



Top 5 considerations while setting up your MySQL backup

A best practice whitepaper
by Zmanda





MySQL databases increasingly power production applications - whether these be web based collaboration tools or CRM applications. A well thought through backup plan and configuration will go a long way to ensure that you can recover your database - when a system or user error deletes important data stored in MySQL - without impacting your business.

In this article we provide a list of the top 5 considerations while planning a solution for backing up MySQL databases in your environment and implementing that plan.

1) How fast and how easy do you want the MySQL Recovery process to be?

Just like planning any other backup implementation, the most important consideration is the recovery process. Here are two key recovery objectives:

1. To what point in time will you need to recover your database?
The Recovery Point Objective (RPO) defines to what point in time you have to recover your database. For example, if a user dropped a table by accident at 12:55 pm, you might want to recover to a point in time just before the delete command. In some circumstances you may want to selectively exclude some rogue transactions during the recovery process.
2. How much time will you have to recover your database?
The Recovery Time Objective (RTO) defines how much time it will take to recover the database. You should carefully consider how long your MySQL application can be offline while you do a recovery.

Therefore, you will need to determine RPO and RTO for MySQL database in your environment and implement your backup solution accordingly.

When planning the recovery process, remember that different circumstances will require different levels of recovery. Depending on the situation, you may need to recover:

- The whole server - all databases (in case of disaster recovery)
- Specific database
- Specific table
- Specific transactions



Make sure to periodically test your recovery procedures by actually conducting recoveries. Actual recoveries happen under stress and this is not the time for practice. Once again it is all about recovery when planning your MySQL Backup solution!

2) What will be the impact of MySQL Backup process on your Application?

Backup window refers to the time it takes for backups to complete. During the backup window your MySQL applications will be either offline or degraded (in performance or functionality or both).

- Some backup methods, e.g. offline backup make MySQL databases totally unavailable.
- Other, e.g. *mysqldump* make the database unavailable for update.
- Still others (replication, snapshots) have very minimal impact on the MySQL databases and applications

The size of your database and rate of transaction activity will determine how long your backup window will be. Depending on these factors and the business requirements of your MySQL based applications you will need to decide on the scheduling of your backup window. You will also need to plan on potentially wide gyrations of the backup window duration - e.g. if some business activity results in a huge number of updates to your database. Ideally your implementation should also be aware of the current activity on your database - e.g. if your database is presently loaded with transactions, your implementation can intelligently postpone a scheduled backup run for a few hours.

Your applications may have data stored in files outside of MySQL database; for example, the configuration files for your application. While implementing a MySQL backup solution make sure to create a process of backing up and recovering these files, which will be needed to completely recover your MySQL based applications.



3) What will your backup configuration look like?

i.e. - the What, Where, When, and How of MySQL Backup.

Your MySQL backup implementation will depend on the specific server and database configuration in your environment. You will need to take an inventory of the following:

- Number of MySQL servers
- Size of MySQL databases
- Storage engines in use (InnoDB, MyISAM, NDB-MySQL Cluster). Note that you may introduce a new MySQL storage engine in the future. Your backup implementation should be flexible to incorporate a new storage engine.
- Database activity
 - Is there a particular time that your database is less active
 - Nature of activity (selects, inserts/updates)
- Use of MySQL replication
- MySQL versions being used
- Use of MySQL stored procedures

While deciding on what databases and tables to backup, make sure to include all databases and tables needed to be backed up together to ensure data consistency at your application level.

Choosing the right location for backup images is also a critical decision. Store backups as far away from the original data as possible! You will be surprised how many DBAs keep the backups on the same RAID as the original data.

A related decision is where you want to recover your database during the recovery process. If recovery is initiated after a simple user error, you may choose to recover the database at its original location. Keep in mind that hardware failure and disaster might require recovering to a different host that could have a different location, hardware, operating system, or version of MySQL. In ideal case, you should have a standby server where you periodically audit your recovery process.

Depending on the size of your databases and your recovery location criteria (i.e. whether you will recover at the same location as your current database or a different location) you will need to plan for allocation of space for backup and recovery purposes. If space is at a premium you can also choose to compress your backup images - although this uses computation resources both at backup and recovery stages.



If you are using MySQL replication for high availability or load balancing, it can also be leveraged for backup purposes - backups can be performed on a slave server without effecting the master server.

4) How will you manage your backup process and backed up data?

Backups should almost always be automated. Backups must be taken consistently and regularly and relying on human intervention should be avoided. Backup catalog should be automatically kept up-to-date keeping track of all copies of backup images.

In typical backup implementations, important pre-backup and post-backup procedures need to be performed. A pre-backup procedure can check whether needed storage will be available for the upcoming backup run. At the other end, a post-backup procedure can remove no longer needed binary logs after a successful backup has finished. Your backup implementation should seamlessly integrate such pre-backup and post-backup procedures.

Security of your MySQL database is a key consideration while implementing a backup solution. While backing up your MySQL database you need to consider whether encryption of your data is required. Just like compression, keep in mind that encryption has a computation cost both during backup and recovery. Also, if you are backing up your data remotely - say from your service provider site to your local data center or vice versa - you should make sure that the transportation mechanism being used is secure against eavesdroppers.

Proper planning should be done as to which personnel in your organization can perform the recovery of the MySQL database. These personnel need to have appropriate (technical and business) permissions to recover, for example, the Accounting database. Proper training for the personnel is essential before performing a recovery.

Appropriate security and access to backup images also plays into the compliance requirements that your organization may have, which leads us to the next top consideration.



5) What kind of tracking, reporting and compliance requirements does your business have from your MySQL backup implementation?

Your MySQL backup implementation should provide timely notifications for critical events such as backup failures. Mechanisms may include email, SMS, or RSS feed captured on an administrator's dashboard.

It is likely that multiple personnel in your organization would be interested in receiving reports of MySQL backups. This is especially true if your environment has multiple entities using MySQL databases for different purposes. MySQL Backup reports can also be used for other interesting purposes, e.g. identifying durations of high MySQL update activity.

Your MySQL backup implementation should automatically implement your Retention Policy - i.e. how long you want keep to your backed up MySQL data. Your backup procedures should account for the possibility that different types of data may have different retention policies - depending on compliance and business requirements. The expired backups should be automatically purged.

Now that you understand these concepts, we invite you to try Zmanda Recovery Manager for MySQL. Contact zsales@zmanda.com to request a demo or a free 30-day trial.