Type of Backups in MySQL, Offsite Storage, Encryption and Restore Testing

Type of MySQL Backups

- It’s important to configure a new and dedicated replica node for backups purposes, due to the high CPU load to avoid any issue on any other replica node (AKA **backup server**).
- Types of Backups :
- Logical Backup
- Physical (Raw) Backup
- Snapshot Backups
- Binary Log Backups

Logical Backups

- This is a dump from logical database structure (CREATE DATABASE, CREATE TABLE statements) and content (INSERT statements).
- This is recommended to be used against smaller amounts of data.
- The disadvantage of this method is slower (backup and restore) if you compare it with physical backups.
  Using mydumper you can backup and restore a single database or a single table if it’s needed.
- This is useful to copy some data to a different environment to run tests. Also, mydumper can take a consistent (as long as all the tables are InnoDB engine) backup and provides accurate master and slave log positions.
- The output is larger than for physical backup, particularly when saved in text format, but it can be compressed on the fly depending on the software you are using.
- mydumper can compress and mysqldump needs to add a pipe to redirect the output to gzip, for example.
- Logical backups are used to address data corruption or the need to restore a subset of tables.

Physical (RAW) Backup

- In short, this consists of exact copies of database directories and files.
- This can be a copy for all or a part from MySQL datadir directory. This kind of backup is most used to restore or create a new replica node easily and quickly and is used to address host failure.
- It’s recommended to restore using the same MySQL version.
- We can use Percona XtraBackup because it can include any related files such as configuration files like cnf config files.

Snapshot Backups

- Some file system implementations enable “snapshots” to be taken.
- These provide logical copies of the file system at a given point in time, without requiring a physical copy of the entire file system.
- MySQL itself does not provide the capability for taking file system snapshots but it is available using third-party solutions such as LVM or ZFS.
- The disadvantage is that sometimes physical backups do not compress much, because data is usually in a binary format and sometimes the table is already compressed.

Binary Log Backups

- Binlog backups specifically address RPO.
- Binary log files contain records of each SQL query executed that made changes.
- From MySQL 5.6 on, you can use mysqlbinlog to stream binary logs from a remote server.
- You can combine binlog backups with Percona XtraBackup or mydumper backup to allow restoration up to the end of the most-recently-backed-up binary log.

Incremental / Differential Backups

- An incremental backup is a backup of everything that has changed since the last backup (a binary log backup is a special case of an incremental backup).
- This is a very good option if the dataset size is huge, as you can take a full backup at the beginning of the week and run incremental backups per day. Also, the backup size is smaller than the full backup.
- The main risks associated with incremental backups are:
- A single corrupt incremental backup may invalidate all the others
- Incremental backups typically negatively affect the RTO
- For a differential backup, it copies the differences from your last backup.
- The advantage is that a lot of data does not change from one backup to the next, so the result can be significantly smaller backups. This saves disk space.
- Percona XtraBackup supports both incremental and differential backups.

Offsite Storage

- It’s highly recommended to copy all the backup methods to another place, like the cloud or an external file server, so in case of host failure or data center failure, you have another copy.
- Not all the backup files need to be uploaded to the cloud, sometimes the time you need to spend in the download is bigger than the time consumed in the recovery process.
- A good approach is to keep 1-7 days locally on the backup server in case a fast recovery is needed, and this depends on your business regulations.

Encryption

- Backups have sensitive data, so it’s highly recommended to encrypt, especially for offsite storage.
- This adds more time when you need to restore a backup but it keeps your data safe.
- GPG is a good option to encrypt backups, and if you use this option or some other alternative
- Don’t forget to get a copy of the keys/passphrase. If you lose it, your backups will be useless.

Restore Testing

- Depending on your business, it’s highly recommended to test your backups at least once per month.
- This action validates your backups are not corrupted and it provides critical metrics on recovery time.
- This process should be automated to get the full backup, restore it, and finally configure this server as a replica from the current primary or another replica.
- This is good as well to validate that the replication process has no errors.
- Many customers are using this methodology to refresh their QA/STG environment to have fresh data from production backups.
- In addition to the above, it is recommended to create a manual or automated restore documentation process to keep all the steps together, so in case of disaster, you can follow it without wasting time.

Retention Requirements

- Last but not least, it is very important to keep multiple copies of different backup types.
- Best recommendation is:
- One or two physical backups locally on the backup server (as long as space allows it).
- Seven daily and four weekly logical backups locally on the backup server.
- 30 days of binlog backups locally on the backup server.
- For offsite backups (like S3, Google Cloud, etc.), keep monthly backups for one year or more.
- For local backups, keep in mind you will need a minimum of **2.5 times** the current dataset size as free disk space to save/meet these retention policies. Don’t forget to encrypt all the backup types!
- Legal or regulatory requirements may also dictate how long data must be archived.