Using Sun Fire™ X4540 Server With Zmanda Recovery Manager 2.2 for MySQL™ Database

Thomas Hanvey (Sun Microsystems)

Dmitri Joukovski and Ken Crandall (Zmanda)

October 2008

Table of Contents

Introduction

This paper discusses how to quickly configure and deploy the Sun Fire X4540 server with Zmanda Recovery Manager 2.2 for MySQL databases. For more information on ZRM Enterprise, see: http://zmanda.com/zrm-mysql-enterprise.html.

MySQL™ software has become the world's most popular open source database because of its ease of use, high performance, and high reliability (see http://www.sun.com/software/products/mysql). Regardless of how you use MySQL software (for example, with a web site or a custom MySQL application), you must ensure successful, secure, and consistent backups with minimal impact on applications.

The Sun Fire™ X4540 server can deliver massive storage capacity and remarkable throughput so it is well-suited as a nearline storage platform for backup and restore applications. Combining the power of the Solaris™ 10 Operating System with the data integrity and simplified administration of ZFS, the Sun Fire X4540 server can be an ideal candidate for streamlining and improving backup and restore operations.

Zmanda Recovery Manager (ZRM) for MySQL provides comprehensive backup and recovery capabilities for MySQL databases. It simplifies database administration by providing an easy-to-use, flexible, and robust backup and recovery solution.

With ZRM you can:

- Perform backups that are optimized for your storage engine and your MySQL configuration.
- Recover a database easily to any point in time or to any particular transaction, for example, just before a user made an error.
- Centrally manage all backup and recovery operations for a single or multiple MySQL servers from a web-based management console.
- Schedule full and incremental backups of your MySQL database.
- Define retention policies and automatically delete backups that have expired.
- Visually review MySQL logs for operational and security reasons.
- Easily identify recovery points.
- Choose flexible logical or faster raw backups of your database.
- Provide hot backups of online MySQL databases via various snapshot technologies.
- Back up your remote MySQL database through a firewall.
- Configure on-the-fly compression and/or encryption of your MySQL backups to meet your storage and security needs.
- Get email or RSS notification about the status of your backups and receive MySQL backup reports.

Configuration

Overview

Figure 1 shows an example of Zmanda Recovery Manager (ZRM) running on the Sun Fire X4540 server and protecting remote MySQL servers running on Solaris OS, Linux, and Microsoft Windows machines with MyISAM and InnoDB tables, and a MySQL Cluster with NDB storage engine.

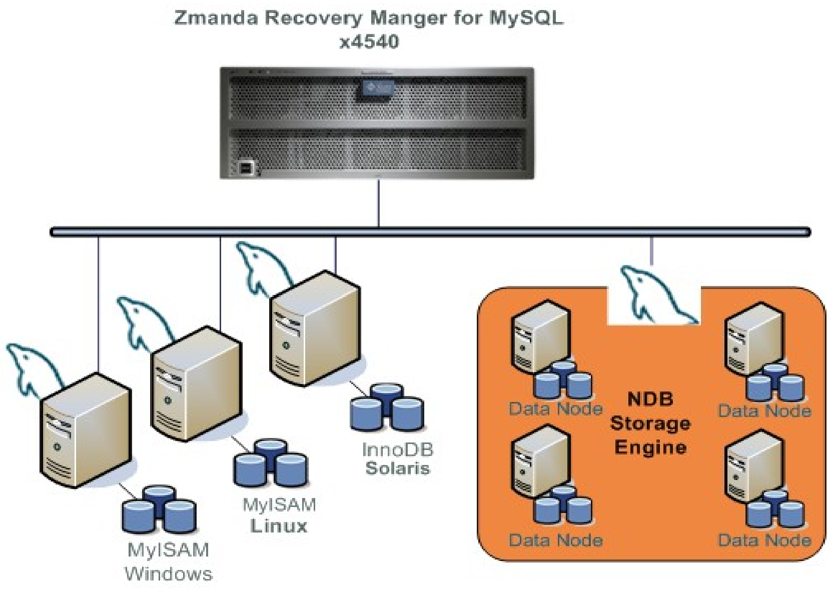


Figure 1. Centralized Backup of MySQL Server With ZRM on Sun Fire X4540 Server

Notes:
X4540 here refers to Sun Fire X4540 Server.
All figures reprinted with permission from Zmanda.

The complete ZRM deployment consists of the following:

- ZRM Server and Zmanda Management Console (ZMC) to centralize all backup and recovery operations for MySQL databases. You need at least one ZRM server for your environment. In Figure 1, ZRM is configured on the Sun Fire X4540 server.
- Zmanda ZRM clients installed on remote Solaris, Linux, or Microsoft Windows systems where you run MySQL.

Solaris Configuration

For this guide, the Sun Fire X4540 server used for ZRM certification was running the Solaris 10 8/07 OS. This is the minimum recommended version to use because various ZFS improvements were introduced in this build.

The Sun Fire X4540 server is powered by up to 8 CPUs operating at 2300 MHz and provides 48 SATA drives with a total capacity close to 50 Tbytes in a 4U enclosure. The two boot drives are mirrored using Solaris Volume Manager with UFS. The rest of the drives are available for backup data.

The default NIC configuration on the Sun Fire X4540 server allows up to four physical Gigabit Ethernet connections to be attached. The connections provide backup ability for up to four separate subnets. Alternatively, the links can be aggregated together using `dladm`(1M) for improved throughput to a single subnet.

Zpool Configuration

The decision about how to configure zpools needs to be based on a number of factors:

- What is the required performance needed? (IOPS/throughput)
- What is the required data protection level needed? (RAIDZ, RAIDZ2, mirror)
- What is the required capacity?

After gathering and analyzing the requirements, a configuration that balances the needs of those requirements can be chosen.

When configuring the system for network backups, the ability of the Sun Fire X4540 server to receive data over the network is also a factor. For example, a 500-Mbyte/sec backup speed requirement cannot be met solely by using the 4 pre-installed Gigabit Ethernet ports. Additional network cards would be required to accommodate such throughput.

Care should be taken that a system is not over-configured for disk performance, at the expense of space or data protection, when the entire system (encompassing MySQL backup clients and network infrastructure) is not able to send data at adequate levels. Likewise, a system should not be over-configured for space at the expense of performance or data protection.

For the purposes of this guide, the Sun Fire X4540 server was configured with one zpool consisting of 12 drives and 2 spares in a RAIDZ2 configuration. This provided 7 Tbyte of capacity for backups with a very high level of protection against drive failure and performance sufficient to handle 300 to 350 Mbyte/sec of backup traffic over the network.

Example zpool creation commands for a pool zmanda:

```
bash-3.00# zpool create -f zmanda raidz2 c0t0d0 c1t0d0 c2t0d0 c3t0d0 c4t0d0 c5t0d0
bash-3.00# zpool add zmanda raidz2 c0t2d0 c1t2d0 c2t2d0 c3t2d0 c4t2d0 c5t2d0

bash-3.00# zpool list
NAME                    SIZE    USED   AVAIL    CAP  HEALTH     ALTROOT
zmanda                 10.9T    204K   10.9T     0%  ONLINE     -

bash-3.00# zpool add zmanda spare c0t7d0 c1t7d0

bash-3.00# zpool list
NAME                    SIZE    USED   AVAIL    CAP  HEALTH     ALTROOT
zmanda                 10.9T    210K   10.9T     0%  ONLINE     -

bash-3.00# zpool status zmanda
  pool: zmanda
 state: ONLINE
 scrub: none requested
config:

        NAME        STATE     READ WRITE CKSUM
        zmanda      ONLINE       0     0     0
          raidz2    ONLINE       0     0     0
            c0t0d0  ONLINE       0     0     0
            c1t0d0  ONLINE       0     0     0
            c2t0d0  ONLINE       0     0     0
            c3t0d0  ONLINE       0     0     0
            c4t0d0  ONLINE       0     0     0
            c5t0d0  ONLINE       0     0     0
          raidz2    ONLINE       0     0     0
            c0t2d0  ONLINE       0     0     0
            c1t2d0  ONLINE       0     0     0
            c2t2d0  ONLINE       0     0     0
            c3t2d0  ONLINE       0     0     0
            c4t2d0  ONLINE       0     0     0
            c5t2d0  ONLINE       0     0     0
        spares
          c0t7d0    AVAIL
          c1t7d0    AVAIL

errors: No known data errors
```

First, we created the RAIDZ2 pool zmanda with six drives. Then we added six more drives and verified that we had a pool with 10.9 Tbyte of physical space. Depending on which RAIDZ configuration you choose, the actual available capacity might be different. See the zpool(1M) man page for more details.

The next step was adding two spare drives. Lastly, we checked the pool status and verified the availability of the `zmanda` pool for configuring ZFS.

RAIDZ2 is a natural choice for a backup server, which contains information vital to business operations. RAIDZ2 can survive up to two drive failures, regardless of whether they are due to corrupt data coming off the drives, a physical failure of the drives, or drives that are missing. With a suitably large number of disks, RAIDZ2 does not impose a greater penalty in capacity or performance than RAIDZ.

ZFS Configuration

The Zmanda Recovery Manager on the Sun Fire X4540 server Backup Server is configured to write backup images to a directory. For this sample installation, we chose to put the backup images for all backup sets on a single ZFS file system. On a system where multiple backup sets might be running simultaneously, it would be advantageous to have separate ZFS file systems and pools for each backup set, in order to spread reads and writes around the available disks.

Example of ZFS file system creation:

```
bash-3.00# zfs create -o mountpoint=/var/lib/mysql-zrm zmanda/zrm

bash-3.00# zfs list
NAME            USED  AVAIL  REFER  MOUNTPOINT
zmanda          203K  7.13T  49.0K  /zmanda
zmanda/zrm      49.0K 7.13T  49.0K  /var/lib/mysql-zrm
```

ZRM Configuration

The Sun Fire X4540 server is configured as a ZRM server with Zmanda Management Console (ZMC). To install ZRM and ZMC, follow these overall steps, which are explained in more detail in subsequent sections.

1. From the Zmanda Network web site (which requires registration), download the following:
   - ZRM installer binary for Solaris 10 systems with Intel® or AMD Opteron™ processors: `ZRM-enterprise-2.2-installer-intel.bin`
   - Prerequisite packages: `solaris-10-zrm-dependency-intel.tar.gz`
2. Extract the prerequisite packages, `solaris-10-zrm-dependency-intel.tar.gz` , and install all the included packages.
3. Extract the binary package and run the Zmanda Rapid Installer to install ZRM.
4. After the installation is complete, the ZMC Welcome Page is displayed, and you can begin configuring backup sets.

Installing the Required Dependencies

The required dependency packages are provided in a compressed tar archive file, `solaris-10-zrm-dependency-intel.tar.gz`, that you can download from the Zmanda Network: http://network.zmanda.com

The dependency packages provided are as follows:

- SMCcoreu
- CSWgcc3corert
- CSWgcc3g++rt
- SMCgpgme
- SMCgrep
- SMClgcc346
- SMClgpger
- SMCliconv
- SMClintl
- SMCpcre
- SMCsudo
- SMCtar

These packages are also available from Sunfreeware at http://sunfreeware.org and from Blastwave at http://blastwave.org.

Once the archive have been downloaded, use the gtar(1M) command to extract the prerequisite packages. Next, use the pkgadd(1M) utility to install the packages.

Installing ZRM

The installation program for ZRM is called the Rapid Installer. The Rapid Installer contains the ZRM Server and ZRM Client, as well as the ZMC web user interface.

Note: The following tasks must be performed as super user (the root user).

To run the Rapid Installer, first make sure that the binary file is executable:

```
# chmod +x ZRM-enterprise-2.2-installer-intel.bin
```

If you are installing from within a graphical environment, simply run the installer by double-clicking it or executing it directly from the command line:

```
# ./ZRM-enterprise-2.2-installer-intel.bin
```

If you are installing from a text console or remotely (for example, via SSH) you can run the Rapid Installer in text mode:

```
# ./ZRM-enterprise-2.2-installer-intel.bin --mode text
```

The Rapid Installer then starts. Follow the on-screen instructions. The installer performs several tasks after creating and populating the Zmanda directories. These tasks are completed after the progress bar (which tracks only the archive extraction) shows 100% completion. These tasks take time. Please wait until they are completed.

The following software components are installed:

- The ZRM for MySQL software
- Zmanda Management Console with MySQL, PHP, Apache, PHPmyadmin, Perl and various Perl modules and their dependencies. These are installed under the `/opt/zmanda/zrm` and `/opt/zmanda/common` directories.

Zmanda Management Console

In addition to a powerful command line interface (CLI), ZRM also provides the Zmanda Management Console (ZMC).

ZMC is a browser-based user interface for setting up and managing all backup and recovery activities. It is integrated with the Zmanda Network, which provides certified ZRM binaries, white papers, demos, technical support, and a knowledge base to help you to deploy your backup solution.

ZMC provides context-sensitive help and the most current information about ZRM from the Zmanda Network.

Using ZMC, a backup administrator can accomplish all day-to-day backup activities, including the following:

- Configure backup policies and schedules for each MySQL server
- Start immediate backups
- Monitor ZRM activities and access reporting about backups and media
- Create, edit, and delete backup users and backup sets
- Recover databases
- Browse the MySQL transaction logs

To configure ZRM backups, all you need to do is specify the following:

- What to back up (specify server, databases, and tables to back up).
- Where to store the backup (specify a destination).
- When to perform the backup (specify scheduling options for full and incremental backup).
- How to perform the backup (specify copy mechanisms to use).

The ZRM user interface is organized into these tasks, and includes a corresponding page (Backup What, Backup Where, Backup When, and Backup How) to accomplish each task, as described in the sections that follow.

Configuring Backup Sets

To get started with ZRM, you need to first configure backup sets. A *backup set* is a key policy concept of ZRM that simplifies and optimizes the database backup and recovery process. ZRM lets you manage multiple backup sets so you can protect different MySQL servers and databases with settings appropriate for the respective applications. The backup set defines the following:

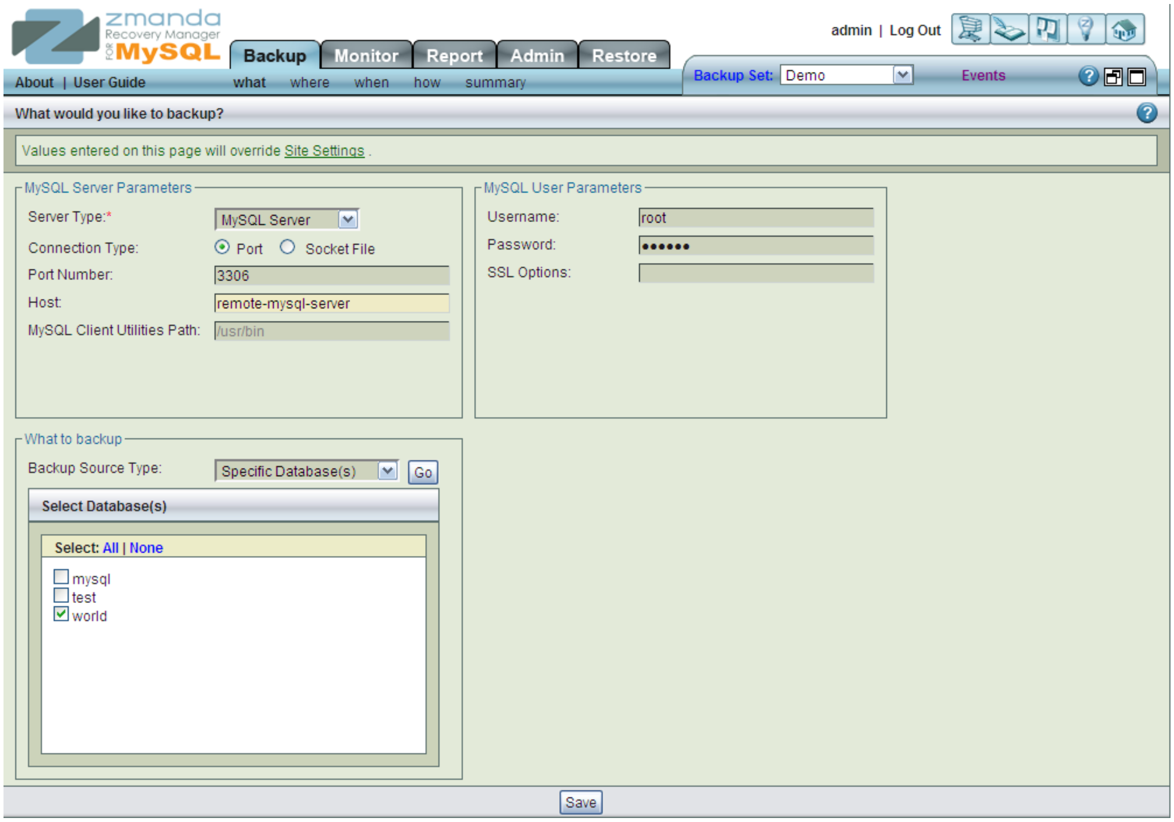
- List of servers and their databases/tables that are to be backed up
- Location where a given backup set should be stored

- Backup time and parameters for backup-level scheduling
- (OPTIONAL) Snapshot technology used to make backups
- (OPTIONAL) Local storage location of snapshots for near-CDP protection of MySQL databases

Each backup set contains at least one database or database table. Backup sets are easy to copy, add, and modify. All backup, reporting, monitoring, and restore actions are performed on the backup sets. This enables the backup administrator to abstract the backup policies and apply the same policy for multiple databases.

When creating backup sets for an application such as Customer Relationship Management (CRM), remember to include all the databases and tables required to maintain the referential integrity and consistency of the application. Put all the related databases in the same backup set.

The Backup What page, shown in Figure 2, simplifies backup source selection by automatically discovering all the databases and tables on a given MySQL server.



Backup Set: Demo

What would you like to backup?

Values entered on this page will override Site Settings .

MySQL Server Parameters

Server Type:* MySQL Server

Connection Type: ● Port ○ Socket File

Port Number: 3306

Host: remote-mysql-server

MySQL Client Utilities Path: /usr/bin

MySQL User Parameters

Username: root

Password: ••••••

SSL Options:

What to backup

Backup Source Type: Specific Database(s)

Select Database(s)

Select: All | None

□ mysql
□ test
☑ world

Figure 2. Backup What Page

Figure 2 shows an example of configuring a backup set called “Demo” for the MyISAM database “world” on a remote MySQL server.

Note: For many ZMC forms, the fields change depending on the context; selecting different values can change the options that are displayed. For example, if you choose to back up the MySQL cluster, Backup What displays options that are different from what is shown in Figure 2.

The Server Parameters section lets you specify the connection details for the MySQL server to back up. Choosing Port lets you enter a port number for communication with the MySQL server; choosing Socket File lets you enter a path to the socket file. You need to specify 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.

If you 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 software 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.

In the MySQL User Parameters section, enter the user name and password of the MySQL user on the MySQL server. If no user is specified here, ensure that you have the MySQL backup user (with appropriate privileges) specified explicitly as a Site Default, or ensure that the MySQL user specified in the `my.cnf` or `--options` file has the necessary privileges described in System Requirements. If the connection from the ZMC server to the remote server is via SSL, specify the MySQL SSL parameters here. If you are backing up databases on the localhost, leave this field empty.

For remote MySQL backups, the UID and GID of the MySQL user need to be the same on the ZRM client machine (where the MySQL software runs) as on the ZRM server.

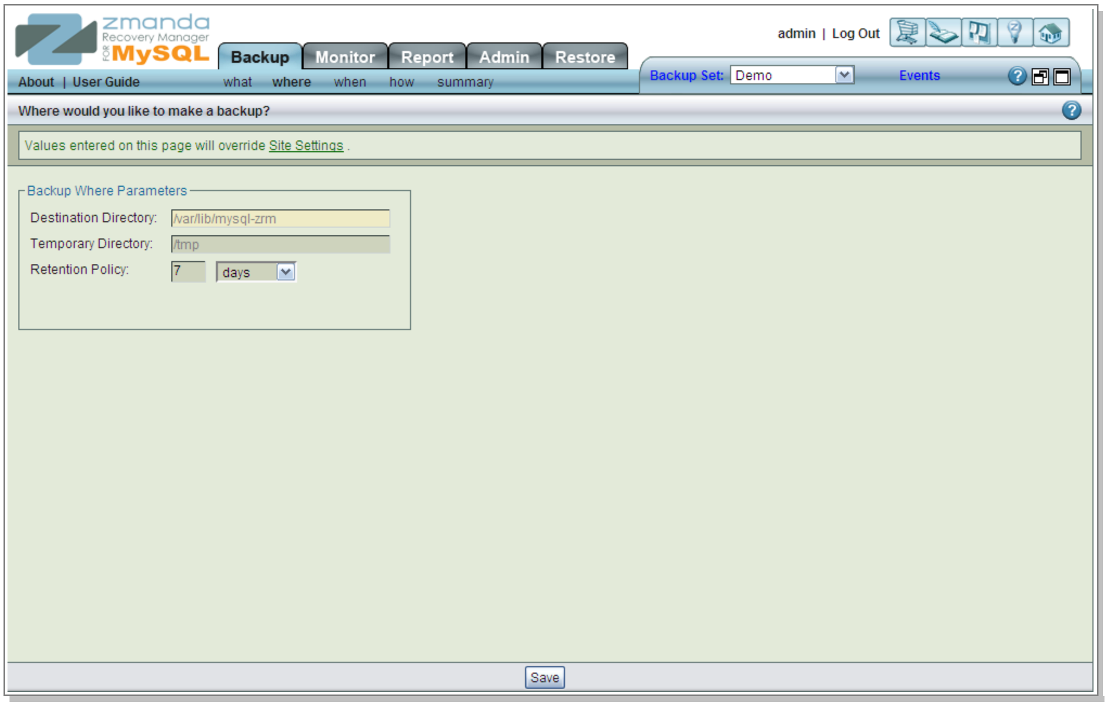
ZRM allows you to back up the following:

- All Database(s). Choosing this option includes all databases and all tables for backup.
- Specific Database(s). Choosing this option displays 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). Choosing this option displays a drop-down menu that lets you select a database from which to select tables to backup, followed by the list of tables in the selected database; select all the tables that you want to add to this backup set.

Clicking Save commits any changes you made to the currently selected backup set. You can then proceed to the Backup Where page to select a target device (or directory) to store the backup image.

Configuring Storage

Figure 4 shows the Backup Where page, which specifies where the backup images for the backup set will be stored and how long to retain them.



Where would you like to make a backup?

Values entered on this page will override Site Settings .

Backup Where Parameters

Destination Directory: /var/lib/mysql-zrm

Temporary Directory: /tmp

Retention Policy: 7 days

Figure 3. Backup Where Page for Configuring Storage

Regardless of where the remote MySQL server is hosted, backups are stored under the backup directory of the local machine where ZMC runs. The default is `/var/lib/mysql-zrm`. If you specify another directory, you must create it yourself on both the ZRM server and the MySQL server, and ensure that its permission settings allow read/write access to the MySQL backup user.

We already configured a ZFS volume at `/var/lib/mysql-zrm` as a repository for MySQL backup archives. For extra protection, the MySQL backup data can be migrated to other storage devices, for example, to a Sun StorageTek™ SL48 Tape Library using network-based backup and recovery software such as Amanda Enterprise.

You must allocate sufficient disk space to store backup copies of all your MySQL databases. In the unlikely event that the Sun Fire X4540 server does not have sufficient disk space allocated, the backup run will fail. The destination directory should have at least 150% of uncompressed backup space available for a successful backup run. This additional space is required even when backups are configured for compression, since the additional space is used for compression during the backup run. The additional space is freed after the backup run. It must also have permissions set to allow read/write access to the MySQL backup user.

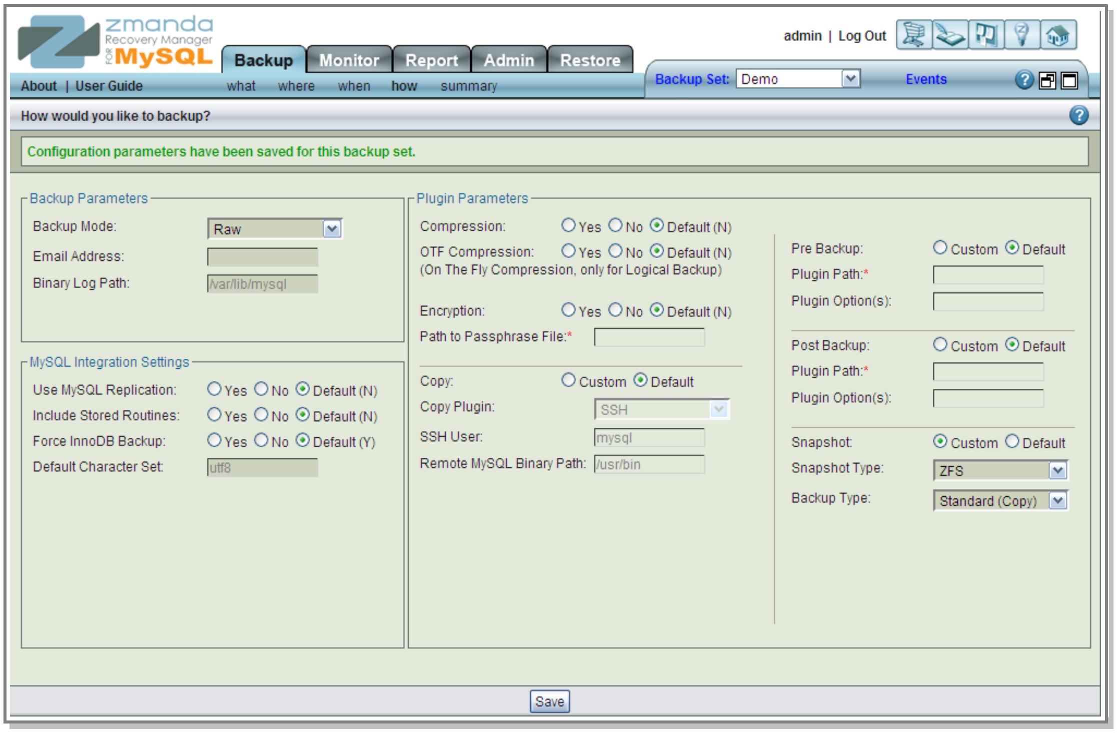
ZMC creates temporary files and directories during backup on both the ZRM server and the MySQL server. The temporary directory specified here should exist on both machines, and should have sufficient space to hold at least one full backup on the MySQL server during backup and on the backup host during restore, plus 10%. How much space this requires in practice depends on the backup method. If `mysqlhotcopy` or the InnoDB Hot Backup plug-in are used, a full backup equals the size of the databases being backed up (+10%). With logical or snapshot backups, only a few gigabytes of temporary space is typically required. The temporary directory must exist on both the ZRM and MySQL server, and have permissions set to allow read/write access to the MySQL backup user.

A retention policy sets the limit of the period for which the backup set will be retained on the hard disk. For example, in Figure 3, the retention policy is set to 7 days and all backup images older than 7 days will be automatically purged.

Configuring ZFS Snapshots

Figure 4 shows the Backup How page. ZRM supports logical-, raw-, and snapshot-based backups of MySQL databases. Currently the following snapshot technologies are supported:

- Solaris ZFS snapshot
- Linux LVM snapshot
- Symantec Veritas VxFS storage checkpoints
- Network Appliance snapshots
- Microsoft Windows VSS (Volume Snapshot Service, also known as Volume Shadow Copy Service) snapshots
- EMC CLARiiON snapshots



Backup Set: Demo

How would you like to backup?

Configuration parameters have been saved for this backup set.

Backup Parameters

Backup Mode: Raw
Email Address:
Binary Log Path: /var/lib/mysql

MySQL Integration Settings

Use MySQL Replication: ○ Yes ○ No ● Default (N)
Include Stored Routines: ○ Yes ○ No ● Default (N)
Force InnoDB Backup: ○ Yes ○ No ● Default (Y)
Default Character Set: utf8

Plugin Parameters

Compression: ○ Yes ○ No ● Default (N)
OTF Compression: ○ Yes ○ No ● Default (N)
(On The Fly Compression, only for Logical Backup)
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: ZFS
Backup Type: Standard (Copy)

Figure 4. Backup How Page Showing Snapshot Options

To take advantage of ZFS snapshots, all MySQL database files (data, log, indexes) belonging to the backup set must be stored in ZFS volumes to ensure consistency. If any of the files are not on ZFS volumes, a raw backup (using `mysqlhotcopy`) or a logical backup (using `mysqldump`) is performed, depending on the storage engines of the tables in each of the databases. Refer to the ZFS documentation mentioned in the “For More Information” section at the end of this document for details on ZFS administration.

Here are some valid scenarios for database storage on ZFS:

- All MySQL data is stored on a single ZFS Volume
- Specific database directories are stored on different volumes

Note: For databases containing InnoDB-based tables, the snapshot can be used only if the database directory, the InnoDB data files, and the InnoDB logs are all on the same ZFS volume.

Choose the preferred method of snapshot backup using the drop-down menu. The Standard (Copy) option specifies that the snapshot should be copied to a specified earlier backup location on the Sun Fire X4540 server. The Quick (No Copy) option specifies that the snapshot itself should be used as a nearline backup.

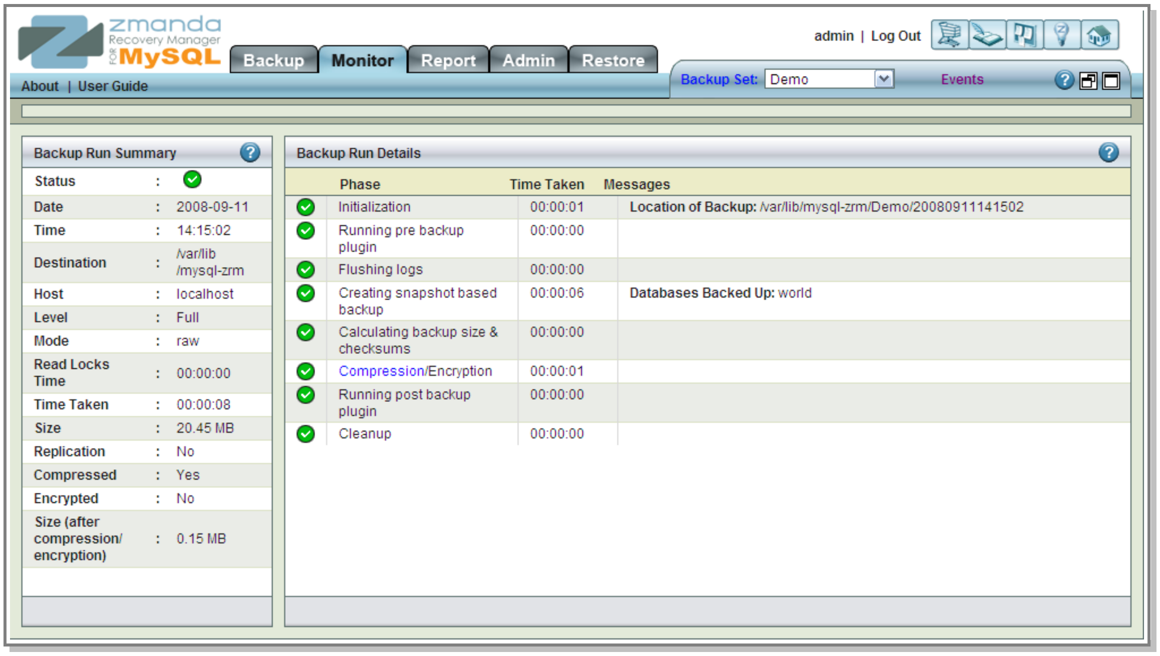
Quick backups are convenient because they provide faster backups and restores, but because they remain on the MySQL server, they do not protect against disk or server failure. Quick snapshot backups can be converted to standard backups stored on the ZRM server using the Convert Backup option available from the Reports menu tab.

To enable backup via a snapshot, Backup Mode (selected from the upper left of the Backup How page) must be Raw. If the current backup mode for the set is Logical, you cannot select a snapshot plug-in from the Backup How page.

The ZFS snapshot does not require additional parameters.

ZRM Monitoring and Reporting

The complexity of backup operations for MySQL databases makes it difficult to estimate how well you meet the business objectives of data protection. The comprehensive monitoring and backup reporting provided by ZRM helps you to ensure your MySQL data is protected well. Figure 5 shows the Monitor page.



Backup Run Summary

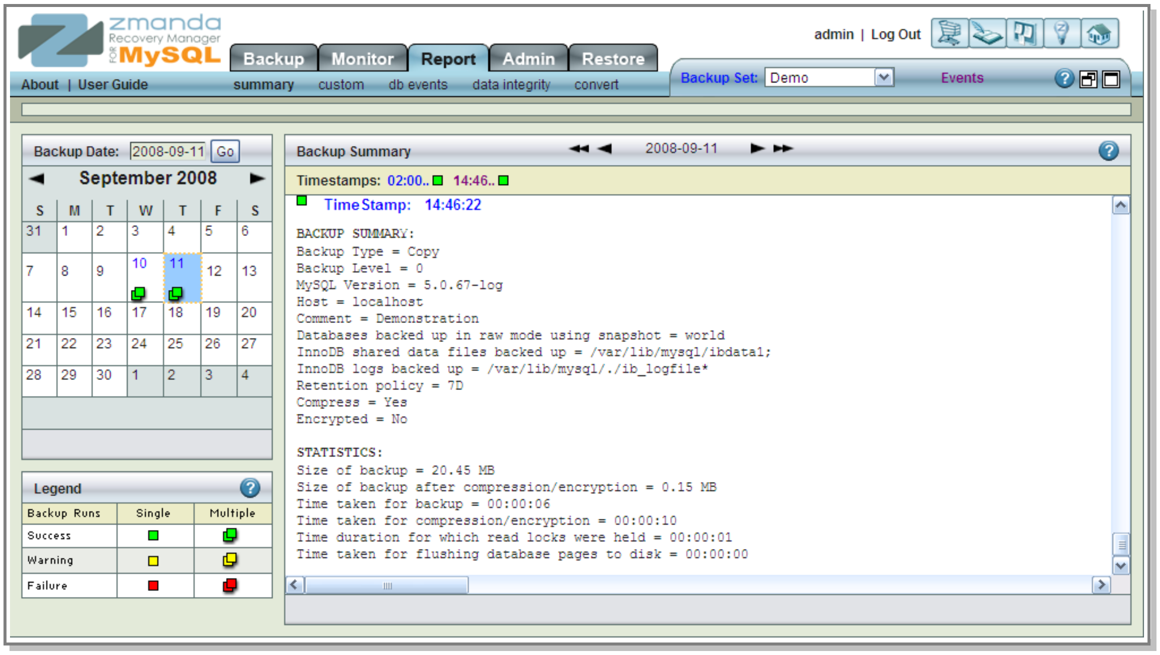
Status	:	✓
Date	:	2008-09-11
Time	:	14:15:02
Destination	:	/var/lib /mysql-zrm
Host	:	localhost
Level	:	Full
Mode	:	raw
Read Locks Time	:	00:00:00
Time Taken	:	00:00:08
Size	:	20.45 MB
Replication	:	No
Compressed	:	Yes
Encrypted	:	No
Size (after compression/ encryption)	:	0.15 MB

Backup Run Details

	Phase	Time Taken	Messages
✓	Initialization	00:00:01	**Location of Backup:** /var/lib/mysql-zrm/Demo/20080911141502
✓	Running pre backup plugin	00:00:00	
✓	Flushing logs	00:00:00	
✓	Creating snapshot based backup	00:00:06	**Databases Backed Up:** world
✓	Calculating backup size & checksums	00:00:00	
✓	Compression/Encryption	00:00:01	
✓	Running post backup plugin	00:00:00	
✓	Cleanup	00:00:00	

Figure 5. Monitor Page Showing Backup Progress

Figure 6 shows the Summary view on the Reports page with an easy-to-understand, visual calendar of your backups for a month at a time with color-coded indicators of success, failure, or warning. Multiple squares indicate several backup runs that day. In case of a problem, you can quickly drill down to a report with details about any backup run.



Backup Date: 2008-09-11 Go

September 2008

S	M	T	W	T	F	S
31	1	2	3	4	5	6
7	8	9	10 🟩	11 🟩	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	1	2	3	4

Legend

Backup Runs	Single	Multiple
Success	🟩	🟩
Warning	🟨	🟨
Failure	🟥	🟥

Backup Summary 2008-09-11

Timestamps: 02:00.. 🟩 14:46.. 🟩

🟩 TimeStamp: 14:46:22

```
BACKUP SUMMARY:
Backup Type = Copy
Backup Level = 0
MySQL Version = 5.0.67-log
Host = localhost
Comment = Demonstration
Databases backed up in raw mode using snapshot = world
InnoDB shared data files backed up = /var/lib/mysql/ibdata1;
InnoDB logs backed up = /var/lib/mysql/./ib_logfile*
Retention policy = 7D
Compress = Yes
Encrypted = No

STATISTICS:
Size of backup = 20.45 MB
Size of backup after compression/encryption = 0.15 MB
Time taken for backup = 00:00:06
Time taken for compression/encryption = 00:00:10
Time duration for which read locks were held = 00:00:01
Time taken for flushing database pages to disk = 00:00:00
```

Figure 6. Calendar View of Recent Backup Reports

ZRM provides nine pre-defined reports, all of which can be easily customized:

- Backup report
- Backup application performance report
- Backup status report
- Backup methods report
- Backup retention policies report
- Backup performance report
- Incremental backup report
- Backup replication report
- Cluster backup report

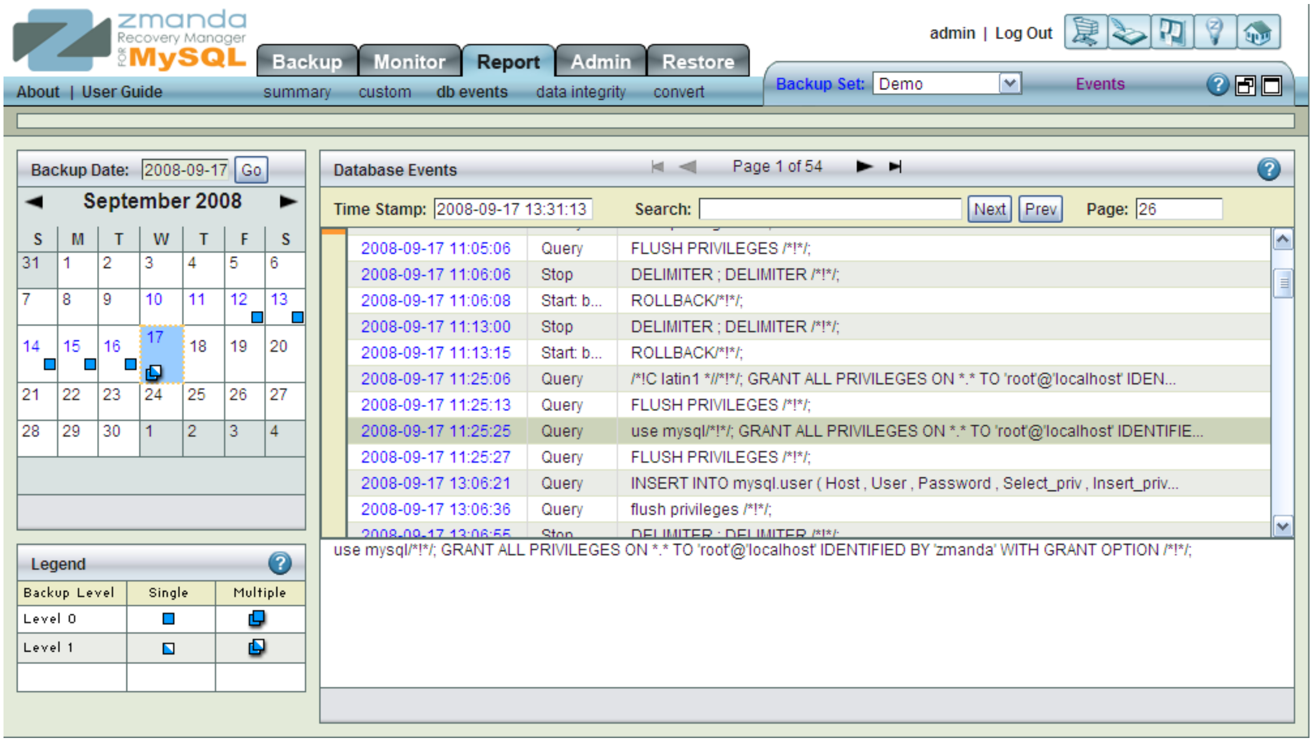
In addition to the pre-defined reports, you can create and save for future use highly customized reports on more than 30 different backup and MySQL parameters. Reports can be generated and delivered in a variety of ways. The output can be either text or HTML. Reports can be delivered as RSS feeds or as email. These options let you integrate ZRM reporting with the monitoring dashboard you are already using.

Database Events Viewer

The Database Events Viewer (Visual Log Analyzer), shown in Figure 7, lets you visually browse the MySQL binary logs used for incremental backup, so you can select exactly what point in time you wish to restore.

The date and time of each database event is shown as an individual record. You can easily scroll through the entries and search the logs with queries. Using the search, you can locate any event that caused harm to your database. You can also jump to a given date and timestamp.

To help you navigate through events, there is a special vertical, orange tool bar to the left of the date and timestamps in the database events list. As you move your cursor up and down this vertical bar, a tool-tip displays the date and time of the events in that relative location. Clicking the tool-tip jumps to the given date and time.



Backup Set: Demo

Backup Date: 2008-09-17 Go

September 2008

S	M	T	W	T	F	S
31	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	1	2	3	4

Backup Level	Single	Multiple
Level 0	■	⧉
Level 1	◪	⧉

Database Events

Time Stamp: 2008-09-17 13:31:13 Search: Next Prev Page: 26

2008-09-17 11:05:06	Query	FLUSH PRIVILEGES /*!*/;
2008-09-17 11:06:06	Stop	DELIMITER ; DELIMITER /*!*/;
2008-09-17 11:06:08	Start: b...	ROLLBACK/*!*/;
2008-09-17 11:13:00	Stop	DELIMITER ; DELIMITER /*!*/;
2008-09-17 11:13:15	Start: b...	ROLLBACK/*!*/;
2008-09-17 11:25:06	Query	/*!C latin1 *//*!*/; GRANT ALL PRIVILEGES ON *.* TO 'root'@'localhost' IDEN...
2008-09-17 11:25:13	Query	FLUSH PRIVILEGES /*!*/;
2008-09-17 11:25:25	Query	use mysql/*!*/; GRANT ALL PRIVILEGES ON *.* TO 'root'@'localhost' IDENTIFIE...
2008-09-17 11:25:27	Query	FLUSH PRIVILEGES /*!*/;
2008-09-17 13:06:21	Query	INSERT INTO mysql.user (Host , User , Password , Select_priv , Insert_priv...
2008-09-17 13:06:36	Query	flush privileges /*!*/;
2008-09-17 13:06:55	Stop	DELIMITER ; DELIMITER /*!*/;

```
use mysql/*!*/; GRANT ALL PRIVILEGES ON *.* TO 'root'@'localhost' IDENTIFIED BY 'zmanda' WITH GRANT OPTION /*!*/;
```

Figure 7. Database Events Viewer

These features let you choose a recovery point right down to the transaction level. You can set the recovery point right from the Database Events Viewer. Just click the date and timestamp link next to the last transaction you wish to recover. ZRM then directs you to the Recover screens, with all the appropriate values filled in to recover the database to that exact date and time.

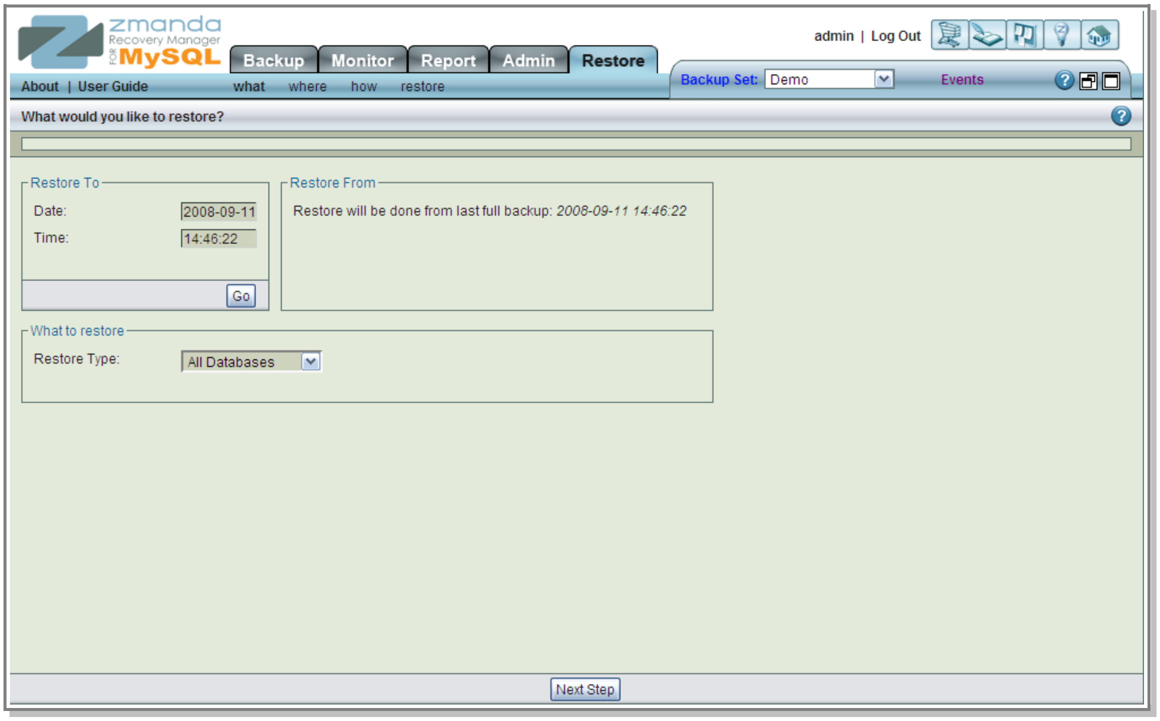
Recovery Using ZMC

The “R” in ZRM stands for Recovery, which of course is the primary objective of ZRM: simple, dependable, and efficient recovery of data.

To restore data, simply use the Restore What page (shown in Figure 8) to select the backup set, the database, or individual tables to recover. You can restore individual tables only if logical backups have been run.

After selecting what to restore from, choose a recovery point objective (RPO). ZRM catalogs all full and incremental backups performed. When you request a point-in-time restore, ZRM automatically finds the full backup and all the incremental backups required to restore your database to the requested RPO.

ZRM allows you to restore to a point in time regardless of whether logical-, raw-, or snapshot-based backup was executed; the only requirement is that you have set up incremental backups as well.



Backup Set: Demo

What would you like to restore?

Restore To

Date: 2008-09-11

Time: 14:46:22

Restore From

Restore will be done from last full backup: *2008-09-11 14:46:22*

What to restore

Restore Type: All Databases

Figure 8. Restore What Page

Depending on how you determined the RPO, you can launch the restore page in several ways:

- Click one of the timestamp links included in any successful backup report.
- Identify the recovery point in the Database Events Viewer and click the timestamp of the given event.
- Open the Restore What page and restore from the last full backup or any other date on which a successful backup was completed.

You can restore to a MySQL server that differs from the original backup source. You can also use restores to instantiate new MySQL replication slaves.

This simplified restore work flow is the same regardless of the storage engine and backup method used. When recovering mission-critical data, having a unified recovery procedure for all MySQL servers helps ensure efficient, dependable, and stress-free recoveries.

Summary

This guide was a quick introduction to configuring the Sun Fire X4540 server as a backup server for Zmanda Recovery Manager 2.2. It provides examples for installing the Zmanda Recovery Manager for MySQL and configuring a zpool and the ZFS file system on the Solaris OS to utilize the Sun Fire X4540 server for storing MySQL backups.

For More Information

Here are additional resources:

- Documentation for Zmanda Recovery Manager from Zmanda Network (requires registration): http://network.zmanda.com/documentationEnterprise.php
    - ZRM datasheet: http://www.zmanda.com/zrm-datasheet.pdf
    - *ZRM: Guide to MySQL Backup & Recovery* and *Backup and Recovery Benchmark for MySQL with ZRM* (requires registration): http://network.zmanda.com/zrm-whitepapers.php
- The following Sun documents at http://docs.sun.com
    - Sun Fire X4540 server document collection: http://docs.sun.com/app/docs/prod/sf.x4540#hic
    - *ZFS Administration Guide* (part of the Solaris 10 System Administration Collection): http://docs.sun.com/app/docs/doc/819-5461
    - *System Administration Guide: IP Services* (part of the Solaris 10 System Administration Collection): http://docs.sun.com/app/docs/doc/816-4554
- MySQL documentation site*:* http://dev.mysql.com/doc/
- *ZFS Best Practices Guide* on the Solaris Internals web site: http://www.solarisinternals.com/wiki/index.php/ZFS_Best_Practices_Guide
- Sun training courses at http://www.sun.com/training/, such as Sun Fire X4540 Server Administration (WET-6179)
- MySQL training courses: http://www.sun.com/software/products/mysql/training.jsp
- MySQL download sites:
    - http://www.sun.com/software/products/mysql/getit.jsp
    - http://dev.mysql.com/downloads/
- Discussions:
    - Zmanda forums: http://forums.zmanda.com/
    - MySQL forums: http://forums.mysql.com/
    - Sun Storage forums: http://forum.java.sun.com/category.jspa?categoryID=66
    - Sun Hardware - Servers forums: http://forums.sun.com/forum.jspa?forumID=830

- Wikis:
  - Sun BluePrints™ wiki (http://wikis.sun.com/display/BluePrints/Main, especially the following:
    - Storage BluePrints page: http://wikis.sun.com/display/BluePrints/Storage+BluePrints
    - Sun BluePrints article: *Optimize MySQL Server on Sun x64 Servers and Storage:* http://wikis.sun.com/display/BluePrints/Optimize+MySQL+Server+on+Sun+x64+Servers+and+Storage
  - Sun Storage Administration wiki: http://wikis.sun.com/display/StorageAdmin/Home
  - BigAdmin Storage Tech Tips wiki: http://wikis.sun.com/display/BigAdmin/Storage+Tech+Tips

  Other resources:
  - Zmanda Recovery Manager for MySQL web site: http://mysqlbackup.zmanda.com/
  - Sun Fire X4540 server web site: http://www.sun.com/servers/x64/x4540/
  - MySQL site on sun.com: http://www.sun.com/software/products/mysql/index.jsp
  - mysql.com web site: http://www.mysql.com/
  - BigAdmin article: *Sun Fire X4540 Server as Backup Server for Zmanda's Amanda Enterprise 2.6 Software*: http://www.sun.com/bigadmin/features/articles/zmanda_sfx4540.jsp
  - BigAdmin Storage Resource Collection: http://www.sun.com/bigadmin/collections/storage.html
  - Sun Storage web site: http://www.sun.com/storagetek/index.jsp
  - Storage Stop Blog: http://blogs.sun.com/storage
- Support:
  - Sun resources:
    - Register your Sun gear: https://inventory.sun.com/inventory/
    - Services: http://www.sun.com/service/index.jsp
    - SunSolve℠ Online: http://sunsolve.sun.com
  - Community system administration experts: http://www.sun.com/bigadmin/content/communityexperts/
- Events of interest to users of Sun products:
  - Sun and MySQL Around the World--Meetup Mashup: http://www.sun.com/events/mysqltour/index.jsp
  - Find a MySQL Meetup Group near you: http://mysql.meetup.com/
  - Worldwide developer events and Sun Tech Days: http://developers.sun.com/events/
  - Current events: http://www.sun.com/events/index.jsp

Licensing Information

Unless otherwise specified, the use of this software is authorized pursuant to the terms of the license found at http://www.sun.com/bigadmin/common/berkeley_license.html.