zmanda
Open Source Data Protection

How to backup a remote MySQL server with ZRM over the Internet

White paper

"As MySQL gains widespread adoption and moves more broadly into the enterprise, ZRM for MySQL addresses the growing need among database administrators to protect their digital assets with a comprehensive backup and recovery solution. Zmanda is a valued member of MySQL's ecosystem, and their open source backup solutions are an excellent complement to MySQL's open source database offerings."

Marten Mickos,
CEO of MySQL AB.

“Zmanda is leveraging their expertise to provide solutions that address data backup issues throughout the enterprise. Backup and recovery software that meets the rigorous requirements of enterprise-class MySQL implementations has been sorely missed in the MySQL market. ZRM for MySQL fills this need with a low-cost package that will strengthen MySQL as a choice for business-critical enterprise applications”
Joshua Greenbaum,
Principal of Enterprise Applications Consulting.

Abstract

This document describes what to consider and how to configure ZRM for secure backup and recovery of remote MySQL server for common use cases including logical and raw backups.

Table of Contents

Please send your comments about this white paper to ***feedback@zmanda.com***

ZRM for MySQL overview

The MySQL database has become the world's most popular open source database because of its consistent fast performance, high reliability and ease of use. You might be using MySQL database for your online forums and wiki located at managed hosting provider, or you might be using it in a remote office for tracking bugs with Bugzilla, or you might just be thinking about developing some new Web 2.0 application that uses MySQL. Either way, if you value information stored in your MySQL database, you will need to ensure successful, secure and consistent backups of MySQL with minimal impact on the database application. Make sure that your backup solution provides the most efficient use of network, server and storage resources.

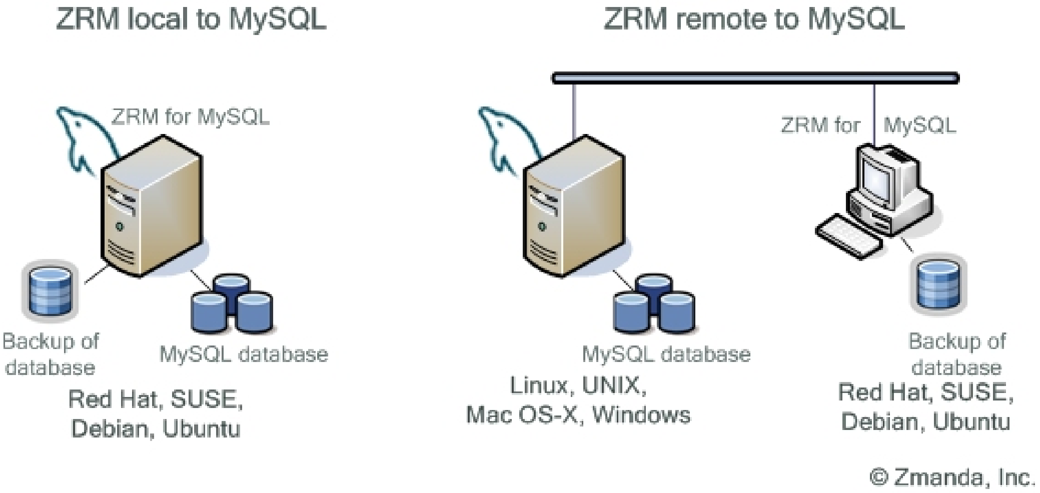
If you are looking for a solution that simplifies your life by providing an easy to use yet flexible and robust backup and recovery for MySQL, Zmanda Recovery Manager (ZRM) will provide you with the following functionality:

- Backup live database with minimal impact on users.
- In addition to immediate backup, you can schedule recurring full and incremental backups of your MySQL database. A scheduling plug-in allows you to postpone backups based on thresholds you define. For example, you can postpone a backup for an hour if more than 50 users access the database. Additional plug-ins make it easy for you to perform certain actions on MySQL before or after a scheduled backup. For example, if you are using an advanced RAID array for MySQL, with a pre-backup plug-in you can perform a hardware snapshot of MySQL and use snapshots for very quick backup and recovery.
- You can choose between more flexible logical or faster raw backups of your database.
- Perform backups that are optimal for your storage engine and your MySQL configuration.
- Configure on-the-fly compression and encryption of your backups to meet your storage and security needs.
- You can back up your remote MySQL database through a firewall.
- Get e-mail notification about the status of your backups and receive backup reports via RSS feeds.
- You can provide comments about your backup that will make it easier for you later to recover data to a particular point in time, for example, just before you upgraded MySQL server.
- Monitor and browse your 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 or security reasons

ZRM for MySQL is open source software and does not use any proprietary data formats. As a matter of fact, in case of emergency you can recover ZRM backups of MySQL even without ZRM being installed.

ZRM supports all major Linux distributions, versions of MySQL from 4.x to 5.x and all storage engines. ZRM scales up for large amounts of data per SQL server and large number of transaction rates. ZMR also scales out by being able to backup and recover multiple MySQL servers from a single workstation.

ZRM provides a rich and easy to script command line interface (CLI) that MySQL DBAs will find intuitive to use. Backup and recovery functionality is the same for either local or remote configurations:



For remote configuration you can backup MySQL running on any operating system that provides support for MySQL client on Linux. For example, you can backup and recover a MySQL database running on a Windows server from your Linux workstation where ZRM is installed.

This document describes in detail what to consider and how to configure ZRM for secure backup and recovery of a remote MySQL server for common use cases including logical and raw backups.

Additional information about ZRM for MySQL is available at http://www.zmanda.com/backup-mysql.html. User documentation is available on ZRM for MySQL wiki at http://mysqlbackup.zmanda.com/. Community edition of ZRM can be downloaded from http://www.zmanda.com/downloads.html

Considerations for remote backup of MySQL

For any backup of a database the primary considerations are the consistency of the backup and impact on users and applications. A backup of the remote MySQL, however, has extra challenges related to:

- network usage
- security, and
- flexibility of recoverability of MySQL data to a different host.

The last point could be important when you don't have full control of your MySQL environment and you want to have an option to recover your data to a different managed hosting provider with a different version of MySQL server or a different operating system.

Incremental backups significantly reduce backup window and network usage because only the changes since the last full or the last incremental backup are moved over the wire. ZRM makes it easy for you to recover your data from incremental backups even if you have to use multiple incremental backup images to get your data back to a particular point in time. The incremental backups require MySQL binary logs to be enabled, but according to MySQL documentation, enabling of binary logs will result in performance hit of less than 1%.

Logical backups provide more flexibility for recovery because the backup file is a text file that contains all MySQL statements to re-create both the database schema and contents. Logical backup works for all storage engines except the NDB engine used for MySQL clustering. The biggest advantage of logical backup is flexibility for a database recovery. You can restore logical backups of MySQL to other architecture and even to another database. Transportability of logical ZRM backup images makes ZRM a convenient tool for migration. For example, you can move your MySQL data:

- From MySQL on Solaris to MySQL on Linux
- From one storage engine to another
- From a 32-bit server to a 64-bit server
- From one managed hosting provider to your datacenter or another provider with a different MySQL configuration

Of course, there is price to pay for such flexible recovery. Restoring data from the logical backups could take a long time since you have to read and replay all MySQL statements. Another downside is that it could be difficult to predict the size of your logical backup. Depending on the type of data and your database schema, the size of the logical backup could be larger that the database itself. One remedy is that since logical backup is basically a text file you can usually get a decent compression.

Raw backup provides a consistent copy of a database with your backup being a binary file. The advantages of raw backups over logical backups are:

- Backup and especially recovers are much faster. For example, it not unusual to see that for the very same database with size of 4-5 GB, the raw backup is 5 times faster than the logical backup, and recovery of the raw backup image is 20 times faster than recovery of the logical image.
- You will always know the exact size of your backup since it is just a copy of a database.
- It provides better scalability which could be important if your MySQL database is rather large, for example, 10-20 GB or more.

Raw backups can be recovered ONLY to the same version of MySQL server on the same operating system as the original data. It means that your chances to recover raw backup images of MySQL to another managed hosting provider are not very high and you should take that into consideration when choosing raw vs. logical backup.

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

One of the important considerations for remote backup of MySQL is the decision about what type of connection to establish between ZRM and remote MySQL server. ZRM provides a plug-in for socket based connection and another plug-in for SSH-based connection. Flexible architecture of ZRM allow users to write their own plug-ins.

As the name suggests, socket copy plug-in establishes a socket that provides communication between ZRM and MySQL over IP-based network. Socket copy plug-in requires the ***xinetd*** service to be running on the MySQL server and the default port 25300 to be open. If needed, a backup administrator can change the port. Socket copy plug-in is not secure and should be used only when security is not of concern or when security is established by other means, for example, when you have a VPN connection between ZRM and your remote MySQL server.

The SSH copy plug-in provides a secure channel between ZRM and the remote MySQL server. It uses public-key cryptography to authenticate the remote MySQL server and the backup user running ZRM. The SSH plug-in requires the standard TCP port 22 to be open and SSH daemon to be running. The SSH copy plug-in is best suited when ensuring security of backup data is important. Since a SSH connection takes extra CPU cycles for encryption, there might be a hit on backup performance comparing to backup with socket connection.

The following table summarizes considerations when choosing socket vs. SSH copy plug-in for remote backup of MySQL.

Remote connection plug-in	Port used	Security	Relative performance	Installation comments
SSH copy	22 (fixed)	Provides strong authentication and encryption for moving backup data over the wire	Lower performance and depends on MySQL server memory, CPU resources and available bandwidth.	Often times you might have SSH connection to a remote MySQL server already established for reasons others than backup and recovery. Otherwise you have to establish SSH connection between ZRM and MySQL server.
Socket copy	25300 (could be changed)	Backup data over the wire is not secure	Higher performance	Additional software should be installed on MySQL server. For example, for Enterprise version of ZRM 1.1 use MySQL-zrm-enterprise-socket-server-1.1-1.noarch.rpm. For Community version, see downloads page for exact package.

With all its rich functionality, ZRM for MySQL is just a tool for implementing backup and recovery strategy that is optimal for your data protection needs. ZRM is robust and easy to use, but depending on your own implementation of remote MySQL server and your specific requirements for backup and recovery, you should consider all trade-offs associated with each operational option provided by ZRM for MySQL.

The rest of the document provides an example of using ZRM for MySQL for several common use cases.

Overview of the operational environment used in this document

To illustrate ZRM functionality for backup and recovery we used “real” Netflix databases available as part of Netflix Prize project targeted on improving the accuracy of movie recommendations based on users’ renting history and preferences.

1. ZRM Server “quartz.zmanda.com”
This is a Linux RHEL 4 server with the following components installed:

- MySQL5 Client
- MySQL-ZRM
- Perl-DBI module

This server (it could be a workstation) needs to have enough disk storage and appropriate fault tolerance to keep multiple full and incremental backup images of the databases and to ensure the safety and availability of the backup data.

2. MySQL5 server “DB1.zmanda.com”
This is a Linux SLES 10 server that has two functional databases running that we want to backup. The first one is an InnoDB database called "moviesinnodb" and the second one is a MyISAM database called "moviesmyisam". Both of these databases are about 115 GB each. In the first use case we will be performing a logical backup of the "moviesinnodb" database as shown in **Fig. 1** over socket connection through a firewall:

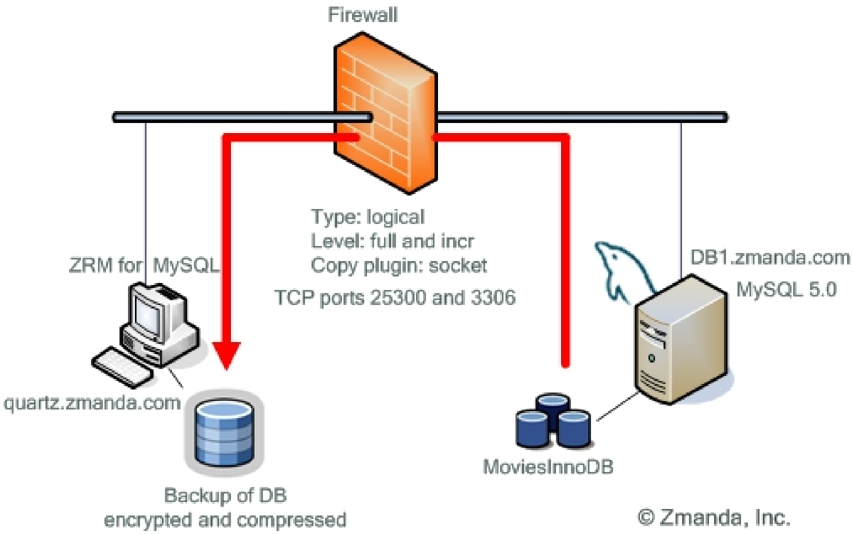


Fig. 1 Logical backup with socket connection

When logical backup arrives to “quartz.zmanda.com” over the socket connection, ZRM will encrypt and compress the backup image, so data at rest will be secured even though it was not secure in transit from MySQL server to ZRM.

In the second use case we will perform a raw backup of the "moviesmyisam" database as shown in **Fig. 2:**

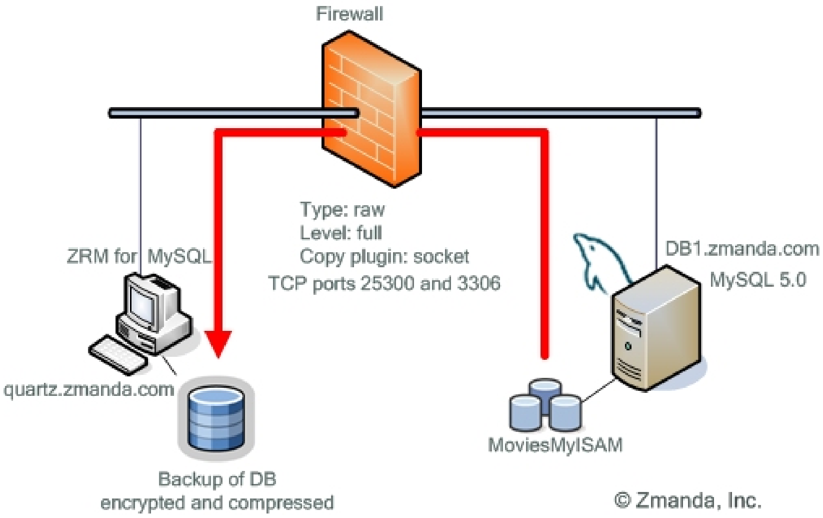


Fig. 2 Raw backup with socket connection.

Similarly to the first use case, when raw backup data arrives to "quartz.zmanda.com" over the socket connection, ZRM will encrypt and compress the backup image, so data at rest will be secured even though it was not secure in transit from MySQL server to ZRM.

The high performance socket copy plug-in offered by ZRM for MySQL will enable us to do both remote raw and remote logical backups with high efficiency.

3. MySQL5 server "DB2.zmanda.com"
This is a Linux SLES 10 server with one functional database running. This is a MyISAM database called "moviesmyisam" and is 115GB in size. We will be performing a raw backup of the "moviesmyisam" database and will be using the SSH plug-in as shown in **Fig. 3:**

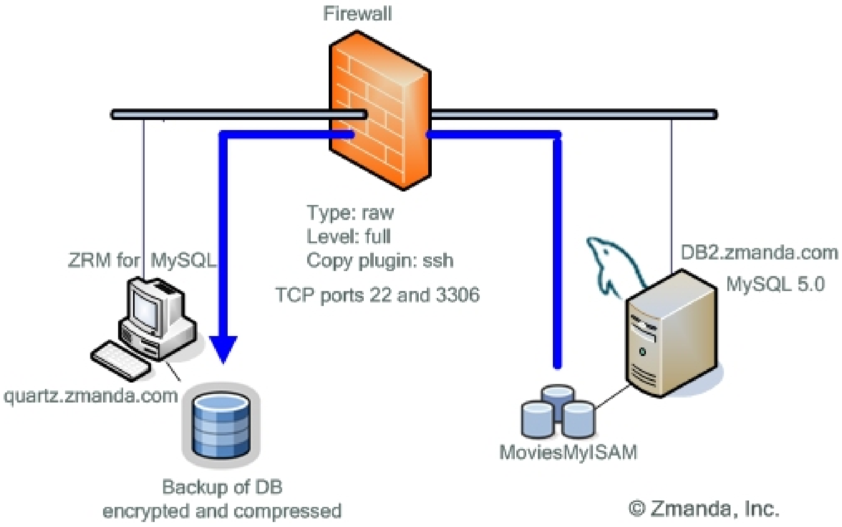


Fig. 3 Raw backup with SSH connection.

When raw backup data arrives to “quartz.zmanda.com” over SSH connection, ZRM will encrypt and compress the backup image, so in this scenario data is secure in transit and at rest.

An example of using ZRM for MySQL for remote backup with socket- and SSH copy plug-ins.

Here is the roadmap of what we cover in the rest of this document:

1. Setup MySQL-ZRM on the backup server “quartz.zmanda.com” and create a backup set called DailyBackupDB1 to back up two databases on “db1.zmanda.com”
2. Prepare db1.zmanda.com to be backed up using the socket copy plug-in
3. Configure firewall for backups between “quartz.zmanda.com” and “db1.zmanda.com”
4. Setup ZRM on the backup server and create a backup set called DailyBackupDB2 to backup MoviesMyISAM database on “db2.zmanda.com”
5. Prepare db2.zmanda.com for back up using the SSH copy plug-in
6. Configure firewall for backups between “quartz.zmanda.com” and “db2.zmanda.com”
7. Perform the backup of both backup sets DailyBackupDB1 and DailyBackupDB2 from the command line
8. Verify backup and check performance statistics
9. Set up Automated backups
10. Recover MySQL data

1. Setting Up ZRM on “quartz.zmanda.com” to backup (moviesinnodb / moviesmyisam) databases on running on “db1.zmanda.com”

a. Install DBI Module

```
[root@quartz ~]# rpm -ivh perl-DBI-1.40-5.i386.rpm
warning: perl-DBI-1.40-5.i386.rpm: V3 DSA signature: NOKEY, key ID db42a60e
Preparing...                ########################################### [100%]
1:perl-DBI                ########################################### [100%]
```

b. Install MySQL5 Client

```
[root@quartz rhelmysql4]# rpm -ivh MySQL-client-standard-5.0.27-0.rhel4.i386.rpm
  warning: MySQL-client-standard-5.0.27-0.rhel4.i386.rpm: V3 DSA signature: NOKEY, key ID 5072e1f5
Preparing...                ########################################### [100%]
1:MySQL-client-standard  ########################################### [100%]
```

c. Install MySQL-ZRM (Download this package from http://network.zmanda.com)

```
[root@quartz latest]# rpm -ivh MySQL-zrm-enterprise-1.1-1.noarch.rpm
warning: MySQL-zrm-enterprise-1.1-1.noarch.rpm: V3 DSA signature: NOKEY, key ID 3c5d1c92
Preparing...                ########################################### [100%]
1:MySQL-zrm-enterprise           ########################################### [100%]
```

d. Create directory and copy default configuration file to create the first ZRM backup set “DailyBackupDB1”

```
[root@quartz mysql-zrm]# mkdir /etc/mysql-zrm/DailyBackupDB1
[root@quartz mysql-zrm]# cp /etc/mysql-zrm/mysql-zrm.conf /etc/mysql-zrm/DailyBackupDB1
```

e. Edit the /etc/mysql-zrm/DailyBackupDB1/mysql-zrm.conf and make the changes to the lines below:

```
compress=1
compress-plugin=/usr/bin/gzip
encrypt=1
encrypt-plugin="/usr/share/mysql-zrm/plugins/encrypt.pl"
databases=moviesinnodb moviesmyisam
user="backup-user"
password="zrmbackup"
host="db1.zmanda.com"
mailto="mysqldba@company.com"
copy-plugin=/usr/share/mysql-zrm/plugins/socket-copy.pl
```

f. Create file that contains the passphrase used for backup encryption

```
[root@quartz /]# touch /etc/mysql-zrm/.passphrase
[root@quartz /]# echo  encryptpass > /etc/mysql-zrm/.passphrase
[root@quartz /]# chmod 700 /etc/mysql-zrm/.passphrase
```

g. In order for encryption to work the /root/.gnupg directory must me created

```
[root@quartz ~]# mkdir /root/.gnupg/
[root@quartz ~]# chmod 600 /root/.gnupg/
```

2. **Prepare “db1.zmanda.com” for backup by ZRM**

a. Login to the MySQL server db1.zmanda.com

b. Connect to MySQL as root user

```
db1:~ # mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2 to server version: 5.0.18
Type 'help;' or '\h' for help. Type '\c' to clear the buffer.
mysql>
```

c. Give Privileges to "backup-user" to connect from remote host "quartz.zamanda.com"

```
mysql> GRANT LOCK TABLES, SELECT, FILE, CREATE, DROP, INDEX, SHUTDOWN,
ALTER, INSERT, SUPER, RELOAD ON *.* TO 'backup-  user'@'quartz.zmanda.com' IDENTIFIED
BY 'zrmbackup';
Query OK, 0 rows affected (0.00 sec)
```

d. Give Privileges to "backup-user" to connect locally from "db1.zmanda.com"

```
mysql> GRANT LOCK TABLES, SELECT, FILE, CREATE, DROP, INDEX, SHUTDOWN,
ALTER, INSERT, SUPER, RELOAD ON *.* TO 'backup-user'@'db1.zmanda.com' IDENTIFIED BY
'zrmbackup';
Query OK, 0 rows affected (0.01 sec)
```

e. Flush Privileges

```
mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.00 sec)
mysql> quit
Bye
```

f. Enable binary logging on the MySQL server by editing the /etc/my.cnf file and specifying the “log-bin” parameter under the “[mysqld]” section. This will enable incremental backups of the InnoDB database “moviesinnodb” .

g. Restart MySQL server

```
db1:~ # service mysql restart
Restarting service MySQL
Shutting down service MySQL                    done
Starting service MySQL                         done
```

h. Install the ZRM server socket package to create the appropriate ***xinetd*** service and files on the database server

db1:~ # rpm -ivh MySQL-zrm-enterprise-socket-server-1.1-1.noarch.rpm (Download this package from http://network.zmanda.com)

```
warning: MySQL-zrm-enterprise-socket-server-1.1-1.noarch.rpm: Header V3 DSA signature: NOKEY, key ID 3c5d1c92
Preparing...                ############################################ [100%]
1:MySQL-zrm-enterprise-so############################################ [100%]
Nov 20 2006 09:28:37: Reloading xinetd configuration... Reload INET services (xinetd).
```

3. Configure firewall for backups between “quartz.zmanda.com” and “db1.zmanda.com”

Ports 25300 and 3306 should be open on host **“db1.zmanda.com”** for TCP traffic originating from **“quartz.zmanda.com”.** Port 25300 is used by the MySQL ZRM socket plug-in and port 3306 is used by the MySQL database.

4. Setting Up ZRM on “quartz.zmanda.com” to backup the (moviesmyisam) database running on “db2.zmanda.com”

a. Create directory and copy default configuration file to create the first ZRM backup set “DailyBackupDB2”

```
[root@quartz mysql-zrm]# mkdir /etc/mysql-zrm/DailyBackupDB2
[root@quartz mysql-zrm]# cp /etc/mysql-zrm/mysql-zrm.conf /etc/mysql-zrm/DailyBackupDB2
```

b. Edit the /etc/mysql-zrm/DailyBackupDB2/mysql-zrm.conf and make the changes to the lines below

```
compress=1
compress-plugin=/usr/bin/gzip
encrypt=1
encrypt-plugin="/usr/share/mysql-zrm/plugins/encrypt.pl"
database=”moviesmyisam”
user="backup-user"
password="zrmbackup"
host="db2.zmanda.com"
mailto="mysqldba@company.com"
copy-plugin=/usr/share/mysql-zrm/plugins/ssh-copy.pl
ssh-user="mysql"
```

c. Create ssh keys for user “root” on “quartz.zmanda.com”

```
[root@quartz /]# mkdir /root/.ssh
[root@quartz /]# chmod 700 /root/.ssh
[root@quartz /]# ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
42:25:b1:dd:c7:b8:a7:11:24:a6:22:b0:39:d5:29:8c root@quartz
```

5. Set up firewall for backups between “quartz.zmanda.com” and “db2.zmanda.com”

Ports 3306 and 22 need to be open on host **“db2.zmanda.com”** for TCP connections originating from host **“quartz.zmanda.com”**. Port 3306 is used by the MySQL database and port 22 is used by the ssh daemon.

6. Prepare “db2.zmanda.com” for backup by ZRM

a. Login to the MySQL server db2.zmanda.com

b. Connect to MySQL as root user

```
db2:~ # mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 5 to server version: 5.0.18
Type 'help;' or '\h' for help. Type '\c' to clear the buffer.
mysql>
```

c. Give Privileges to "backup-user" to connect from remote host "quartz.zamanda.com"

```
mysql> GRANT LOCK TABLES, SELECT, FILE, CREATE, DROP, INDEX, SHUTDOWN, ALTER, INSERT, SUPER, RELOAD ON *.* TO 'backup-user'@'quartz.zmanda.com' IDENTIFIED BY 'zrmbackup';
Query OK, 0 rows affected (0.01 sec)
```

d. Give Privileges to "backup-user" to connect locally from "db2.zmanda.com"

```
mysql> GRANT LOCK TABLES, SELECT, FILE, CREATE, DROP, INDEX, SHUTDOWN, ALTER, INSERT, SUPER, RELOAD ON *.* TO 'backup-user'@'db2.zmanda.com' IDENTIFIED BY 'zrmbackup';
Query OK, 0 rows affected (0.00 sec)
```

e. Flush Privileges

```
mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.00 sec)
mysql> quit
Bye
```

f. The default “mysql” user will be used to connect to "db2.zmanda.com" and needs to be created if doesn't already exist.

g. Create the .ssh directory where the public ssh key will be stored. This user is needed to establish a connection from the MySQL-ZRM server “quartz.zmanda.com”.

```
b2:~ # mkdir /var/lib/mysql/.ssh
db2:~ # chown  mysql  /var/lib/mysql/.ssh
db2:~ # chmod 700 /var/lib/mysql/.ssh
```

h. Copy the contents of /root/.ssh/id_rsa.pub file located on quartz.zmanda.com to the file “authorized_keys” on db2.zmanda.com

On server “quartz.zmanda.com” run:

```
[root@quartz .ssh]# scp /root/.ssh/id_rsa.pub root@db2.zmanda.com:/var/lib/mysql/.ssh/authorized_keys
The authenticity of host 'db2.zmanda.com (192.168.10.140)' can't be established.
```

```
RSA key fingerprint is ff:c0:41:4f:30:2b:97:30:49:4f:0e:e2:47:93:f5:a7.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'db2.zmanda.com,192.168.10.140' (RSA) to the list of known hosts.
Password:
id_rsa.pub                                        100%  221     0.2KB/s   00:00
```

On server “db2.zmanda.com” run:

```
db2:/ # chmod 600 /var/lib/mysql/.ssh/authorized_keys
db2:/ # chown mysql /var/lib/mysql/.ssh/authorized_keys
```

7. **Execute a test backup of databases on both “db1.zmanda.com” and “db2.zmanda.com”**

a. Execute backup of databases (moviesinnodb / moviesmyisam) on server “db1.zmanda.com”

```
[root@quartz /]# mysql-zrm-scheduler --backup-set DailyBackupDB1 --backup-level 0 --now
Logging to /var/log/mysql-zrm/mysql-zrm-scheduler.log
INFO: ZRM for MySQL Enteprise Edition - version 1.1
INFO: backup-set=DailyBackupDB1
INFO: backup-date=20061127120644
INFO: host=db1.zmanda.com
INFO: backup-date-epoch=1164658004
INFO: mysql-zrm-version=ZRM for MySQL Enteprise Edition - version 1.1
INFO: mysql-version=5.0.18-log
INFO: backup-directory=/var/lib/mysql-zrm/DailyBackupDB1/20061127120644
INFO: backup-level=0
INFO: raw-databases=moviesmyisam
WARNING: The database(s) moviesinnodb  will be backed up in logical mode since they contain tables that use a transactional engine.
INFO: logical-databases=moviesinnodb
INFO: next-binlog=mysql-bin.000014
INFO: last-backup=/var/lib/mysql-zrm/DailyBackupDB1/20061127101552
INFO: /var/lib/mysql-zrm/DailyBackupDB1/20061127120644/backup.sql=0e4e735486c81fc44b0455b04bad8a6c
INFO: /var/lib/mysql-zrm/DailyBackupDB1/20061127120644/moviesmyisam/MovieID.MYI=85da29d2d3ddbb813bbb07409b5df3c8
INFO: /var/lib/mysql-zrm/DailyBackupDB1/20061127120644/moviesmyisam/MovieID.frm=7f1b3cafe572643b871b58350d8a00f5
INFO: /var/lib/mysql-zrm/DailyBackupDB1/20061127120644/moviesmyisam/MovieRatings.MYD=8aea114faf8da82447ed4a715d21199a
INFO: /var/lib/mysql-zrm/DailyBackupDB1/20061127120644/moviesmyisam/MovieRatings.MYI=0afb64f33533302688d8b2d7b5de8a68
INFO: /var/lib/mysql-zrm/DailyBackupDB1/20061127120644/moviesmyisam/MovieID.MYD=de4dc815e1828a12b00a4299f4c368fb
INFO: /var/lib/mysql-zrm/DailyBackupDB1/20061127120644/moviesmyisam/db.opt=b8da794f1097aa20b8ee2653b6eb8ef4
INFO: /var/lib/mysql-zrm/DailyBackupDB1/20061127120644/moviesmyisam/MovieRatings.frm=92eb4ca8743e220ac15d14ad485fc21a
INFO: backup-size=261.84 MB
INFO: compress=/usr/bin/gzip
INFO: encrypt=/usr/share/mysql-zrm/plugins/encrypt.pl
INFO: decrypt-option=-d
INFO: backup-size-compressed=69.43 MB
INFO: read-locks-time=00:03:00
INFO: flush-logs-time=00:00:01
INFO: compress-encrypt-time=00:03:09
INFO: backup-time=00:06:28
INFO: backup-status=Backup succeeded
INFO: Backup succeeded
/usr/bin/mysql-zrm started successfully
```

b. Execute backup of database (moviesinnodb) on server “db2.zmanda.com”

```
[root@quartz /]# mysql-zrm-scheduler --backup-set DailyBackupDB2 --backup-level 0 --now
Logging to /var/log/mysql-zrm/mysql-zrm-scheduler.log
INFO: ZRM for MySQL Enteprise Edition - version 1.1
WARNING: Binary logging is off. Incremental backups cannot be done.
INFO: backup-set=DailyBackupDB2
INFO: backup-date=20061127121543
INFO: host=db2.zmanda.com
INFO: backup-date-epoch=1164658543
```

```
INFO: mysql-zrm-version=ZRM for MySQL Enteprise Edition - version 1.1
INFO: mysql-version=5.0.18
INFO: backup-directory=/var/lib/mysql-zrm/DailyBackupDB2/20061127121543
INFO: backup-level=0
INFO: raw-databases=moviesmyisam
INFO: last-backup=/var/lib/mysql-zrm/DailyBackupDB2/20061127115257
INFO: /var/lib/mysql-
zrm/DailyBackupDB2/20061127121543/moviesmyisam/MovieRatings.frm=92eb4ca8743e220ac15d14ad485fc21a
INFO: /var/lib/mysql-zrm/DailyBackupDB2/20061127121543/moviesmyisam/db.opt=b8da794f1097aa20b8ee2653b6eb8ef4
INFO: /var/lib/mysql-
zrm/DailyBackupDB2/20061127121543/moviesmyisam/MovieID.MYI=6ad746291455db9a3ce026530dbcdd38
INFO: /var/lib/mysql-
zrm/DailyBackupDB2/20061127121543/moviesmyisam/MovieID.frm=7f1b3cafe572643b871b58350d8a00f5
INFO: /var/lib/mysql-
zrm/DailyBackupDB2/20061127121543/moviesmyisam/MovieRatings.MYD=8aea114faf8da82447ed4a715d21199a
INFO: /var/lib/mysql-
zrm/DailyBackupDB2/20061127121543/moviesmyisam/MovieRatings.MYI=b3d6c79516a7b32da764824b86b8f251
INFO: /var/lib/mysql-
zrm/DailyBackupDB2/20061127121543/moviesmyisam/MovieID.MYD=de4dc815e1828a12b00a4299f4c368fb
INFO: backup-size=115.47 MB
INFO: compress=/usr/bin/gzip
INFO: encrypt=/usr/share/mysql-zrm/plugins/encrypt.pl
INFO: decrypt-option=-d
INFO: backup-size-compressed=35.41 MB
INFO: read-locks-time=00:02:11
INFO: flush-logs-time=00:00:00
INFO: compress-encrypt-time=00:01:23
INFO: backup-time=00:03:39
INFO: backup-status=Backup succeeded
INFO: Backup succeeded
/usr/bin/mysql-zrm started successfully
```

8. Verify both backups and check performance stats

a. Execute this command to verify the integrity of the DailyBackupDB1 backup

```
[root@quartz DailyBackupDB2]# mysql-zrm --action verify-backup --backup-set
DailyBackupDB1
INFO: ZRM for MySQL Enterprise Edition - version 1.1
INFO: Verification successful
```

b. Execute this command to verify the integrity of the DailyBackupDB2 backup

```
[root@quartz DailyBackupDB2]# mysql-zrm --action verify-backup --backup-set
DailyBackupDB2
INFO: ZRM for MySQL Enterprise Edition - version 1.1
INFO: Verification successful
```

c. Execute this command to see the performance statistics for completed backups

```
[root@quartz ~]# mysql-zrm-reporter --show backup-performance-info
backup_set      backup_date                      backup_level  backup_siz   backup_siz_compr  backup_time   compress_encrypt_time
----------------------------------------------------------------------------------------------------------------------------------
DailyBackupDB2  Tue 28 Nov 2006 11:54:45 AM PST   0             115.47 MB   35.41 MB                     00:05:26       00:01:26
DailyBackupDB1  Tue 28 Nov 2006 11:31:07 AM PST   0             261.84 MB   69.43 MB                     00:06:55       00:02:57
```

9. Setup Automated backups using ZRM scheduler

a. Configure automated Full and Incremental backups for DailyBackupDB1. Incremental Mon-Fri at 1:00am and Full on Saturday 1:00am.

```
[root@quartz DailyBackupDB2]# mysql-zrm-scheduler --add --interval weekly --start-time
1:00 --day-of-week 1-5 --backup-level 1 --backup-set DailyBackupDB1
Logging to /var/log/mysql-zrm/mysql-zrm-scheduler.log
```

```
DONE
```

```
[root@quartz DailyBackupDB2]# mysql-zrm-scheduler --add --interval weekly --start-time 1:00 --day-of-week 6 --backup-level 0 --backup-set DailyBackupDB1
Logging to /var/log/mysql-zrm/mysql-zrm-scheduler.log
DONE
```

b. Configure automated Full backups for DailyBackupDB2. Full backups on Mon-Fri 1:00am.

```
[root@quartz DailyBackupDB2]# mysql-zrm-scheduler --add --interval weekly --start-time 1:00 --day-of-week 1-5 --backup-level 0 --backup-set DailyBackupDB2
Logging to /var/log/mysql-zrm/mysql-zrm-scheduler.log
DONE
```

c. List all the scheduled backups on the ZRM backup server

```
root@quartz DailyBackupDB2]#  mysql-zrm-scheduler  --query
Logging to /var/log/mysql-zrm/mysql-zrm-scheduler.log
0 1 * * 1-5 /usr/bin/zrm-pre-scheduler --action backup --backup-set DailyBackupDB1 --backup-level 1 --interval weekly
0 1 * * 6 /usr/bin/zrm-pre-scheduler --action backup --backup-set DailyBackupDB1 --backup-level 0 --interval weekly
0 1 * * 1-5 /usr/bin/zrm-pre-scheduler --action backup --backup-set DailyBackupDB2 --backup-level 0 --interval weekly
```

10.   Perform a test recovery of the databases on both database servers

We want to simulate a situation when both databases “moviesinnodb” and “moviesmyisam” residing on “db1.zmanda.com” have been dropped and need to be restored.

a. Login to MySQL-ZRM server “quartz.zmanda.com”

b. Run this command to view restore information information for backup set DailyBackupDB1

```
root@quartz mysql-zrm]# mysql-zrm-reporter  --show restore-info --where backup-set=DailyBackupDB1
```

```
backup_set       backup_date                                  backup_level    backup_directory
--------------------------------------------------------------------------------------------------------------------------------------------------------------
DailyBackupDB1  Tue 28 Nov 2006 02:46:11 PM PST    0                  /var/lib/mysql-zrm/DailyBackup/DB1/20061128144611
```

c.

```
[root@quartz mysql-zrm]#  mysql-zrm --action restore --backup-set DailyBackupDB1 --source-directory /var/lib/mysql-zrm/DailyBackupDB1/20061128144611
INFO: ZRM for MySQL Enteprise Edition - version 1.1
INFO: Restored database(s) from logical backup:  moviesinnodb
INFO: Restore done in 608 seconds.
MySQL server has been shutdown. Please restart after verification.
```

d. Login to “db1.zmanda.com and start the MySQL service

```
db1:/var/lib/mysql # service mysql start
Starting service MySQL                                    done
```

The scenario here is that one database “moviesmyisam” residing on “db2.zmanda.com” has been dropped and needs to be restored.

a. Login to MySQL-ZRM server “quartz.zmanda.com”

b. Run this command to view restore information for backup set DailyBackupDB2

```
root@quartz mysql-zrm]# mysql-zrm-reporter  --show restore-info --where backup-set=DailyBackupDB2
```

```
backup_set        backup_date                                   backup_level     backup_directory
------------------------------------------------------------------------------------------------------------------------------------------------------
DailyBackupDB2  Wed 29 Nov 2006 01:00:06 AM PST      0                  /var/lib/mysql-zrm/DailyBackupDB2/20061128145543
```

c. Run this command to initiate the restore process

```
[root@quartz DailyBackupDB2]#  mysql-zrm --action restore --backup-set DailyBackupDB2 --source-directory /var/lib/mysql-zrm/DailyBackupDB2/20061128145543
```

```
INFO: ZRM for MySQL Enteprise Edition - version 1.1
WARNING: Binary logging is off. Incremental backups cannot be done.
INFO: Restore done in 169 seconds.
MySQL server has been shutdown. Please restart after verification.
```

d Start the MySQL service

```
b2:/ # service mysql start
Starting service MySQL                                   done
```

Our recovery was successful!

Conclusion

We hope that this document helped you to understand what considerations should be taken into account when you design your backup and recovery operations for remote MySQL server. Our goal was not to give you ‘the right answer” for your specific backup requirements, but to give you enough information so you can make more informative decision about trade-offs associated with each operational option provided by ZRM for MySQL.