysql started  
Starting mysqld daemon with databases from /opt/Zmanda/zrm/mysql/data/  
Syntax OK  
/opt/Zmanda/zrm/zmandactl.sh : httpd started  
/usr/sbin/setenforce: SELinux is disabled  
Checking ZMC Services Status...  
EventServer is not running.  
DbLogger is not running.  
Rss2event is not running.  
Starting EventServer.  
Starting dblogger.
```

```
Starting rss2event.  
log_parser is not running  
report is not running  
start_monitor is not running  
stop_monitor is not running
```

```
Starting ZRM services  
log_parser is running  
report is running  
start_monitor is running  
stop_monitor is running
```

## 9 REFERENCES

- For NetApp:  
[http://www.netapp.com/us/library/technical-reports/tr\\_3601.html](http://www.netapp.com/us/library/technical-reports/tr_3601.html)
- For ZRM:  
<http://network.Zmanda.com/documentationEnterprise.php>  
<http://www.Zmanda.com/zrm-mysql-enterprise.html>

## 10 ACKNOWLEDGEMENTS

The authors would like to thank the following individuals for their direction and guidance in creating this paper:

Daniel Morgan—Director, Solution Technologies, NetApp

Uday Shet—Manager, NB PPE (Solutions Engineering), NetApp

Michelle Nguyen—Senior Product Manager, Enterprise Applications, NetApp

Michael B. Flannery—Field Technology Lead, NetApp

Paddy Sreenivasan—Vice President, Engineering, Zmanda, Inc.

Bill Mason—Director, Strategic Alliances, Zmanda, Inc.

## 11 DISCLAIMER

Each environment has its own specific set of requirements and no guarantees can be given that the results presented in this report will work as expected on other platforms. This paper should assist in the research and troubleshooting that may be required in a particular case and serve as a checklist of items to be aware of. Please forward any errors, omissions, differences, new discoveries, and comments about this paper to [nkarthik@netapp.com](mailto:nkarthik@netapp.com).

© 2008 NetApp. All rights reserved. Specifications are subject to change without notice. NetApp, the NetApp logo, Go further, faster, Data ONTAP, FlexVol, RAID-DP, SnapManager, SnapMirror, Snapshot, SnapRestore, and WAFL are trademarks or registered trademarks of NetApp, Inc. in the United States and/or other countries. Mac is a registered trademark of Apple, Inc. UNIX is a registered trademark of The Open Group. Veritas is a trademark of Symantec Corporation. Solaris is a trademark of Sun Microsystems, Inc. Linux is a registered trademark of Linus Torvalds. Windows is a registered trademark of Microsoft Corporation. All other brands or products are trademarks or registered trademarks of their respective holders and should be treated as such.

