

These slides are available at <http://mysqlbackup.zmanda.com/>



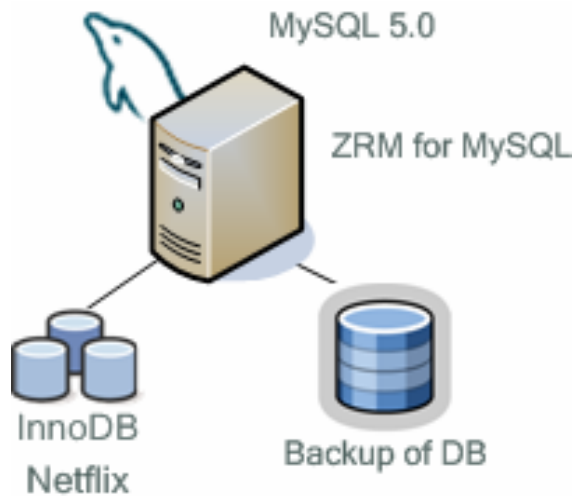
Zmanda Recovery Manager for MySQL

Open source MySQL backup and recovery software

Paddy Sreenivasan paddy@zmanda.com
VP, Engineering

- Backup requirements and backup options for MySQL
- Zmanda Recovery Manager for MySQL features
- Example use case
- Roadmap
- Community edition
- Zmanda network

ZRM for MySQL Demo with Netflix movie database



- LVM snapshot backup
 - Incremental backup
 - Recovery from snapshot
-
- Point in time recovery

- Loss of MySQL data can be catastrophic, resulting in lost revenues, lost customers and lost reputations
- Common reasons for loss of data
 - Hardware and software failures
 - User error, e.g. accidental dropping of a table
 - Damaged tables because of improper shutdown or power failure
- Common ways to protect against data loss
 - Transactional logging supported by databases such as InnoDB, Falcon
 - Replication (including snapshots)
 - Backup

Requirements for Backup by MySQL DBAs

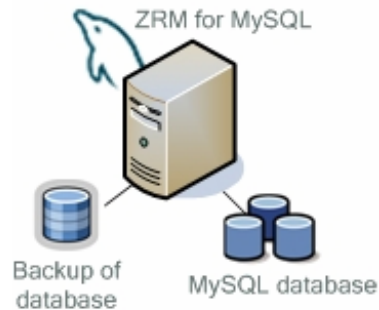


- Backup live database with minimal impact on application and users
- Versatile
 - support for a variety of MySQL implementations
 - scale up and scale out
 - backup of local or remote MySQL servers
- Granular Recovery Point Objective
 - DBA should be able to recover to a particular point back in time
 - Often time a DBA wants to go back to a particular database event
- Enterprise management functionality
 - Scheduler
 - Backup policy
 - Monitoring and reporting
- Flexible, easy to use and secure
- No vendor lock-in, open and defined data formats

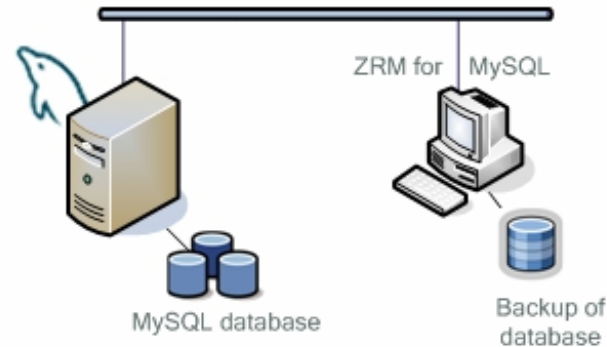
Introducing Zmanda Recovery Manager for MySQL



ZRM local to MySQL



ZRM remote to MySQL



© Zmanda, Inc.

- Schedule full and incremental backups of MySQL database
- Logical backups for portability and raw backups for faster restores
- Perform backup that is the best match for used storage engine and MySQL configuration
- Monitor and report about backups, e-mail or RSS feed reports
- Backup compression and encryption
- Enforcement of site or application specific backup policy
- Plug-in architecture

ZRM is the most comprehensive backup for MySQL



	ZRM for MySQL	MySQL Online Backup API	<i>mysqldump</i>	<i>mysqlhotcopy</i>	InnoDB hotbackup	Open source MySQL projects
Storage engines supported	All	All	All	MyISAM and Archive only	InnoDB only	Limited
Scheduler	Yes	No	No	No	No	Yes
Support for remote MySQL servers	Yes	Yes	Yes	No	No	Yes
Incremental backup	Yes	No	No	No	No	No
Backup policy, e.g. removing backups after 30 days	Yes	No	No	No	No	Yes
Granular recovery	Yes	No	No	No	Yes	No
Open source?	Yes	Yes	Yes	Yes	No	Yes
Reporting	Yes	No	No	No	No	Limited
Selective recovery	Yes	No	Yes	No	No	Limited
MySQL configuration aware	Yes	No	No	No	No	No

Supported platforms



- Supporting MySQL 4.x and 5.x (including MySQL enterprise server)

- ZRM for MySQL can run on:



- Qualified with RHEL, SLES, Debian, Ubuntu distributions
 - Open source and can run on any UNIX platform that has perl support
- For remote backup ZRM has to be on supported Linux, but MySQL can be running on any platform that supports MySQL client running on Linux
- ZRM images of logical backups are transportable and can be used for MySQL migration:
 - MySQL on Solaris and MySQL on Linux
 - From one storage engine to another
 - From one version of MySQL to newer versions
 - From 32-bit to 64-bit



Useful for bare metal recovery of MySQL

Logical backup contains statements that can re-construct the database schema and contents

Advantages

- Works for all storage engines except NDB (clustering)
- Allows remote backups
- Can be restored to any platform architecture and even to another database
- Warm backups

Disadvantages

- Slow, especially for restores
- Backup size can be larger than the database

Raw backup provides a binary copy of the database

Advantages

- Faster than logical backups
- Size of the copy is the same as the database size
- Better database size scalability compared to logical backups
- Warm backups

Disadvantages

- Can be restored **ONLY** to the same version of MySQL server on the same platform as the original data

- DB files must reside on LVM2 logical volume
- Works best for transactional engine such as InnoDB

Advantages

- Provides hot backup for InnoDB and warm backup for other storage engines
- Scalable
- Almost instantaneous implementation of raw backup

Disadvantages

- Works best for filesystems that support freeze operations such as XFS and VxFS
- Disk space for logical volume snapshots

MySQL replication provides hot backup



- ZRM for MySQL does not set up MySQL replication, it rather takes advantage when replication is used
- ZRM backs up the replication related files as well

Advantages

- Can be used for disaster recovery also
- Replication slave can be used for load balancing and HA
- Hot backup

Disadvantages

- Not scalable as replication slave is always behind the master
- Expensive (additional server, more network bandwidth)

- Each MySQL environment is unique and users have different requirements for backup and recovery
- ZRM provides a unified interface to different methods of backup for MySQL
- ZRM makes intelligent decisions what is the best match for a particular setup
- Master one tool that is flexible enough to work in all situations

Incremental backups reduce load on resources



- Full backups
 - Logical and raw backup support
 - Local and remote backups
 - Multiple databases or multiple tables in database
- Incremental backups - incremental to prior incremental or to full backup
 - Incremental backups require binary logs to be enabled on the MySQL server
 - Incremental backups can not be DB or table specific
 - Incremental raw and logical backups of remote MySQL servers

Backup policies = Success of recovery



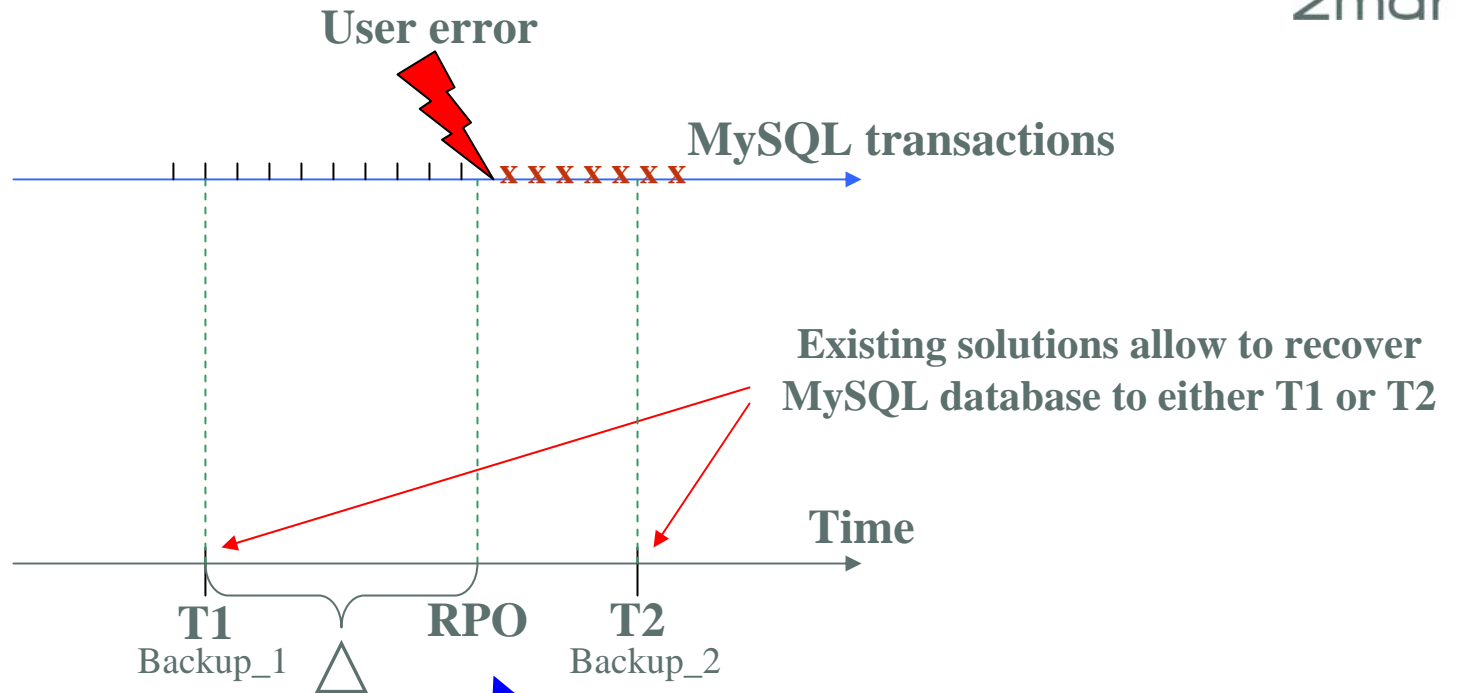
- Backup retention policy
 - How long backup images to be retained
 - Backups can be retained for weeks, months or years
- Backup scheduling policy
 - Backups can be done immediately or scheduled
 - Scheduling parameters
 - Daily backups at specific time
 - Weekly backups at specific time
 - Monthly backups at specific time
- Backup compression and encryption using platform tools

“R” in ZRM is for Recovery



- Data can be recovered from full and incremental backups
- Selective recovery of a table/database from a backup image
- Browse database events in the incremental backup images
- Database recovery to any point in time or to any database event
 - Recovering from operator errors
 - Recovering from malicious activities

Granular Recovery Point Objective



Only ZRM for MySQL allows recovery to a granular Recovery Point Objective , e.g. just before user error

- Backup notifications - Email, RSS feed
 - Pre-defined backup reports
 - Custom reports of 25 different backup parameters
 - User configurable backup report formats
 - Backup reports - Text or HTML
- Backup status
 - Backup methods used
 - Backup retention report
 - Backup performance report
 - Backup application performance report
 - Backup replication information
 - Backup restoration information
 - Selective recovery report

Plug-in architecture



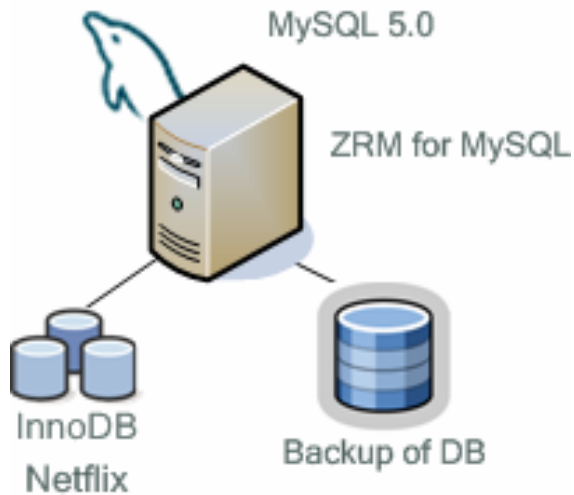
- Pre- and post- backup
- Pre-scheduler
 - Flexible scheduling based on server load
- Secure copy
 - Securely transfer data between MySQL server and host running ZRM for MySQL
- Binary log parser
 - Support handling millions of database events
- Backup compression and encryption

Flexible configuration - Backup sets

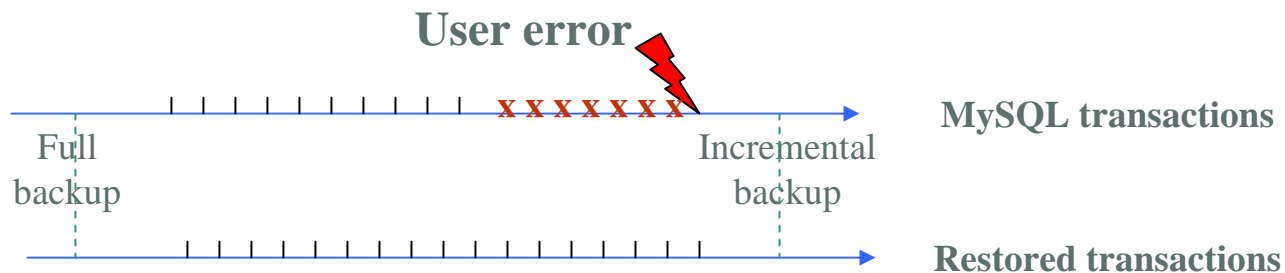


- What to backup? (MySQL database, tables)
- Which method to use? (raw, logical, replication, LVM snapshot)
- When and how to backup? (retention, scheduling, compression, encryption)
- MySQL server parameters and custom plug-ins
- Configuration is stored in a file
 - All parameters are optional and *my.cnf* MySQL configuration file is used for server parameters

ZRM for MySQL Demo with Netflix movie database



- LVM snapshot backup
 - Incremental backup
 - Recovery from snapshot
-
- Point in time recovery



Example use case - Backup Set - Netflix



- */etc/mysql-zrm/mysql-zrm.conf*
mail to="mysql dba@company. com"
- */etc/mysql-zrm/netflix/mysql-zrm.conf*
comment=MySQL Webinar Jan 17th, 2007
backup-level=0
backup-mode=raw
lvm-snapshot=30M
Compress=1
retention-policy=10D
databases=netflix
html-reports=backup-status-info, backup-performance-info
html-report-directory=/var/www/mysql-zrm/reports/

Example use case - Full backup



```
# mysql -zrm-scheduler --now --backup-set netflix
```

```
INFO: ZRM for MySQL Enterprise Edition - version 1.1
INFO: backup-set=netflix
INFO: backup-size=1.24 GB
INFO: Compressing backup
INFO: backup-size-compressed=455.40 MB
INFO: Removing all of the uncompressed/unencrypted data
INFO: Executing post-backup-plugin
INFO: read-locks-time=00:00:00
INFO: flush-logs-time=00:00:00
INFO: compress-encrypt-time=00:04:41
INFO: backup-time=00:05:27
INFO: backup-status=Backup succeeded
INFO: Backup succeeded
```

Example use case - Incremental backup



```
# mysql -zrm-scheduler --now --backup-set netflix \  
--backup-level 1
```

```
INFO: ZRM for MySQL Enterprise Edition - version 1.1  
INFO: backup-set=netflix  
INFO: backup-size-compressed=0.00 MB  
INFO: Removing all of the uncompressed/unencrypted data  
INFO: Executing post-backup-plugin  
INFO: read-locks-time=00:00:00  
INFO: flush-logs-time=00:00:00  
INFO: compress-encrypt-time=00:00:00  
INFO: backup-time=00:00:00  
INFO: backup-status=Backup succeeded  
INFO: Backup succeeded
```


Example use case - Backup report



ZRM Backup Performance Info

backup_set	backup_date	backup_level	backup_size	backup_size_compressed	backup_time	compress_encrypt_time
netflix	Thu 11 Jan 2007 05:18:30 PM PST	0	1.24 GB	455.40	00:05:27	00:04:41

ZRM Backup Status Info

backup_set	backup_date	backup_level	backup_status	comment
netflix	Thu 11 Jan 2007 05:18:30 PM PST	0	Backup succeeded	MySQL Webinar Jan 17 2007

Example use case - Browsing incremental backups



```
# mysql -zrm --action parse-binlogs --backup-set netflix\  
--source-directory /var/lib/mysql -\  
zrm/netflix/20070112175628/
```

```
-----  
Log filename | Log Position | Timestamp | Event Type | Event  
-----
```

```
/var/lib/mysql-zrm/netflix/20070111175628/mysql-bin.000005 | 98 | 07-01-12 17:55:56 | |  
/var/lib/mysql-zrm/netflix/20070111175628/mysql-bin.000005 | 507 | 07-01-12 17:55:56 | Execute_load_query | use  
netflix; LOAD DATA LOCAL INFILE '/tmp/SQL_LOAD_MB-12-1' INTO TABLE `MovieID` FIELDS TERMINATED  
BY ',' ENCLOSED BY '"' ESCAPED BY '\\' LINES TERMINATED BY '\n';  
/var/lib/mysql-zrm/netflix/20070111175628/mysql-bin.000005 | 728 | 07-01-12 17:55:56 | Xid = 691 | COMMIT;  
/var/lib/mysql-zrm/netflix/20070111175628/mysql-bin.000005 | 755 | 07-01-12 18:56:16 | Query | DELETE FROM  
`MovieID` WHERE `MovieID`.`MovieID` = 17771;  
/var/lib/mysql-zrm/netflix/20070111175628/mysql-bin.000005 | 876 | 07-01-12 18:56:16 | Xid = 695 | COMMIT;  
/var/lib/mysql-zrm/netflix/20070111175628/mysql-bin.000005 | 903 | 07-01-12 18:56:16 | Query | DELETE FROM  
`MovieID` WHERE `MovieID`.`MovieID` = 17772;  
/var/lib/mysql-zrm/netflix/20070111175628/mysql-bin.000005 | 1024 | 07-01-12 18:56:16 | Xid = 696 | COMMIT;  
/var/lib/mysql-zrm/netflix/20070111175628/mysql-bin.000005 | 1051 | 07-01-12 18:56:16 | Query | DELETE FROM  
`MovieID` WHERE `MovieID`.`MovieID` = 17773;  
/var/lib/mysql-zrm/netflix/20070111175628/mysql-bin.000005 | 1791 | 07-01-12 18:56:16 | Query | drop table MovieID;  
/var/lib/mysql-zrm/netflix/20070111175628/mysql-bin.000005 | 1875 | 07-01-12 18:56:28 | Rotate to mysql-bin.000006  
pos: 4 |  
-----
```

Example use case - Operator Error

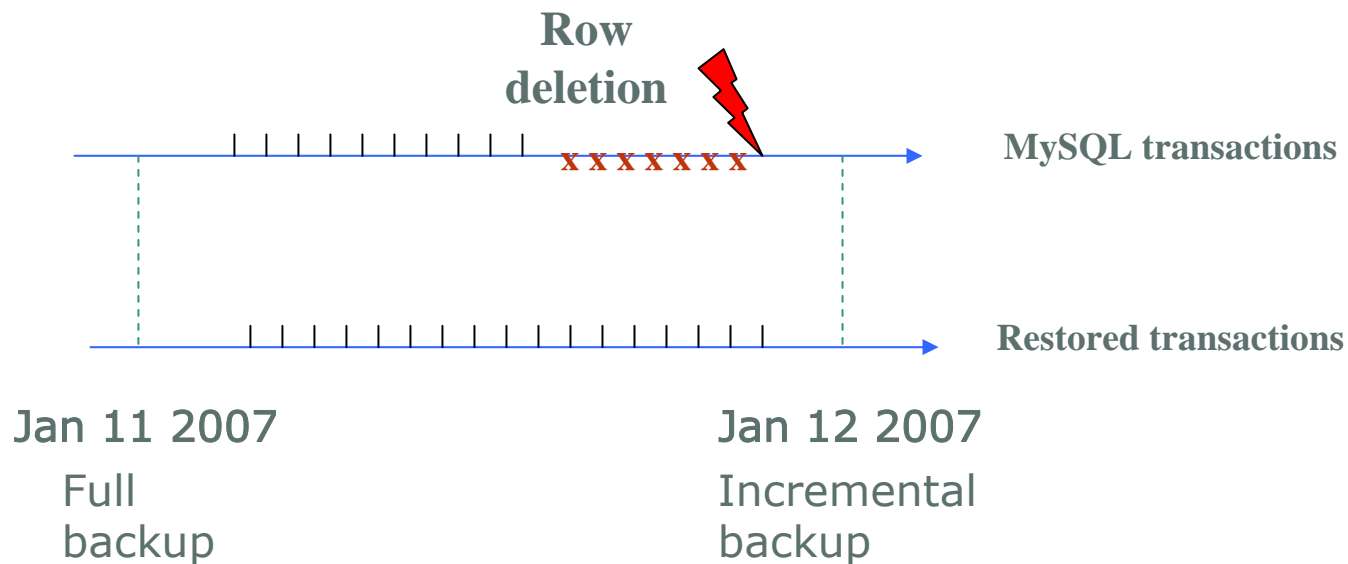


```
# mysql -zrm --action parse-binlogs --backup-set netflix\  
--source-directory /var/lib/mysql -\  
zrm/netflix/20070112175628/
```

Log filename | Log Position | Timestamp | Event Type | Event

```
/var/lib/mysql-zrm/netflix/20070111175628/mysql-bin.000005 | 98 | 07-01-12 17:55:56 | |  
/var/lib/mysql-zrm/netflix/20070111175628/mysql-bin.000005 | 507 | 07-01-12 17:55:56 | Execute_load_query | use  
netflix; LOAD DATA LOCAL INFILE '/tmp/SQL_LOAD_MB-12-1' INTO TABLE `MovieID` FIELDS TERMINATED  
BY ',' ENCLOSED BY '"' ESCAPED BY '\\' LINES TERMINATED BY '\n';  
/var/lib/mysql-zrm/netflix/20070111175628/mysql-bin.000005 | 728 | 07-01-12 17:55:56 | Xid = 691 | COMMIT;  
/var/lib/mysql-zrm/netflix/20070111175628/mysql-bin.000005 | 755 | 07-01-12 18:56:16 | Query | DELETE FROM  
`MovieID` WHERE `MovieID`.`MovieID`= 17771;  
/var/lib/mysql-zrm/netflix/20070111175628/mysql-bin.000005 | 876 | 07-01-12 18:56:16 | Xid = 695 | COMMIT;  
/var/lib/mysql-zrm/netflix/20070111175628/mysql-bin.000005 | 903 | 07-01-12 18:56:16 | Query | DELETE FROM  
`MovieID` WHERE `MovieID`.`MovieID` = 17772;  
/var/lib/mysql-zrm/netflix/20070111175628/mysql-bin.000005 | 1024 | 07-01-12 18:56:16 | Xid = 696 | COMMIT;  
/var/lib/mysql-zrm/netflix/20070111175628/mysql-bin.000005 | 1051 | 07-01-12 18:56:16 | Query | DELETE FROM  
`MovieID` WHERE `MovieID`.`MovieID` = 17773;  
/var/lib/mysql-zrm/netflix/20070111175628/mysql-bin.000005 | 1791 | 07-01-12 18:56:16 | Query | drop table MovieID;  
/var/lib/mysql-zrm/netflix/20070111175628/mysql-bin.000005 | 1875 | 07-01-12 18:56:28 | Rotate to mysql-bin.000006  
pos: 4 |  
-----
```

ZRM for MySQL Demo with Netflix movie database



Example use case - Recovery from full backup



```
# mysql -zrm --action restore --backup-set netflix --\
  source-registry /var/lib/mysql -\
  zrm/netflix/20070111171830/
```

INFO: ZRM for MySQL Enterprise Edition - version 1.1

INFO: source-registry=/var/lib/mysql -zrm/netflix/20070111171830

INFO: InnoDB data file are /var/lib/mysql/ibdata1

INFO: InnoDB log dir is /var/lib/mysql/.

INFO: Shutting down MySQL

INFO: Restored database from raw backup: netflix

INFO: Restore done in 61 seconds.

MySQL server has been shutdown. Please restart after
verification.

Example use case - Recovery to a point in time



```
# mysql -zrm --action restore --backup-set netflix --stop-\  
  datetime "20070112180000" --source-directory  
  /var/lib/mysql -\ zrm/netflix/20070112175628/
```

INFO: Incremental restore done for database netflix

INFO: Shutting down MySQL

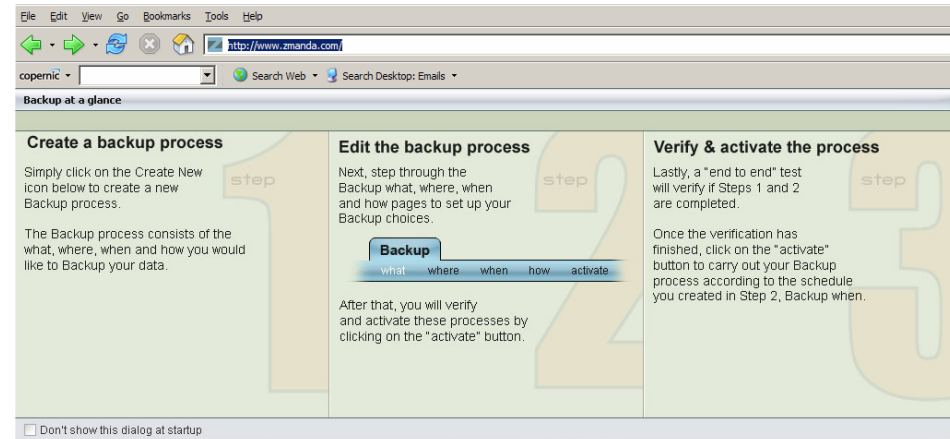
INFO: Restore done in 3 seconds.

MySQL server has been shutdown. Please restart after
verification.

ZRM for MySQL Roadmap



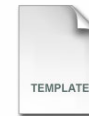
- Support for ZRM itself running on Windows, Solaris
- Improvements to ease of use
 - Browser based Management Console
- Support for MySQL streaming online backup API
- Support for more storage engines (MySQL cluster, Falcon)
- Deeper integration with applications that use MySQL



Create NEW backup process



Create from TEMPLATE




Open EXISTING



- Open source backup and recovery software leader
- Mission: Bring open source benefits (lower cost and user control) to the backup and recovery software market
- Products/ Services:
 - Suite of software and services based on Amanda - The world's leading open source backup and recovery software
 - Application/ function specific backup and recovery solutions such as ZRM for MySQL
- Active contributor and supporter of open source projects

- Community project launched in August 2006
- New release every month
- More than 10,000 downloads so far
- Community code as well as documentation contributions
- Resources
 - Downloads <http://www.zmanda.com/downloads.html>
 - Wiki documentation <http://mysqlbackup.zmanda.com/>
 - Forums <http://forums.zmanda.com/>
 - Bug/Feature requests
<http://forums.zmanda.com/bugzilla/>

- Free registration
- ZRM for MySQL enterprise edition available on annual subscription
- Zmanda support and services



The screenshot shows the Zmanda Network login interface. On the left, there is a login form with fields for 'Username:' and 'Password:', a 'Login' button, and links for 'Lost your password?' and 'Click here to register'. On the right, there is a header with the Zmanda logo and 'zmanda Open Source Backup'. Below the header, it states 'Zmanda Network provides access to:' followed by a list of services: 'white papers', 'tutorials and training', 'community forums', and 'product alerts relevant to your environment'. At the bottom, there is a footer with 'Zmanda Inc. © 2006 All Rights Reserved' and '1 888 496 2632 or support@zmanda.com'.

What to consider when you back up a remote MySQL server with ZRM over the Internet

By Dmitri Joukovski and Pavel Pragin

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.

- MySQL backup white papers

Next steps



- Download and try community edition -
<http://www.zmanda.com/downloads.html>
- Join us for our next webinar on Jan 25, 2007 10am PT
ZRM for MySQL backups over internet
<http://www.zmanda.com/webinar.html>
- Review documentation, ask questions in Zmanda forums
<http://mysqlbackup.zmanda.com>
<http://forums.zmanda.com>
- Buy ZRM for MySQL subscriptions for enterprise edition
<http://network.zmanda.com/>
- Meet MySQL backup and recovery experts from Zmanda at MySQL users conference, April 2007