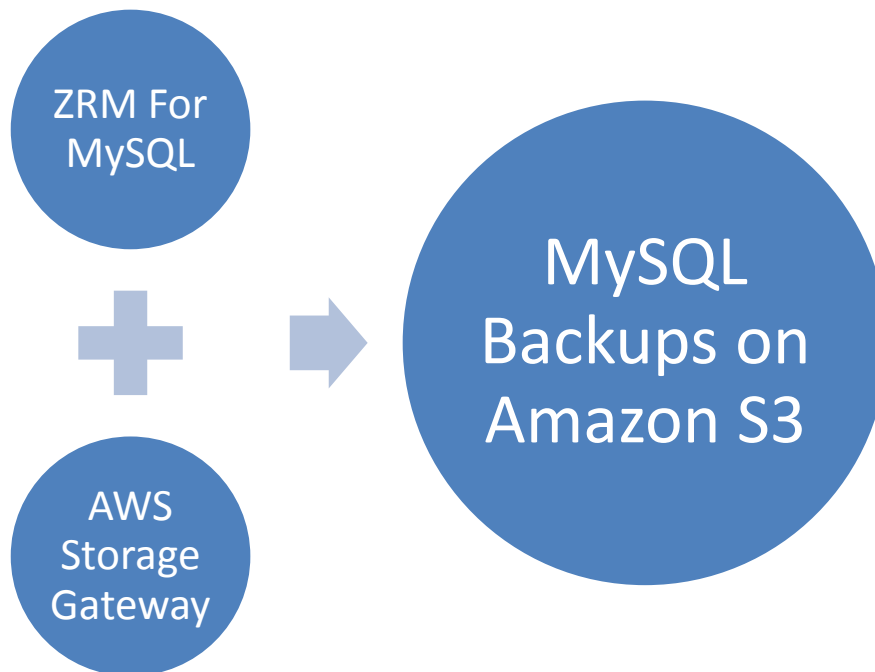


MySQL backups to Amazon S3 using Zmanda Recovery Manager & Amazon Storage Gateway

A MySQL Disaster Recovery Solution



April 2012

1	Zmanda Recovery Manager for MySQL	3
2	Amazon Storage Gateway and ZRM	3
3	Configuration	4
3.1	Requirements.....	4
3.2	Configure Amazon Storage Gateway VM.....	4
3.3	ZRM Server Configuration.....	4
3.4	Disaster Recovery of ZRM server and MySQL servers	6
4	References	7

1 Zmanda Recovery Manager for MySQL

Zmanda Recovery Manager (ZRM) for MySQL offers an easy-to-use yet flexible and robust backup and recovery solution for MySQL servers. ZRM for MySQL allows you to:

- Schedule full and incremental backups of your MySQL database.
- Start immediate backup or postpone scheduled backups based on thresholds defined.
- Choose to do more flexible logical or faster raw backups of your database.
- Perform backup that is the best match for your storage engine and your MySQL configuration.
- Backup 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 e-mail notification of the status of your backups and receive MySQL backup reports via RSS feed.
- Monitor and browse your backups.
- Define retention policies and automatically delete backups that have expired.
- Recover a database easily to any point in time or to any particular transaction, e.g. just before a user made an error.
- Parse binary logs to search and filter MySQL logs for operational and security reasons
- Snapshot live MySQL with Linux LVM, Windows VSS, Solaris ZFS, NetApp SnapManager and Veritas VxFS to minimize locking on the database. This allows you to backup mission-critical MySQL production environments without interrupting online data access.
- Global management of backup and recovery of hundreds of MySQL databases from a single web-based Zmanda Management Console.

2 Amazon Storage Gateway and ZRM

Amazon Storage Gateway provides a storage gateway to the Amazon S3 Cloud by asynchronously replicating the volume in the data center to an Amazon Elastic Block Storage (EBS) volume in the cloud.

Amazon Storage Gateway can be used to replicate the ZRM configuration and ZRM for MySQL backup images from the storage volume in the data center to the Amazon Cloud. The MySQL database restores can be performed from the backup data stored in the local data center for quick recovery of databases.

In case of catastrophic data center failures, the ZRM server configuration and backup data can be recovered to a physical or virtual machine in the local data center or to Amazon EC2 virtual machine in the cloud. After ZRM server recovery, all MySQL databases can be restored and brought online.

This white paper provides information on how to configure ZRM server with Amazon Storage Gateway volume and provides steps to perform disaster recovery.

3 Configuration

3.1 Requirements

- ZRM for MySQL server – It can be a physical or virtual machine. *iscsi* client software packages should be installed.
- Amazon Storage Gateway - A VMware VM running on ESX to install and configure the storage gateway software with sufficient storage to store all MySQL backups.
- MySQL server(s) that are backed up using the ZRM server
- Amazon Web Services account and Amazon EBS volume to mirror storage gateway volume.

3.2 Configure Amazon Storage Gateway VM

Please follow steps in the Amazon manuals to install and configure the storage gateway virtual machine.

1. <http://docs.amazonwebservices.com/storagegateway/latest/userguide/SettingUpAWSStorageGateway.html>
2. <http://docs.amazonwebservices.com/storagegateway/latest/userguide/ManagingActivatedGateway.html>

3.3 ZRM Server Configuration

An existing ZRM server configuration can be used for this setup. If you are planning to install and configure ZRM server, please follow instructions from the ZRM manual available at <http://docs.zmanda.com>.

Following procedure configures the Amazon Storage gateway volumes as the destination for MySQL backups. These steps have to be performed as a superuser.

3.3.1 iSCSI Installation

Installation iSCSI client initiator on the ZRM server so that it can access storage volumes from the Amazon Storage Gateway

```
For RHEL/CentOS : yum install iscsi-initiator-utils
```

For Ubuntu: `sudo apt-get install open-iscsi`

3.3.2 Discover and Add Target

1. Run and preserve output of “fdisk -l”
2. Start the iscsi service

```
/etc/init.d/iscsi start
```

3. Run following to discover iscsi target

```
iscsiadm -m discovery -t sendtargets -p <aws storage gateway IP addr.>
```

4. To add, restart iscsi service

```
/etc/init.d/iscsi restart
```

3.3.3 Configure iSCSI Volume

After discover and add target, iscsi volume would be added to your ZRM server machine.

To identify run

```
fdisk -l
```

or

```
tail -f /var/log/messages
```

A new device like “/dev/sd..” or “/dev/xvd..” would be seen, to easily identify compare it with “fdisk -l” output preserved before the iscsi service restart.

Format the volume (please double check that this volume does not contain a needed filesystem):

```
mkfs.ext3 /dev/sdd1
```

3.3.4 Mount the Volume

Mount this volume on the ZRM server in “/var/lib/mysql-zrm” mount point - the default location.

1. If you have existing ZRM server, copy the data of /var/lib/mysql-zrm/ to a safe location,

```
cp -r /var/lib/mysql-zrm/* /tmp/mydata/
```

2. Remove the content of /var/lib/mysql-zrm/

```
rm -rf /var/lib/mysql-zrm/*
```

```
mount the volume on /var/lib/mysql-zrm/
```

```
mount /dev/<volume-name> /var/lib/mysql-zrm
```

3.3.5 Migrate ZRM configurations directory to AWS Storage Volume

1. Create Configuration Directory in AWS storage volume

```
mkdir /var/lib/mysql-zrm/mysql-zrm
```

2. Copy the configurations

```
cp -r /etc/mysql-zrm/* /var/lib/mysql-zrm/mysql-zrm/
```

3. Remove the configuration directory content.

```
rm -rf /etc/mysql-zrm
```

4. Create a soft link to the aws volume configuration directory

```
ln -s /var/lib/mysql-zrm/mysql-zrm /etc/mysql-zrm
```

3.3.6 Apply permissions for mysql user.

```
chown mysql:mysql -R /var/lib/mysql-zrm
```

```
chown mysql:mysql /etc/mysql-zrm
```

3.3.7 Restart ZRM to apply changes

```
/etc/init.d/zmc_zrm restart
```

At this point your ZRM configuration and your MySQL backups will be asynchronously sent to Amazon's cloud storage by the AWS Gateway.

3.4 Disaster Recovery of ZRM server and MySQL servers

Disaster recovery can be performed in two ways

1. Recover ZRM and MySQL backup data in EC2 AMI.
2. Recover backup and configuration data on Physical machine or virtual machine in your local data center.

After recovering the ZRM server, you can recover the MySQL server data using the Zmanda Recovery Manager. Please follow the procedures documented in the ZRM manual.

3.4.1 Recover ZRM and MySQL backup data in EC2 instance

1. Identify the volume name in AWS Storage Gateway Console
2. Go to "Gateways" > "Volumes"

3. Go to AWS EC2 Console > Snapshots
4. Identify Snapshots of your Gateway Volume.
5. Create a Volume out of snapshot as per your required timestamp or the latest snapshot.
6. Launch a Linux EC2 AMI (preferably the same distribution and version as the original ZRM server) and attach the volume created by snapshot to the AMI.
7. Identify the device created by attaching the snapshot, mount the device to `/var/lib/mysql-zrm` mount point.
8. Configure ZRM as described above from 3.3.4 through 3.3.7, your configuration would be recovered, enabling you to perform MySQL recovery from a desired backup image.

3.4.2 Recover ZRM and MySQL backup data on a machine in your data center

1. Identify you volume name in AWS Storage Gateway Console
2. Go to “Gateways” > “Volumes”
3. Go to AWS EC2 Console > Snapshots
4. Identify Snapshots of your Gateway Volume.
5. Create a Volume out of snapshot as per your required timestamp or the latest snapshot.
6. Launch a Linux AMI and attach the volume created by snapshot to the AMI.
7. Identify the device created by attaching the snapshot, mount the device mount point.
8. Copy the backup data `/var/lib/mysql-zrm` using `scp` to your local ZRM machine.
9. Copy the configurations `/etc/zmanda/` using to your local ZRM machine.
10. Ensure permissions of directories are “mysql:mysql”.
11. Restart ZRM services, your configuration would be recovered, enabling you to perform MySQL recovery from a desired backup image.

4 References

1. ZRM for MySQL product - <http://www.zmanda.com/zrm-mysql-enterprise.html>
2. ZRM for MySQL manuals - <http://docs.zmanda.com/>
3. Amazon Storage Gateway - <http://docs.amazonwebservices.com/storagegateway/latest/userguide/SettingUpAWSStorageGateway.html>