Pythian

MySQL backup solutions in 2021

Matthias Crauwels
Lead Database Consultant (MySQL)

Percona Live Online - May 13 2021
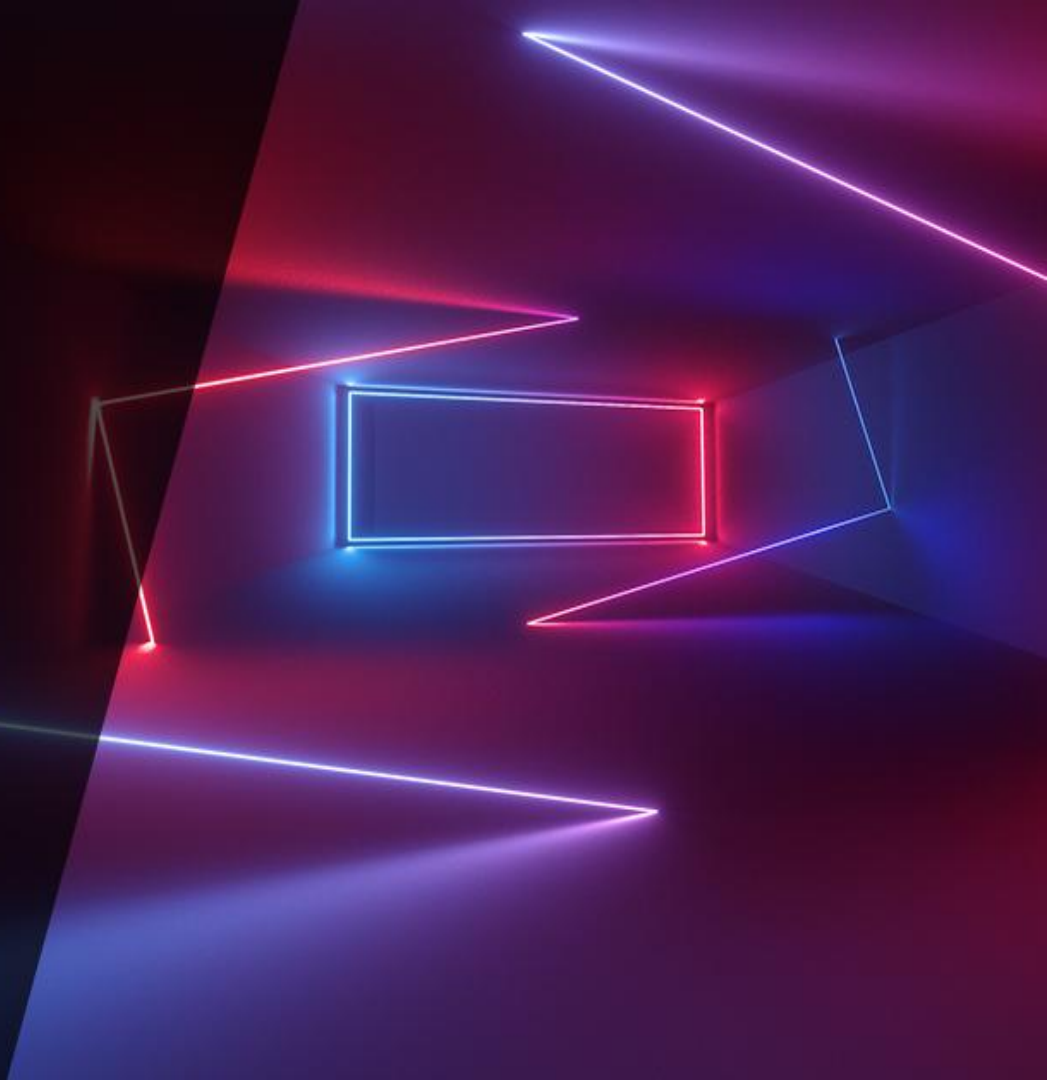
Who am I?

Matthias Crauwels

- Living in Ghent, Belgium
- Bachelor Computer Science
- ~20 years Linux user / admin
- ~10 years PHP developer
- ~8 years MySQL DBA
- 3rd year at Pythian
- Currently Lead Database Consultant
- Father of Leander

Pythian

Helping businesses use data to compete and win

Pythian

L♥VE YOUR DATA

Pythian

Founded in 1997, Pythian is a global IT services company that helps organizations transform how they compete and win by helping them turn data into valuable insights, predictions and products. From cloud automation to machine learning, Pythian designs, implements and supports customized solutions to the toughest data challenges.

Pythian

Agenda

- Types of backups
- Tools
- Best practices
- Questions

Pythian

Types of backup

Types of backups

Why backups?

- Accidents do happen
  - Disaster Recovery
- Not high availability
- Recovery testing
  - no testing == no backups
  - preferably automated

Pythian

Types of backups

Logical backups

- *"A backup that reproduces table structure and data, without copying the actual data files."*
  https://dev.mysql.com/doc/refman/8.0/en/glossary.html#glos_logical_backup
- Logical backups are typically human readable
- Flexible
- Easy to implement
- Slow

Pythian

Types of backups

Physical backups

- Copy of the files on disk, mostly not human readable
- Cold backups are easy
    - stop MySQL
    - copy the files
    - start MySQL
- Hot backups need more specialized tools to guarantee consistency

**Pythian**

Types of backups

Impact

- Recovery Point Objective (RPO)
    - The point in time to where you want / need to be able to restore your data when disaster strikes.
    - Typically you want this to be as small as possible because this might imply data loss
- Recovery Time Objective (RTO)
    - The time it can take to recover your database service to be operational
    - Typically you want this to be as small as possible because the longer it takes, the longer your application/service will be down

Pythian

Types of backups

A picture says more than a 1000 words
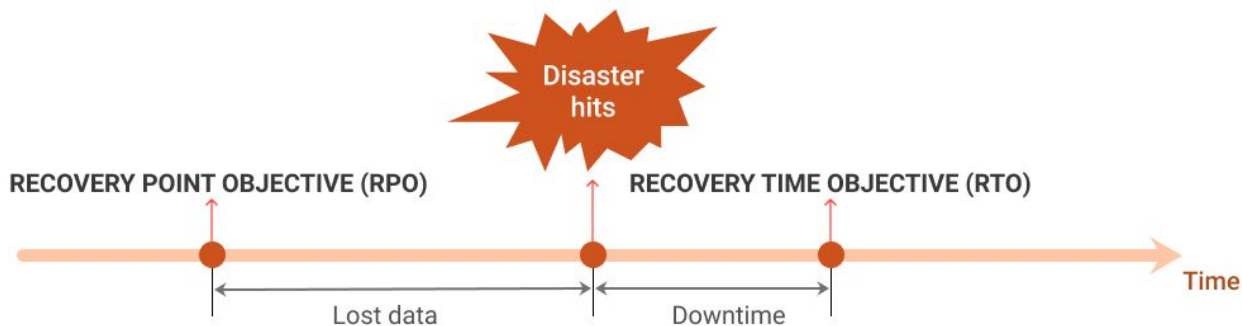
Types of backups

Full backup vs Incremental backup

- In a full backup you copy all data that is in the database. Restores are easy as they will have no prerequisites.
- In an incremental backup you copy only the changes since a previous (base) backup.
    - base backups can be full or incremental backups depending on the tool you use
    - restores are more complex as you need to follow the correct order in which to restore them and follow the correct procedure for the tool you use.

Pythian

Types of backups

Binary log backup

- Binary logs contain a stream of changes that happen on the database.
- These logs are used for replication to transfer the changes on the source (formerly known as master) to the replica.
- Binary log events have timestamps attached to them and the can be used for point-in-time recovery to significantly lower the RPO.
- Backup up binary logs in real time (streaming them somewhere) really helps.

Pythian

Tools

Tools

Cold backups

- For creating cold backups there are no specific tools required
- You need to stop MySQL
- For faster recovery on InnoDB tables
  - preferably stop MySQL cleanly so there is no need for crash recovery at startup
  - lower `innodb_max_dirty_pages_pct` (monitor `Innodb_buffer_pool_pages_dirty` until it has decreased enough)
  - set `innodb_fast_shutdown = 0`
- Copy the datafiles to a backup medium using the tools for your OS
- Start MySQL again

Pythian

Tools

mysqldump

- Deployed along the MySQL client packages
- Logical backup tool
  - Will select all data and writes it to `stdout` as an INSERT statement
- Is able to create per schema or per table dumps
- Supports all storage engines
  - For transactional engines (such as InnoDB)
    - `--single-transaction` option
  - For non-transactional engine
    - requires a global lock to create a consistent backup

Pythian

Tools

mysqlpump

- **mysql p**arallel **dump**
- rewrite of the mysqldump tool to support parallelisation (multiple threads)
- default parallelism is 2 (but configurable)
- also part of the default mysql client tools

Pythian

Tools

mydumper / myloader

- Open Source tool for multi-threaded logical backup
- Supports consistent backup with different storage engines
- When backing up non-transactional (MyISAM) tables you still need a lock across those tables
    - mydumper as has a `--less-locking` option to prevent requiring a long lock on your transactional tables
- dumps an sql file per table
- `myloader` to load the backup back

Pythian

Tools

mysql-enterprise-backup

- The original tool for creating online (hot) physical backups for MySQL
- Requires an Enterprise license
- Works by copying the datafiles on hand and copying the stream of changes in the redo logs.
- Once all InnoDB tables are copied, a lock is issued for non-transactional tables
- Non-transactional tables are copied
- A short full lock is issued to grab a consistent state of the database
- All lock are release

Pythian

Tools

xtrabackup

- Open Source tool developed by Percona
- Works in a very similar way as mysql-enterprise-backup
- requires no license
- Follows the MySQL release cycles but they need time to test and develop features if MySQL releases a backwards incompatible change: **check compatibility before upgrading your MySQL version!!**
- Use the correct version for your database version:
  - MySQL 5.6 (deprecated) => xtrabackup-2.3
  - MySQL 5.7 => xtrabackup-2.4
  - MySQL 8.0 => xtrabackup-8.0

Pythian

Tools

Disk snapshots

- Many modern storage solutions and most cloud providers provide a way of snapshotting your volumes. In many cases these snapshots are mostly instantaneous copies of your data volumes.
- This works too for MySQL but you'll have to script some steps:
    - To ensure recoverability you will need to issue a `FLUSH TABLES WITH READ LOCK;` to stop all writing and flush all tables to the disk
    - While this lock is active dump your replication coordinates to the disk (`SHOW MASTER STATUS\G` and `SHOW SLAVE STATUS\G`)
    - Freeze the filesystem(s) (ex `xfs_freeze`)
    - Create the snapshot using the tools provided by your hardware vendor or cloud provider
    - Unfreeze the filesystem(s) and `UNLOCK TABLES`.

Pythian

Tools

Others

- Other tools are available but in many cases these tools will utilise the before mentioned tools under the hood
    - holland-backup uses mysqldump or xtrabackup
    - commercial tools such as commvault include support for all of the physical backup tools we discussed
    - probably more tools are available…

Pythian

Best practices

Best practice

Disclaimer

- These best practices are from what I have seen or experienced
- Feel free to disagree!
- Mostly: use common sense!

Pythian

Best practices

#1 Backup your binary logs

- Having a copy of your binary logs helps significantly to lower your RPO.
- Stream your binary logs to another server or a network-file system.
- Binlog-servers were a thing a while ago but they are certainly usable as a binlog-backup solution
- `mysqlbinlog` tool (included with the MySQL client tools) can also stream binary logs:

```
mysqlbinlog --read-from-remote-server
--host=<mysql-hostname> --raw --stop-never
--start-position=4 --result-file=binlogs/backup-
localhost-bin.000001
```

Pythian

Best practices

#1 Backup your binary logs

```
mysql> SHOW BINARY LOGS;
+----------------------+-----------+-----------+
| Log_name             | File_size | Encrypted |
+----------------------+-----------+-----------+
| localhost-bin.000001 |      5246 | No        |
| localhost-bin.000002 |      1834 | No        |
+----------------------+-----------+-----------+
2 rows in set (0.00 sec)

[root@localhost ~]# ls -hl binlogs/
total 12K
-rw-r-----. 1 root root 5.2K Apr 24 16:04 backup-localhost-bin.000001
-rw-r-----. 1 root root 1.8K Apr 24 16:04 backup-localhost-bin.000002
```

Pythian

Best practices

#2 Incremental backups increase your RTO

- Having incremental backups can make sense if your write volume is really really high...
- But generally incremental backups make your restore procedures more complex
- Restoring a full backup and apply the binary logs backups usually has the same effect as incremental backups.

Pythian

Best practices

#3 Full backups and binary log point-in-time recovery can reduce RPO to 0

- A successful full backup that is recent enough combined with binary log up until the time of the failure can reduce your RPO to 0
- Procedure
  - Restore the full backup
  - Identify the point in time of the failure (ex the developer ran a TRUNCATE TABLE) from the binary logs and record the file and position of that event
  - Use `mysqlbinlog` tool to feed the contents of the binary logs into the restored backup copy and add `--stop-position=<end log position of the previous event>`

Pythian

Best practices

#4 There is no one-size-fits-all solution

- in most cases you will want to use a combination of backup solutions
- physical backups are great for full restores of a server but if you need to restore a single table it is often a very lengthy process for a task that would be much easier with a logical backup
- if possible make single table backups with a tool like `mysqldump` / `mydumper` if you regularly get asked for doing table restores

Pythian

Best practices

#5 use a replica to create backups

- as described in the different tools there always is some kind of a (short) lock required to get a consistent copy of the data. On a busy master server if might be hard to get that lock and if you get it, it will interrupt normal operations
- when backing up non-transactional tables, or when you want to create a consistent dump of different tables you can issue a
  `STOP SLAVE [SQL_THREAD];`
  in an ideal world you are not writing to the replica so stopping the SQL thread from applying changes has the same effect as running a
  `--single-transaction` (and it also works for non-transactional tables)

Pythian

Best practices

#6 off-site backups

- When disaster strikes, it might strike hard
- Make sure to keep a copy of your data in an offsite location
- Cloud Object Storage solutions (AWS S3, Google Cloud Storage, …) can be very helpful in this regard.
- When you need even more guarantees there are also manage services such as for example Iron Mountain

Pythian

Pythian

Questions?

Pythian

Thank you!

we are hiring for
MySQL, MongoDB, PostgreSQL, ...
https://pythian.com/careers/