Best Practices for a Comprehensive Business Continuity Plan with MariaDB

Jens Bollmann, Principal Customer Engineer
Santiago Lertora, Customer Engineering Manager

Backup Concepts

Loosing data szenarios

Some examples of potential impacts to a company

- power outages
- hardware damage
- cyber attacks
- flooding, tornados and fire
- theft
- etc.

Balance budget and resources to mitigate the risk

Important for your business

MariaDB

Recovery Requirements Determine Backups

Three important terms define backup and recovery systems for organizations:

- Recovery Time Objective (RTO)
- Recovery Point Objective (RPO)
- Risk Mitigation

Recovery Time Objective (RTO)

The amount of time that may pass during a disruption before it exceeds the maximum allowable time specified in the Business Continuity Plan

- How long does it take you to get recovered from previous backups
- This includes the entire recovery time for all systems

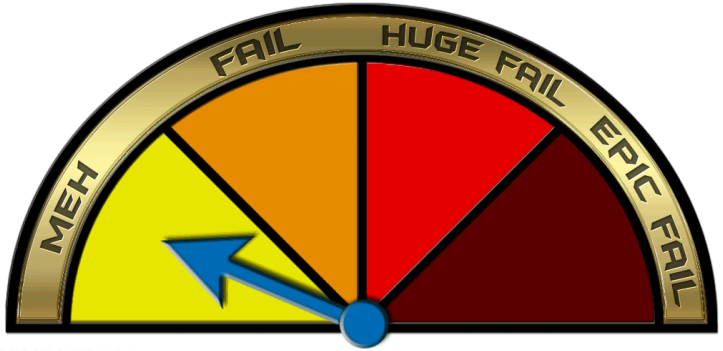
This can be reduced with DR replication and a global load balancer or DNS failover or SkySQL.

MariaDB

Recovery Point Objective (RPO)

Duration of time and service level within which a business process must be restored after a disaster in order to avoid unacceptable consequences associated with a break in continuity

- “How much data can I lose?”
- When was the last backup before failure?



MariaDB

Achieving RTO or RPO close to zero

For a company it is very costly to have all RTO and RPO close to zero for all applications

- All RTO and RPO differ prioritised on individual applications
- company expenditures to achieve
  - 100% RTO
  - 0 data loss on RPO

-> invest in redundancy

Risk Mitigation

What failure scenarios must the data be protected against?

Risk Mitigation

- Can help mitigate failure scenarios
  - (Multiple) Host Failure
  - (Multiple) Data Center Failure
  - Data Corruption or Loss
- Satisfy Legal Regulations
  - Legislation (GDPR)
  - Regulation
- Fulfill Industry or Legal Standards
  - PCI
  - HIPAA
  - et al.
- order of recovery

Retention

- Classic
  - One week on local server
  - Two weeks in local DC
  - remote for several months or annual backups
- Future
  - in cloud
  - with remote or local DC backups

Business Continuity with Backups

FULL, DIFFERENTIAL, INCREMENTAL

Full Backup Methods

Physical

- A copy of the entire database on disk
- Used to mitigate a single or multiple host failure
- Can build replicas
- Quick Full Recovery Time
- Slow to recover single row or table (user error)

- MariaDB Enterprise Backup

Logical

- Generates SQL files containing data that can regenerate a database
- Easily restore single row, table, or database
- Restore process automatically replicated
- Long Full Restore Time

- MariaDB Dump
- Mydumper

Block Level

- Can be remounted in seconds
- Binlogs are part of the snapshot so PITR still possible
- Encrypted by default by cloud provider
- Cloud provider block storage has over 99.99999999% durability

- Storage Snapshots

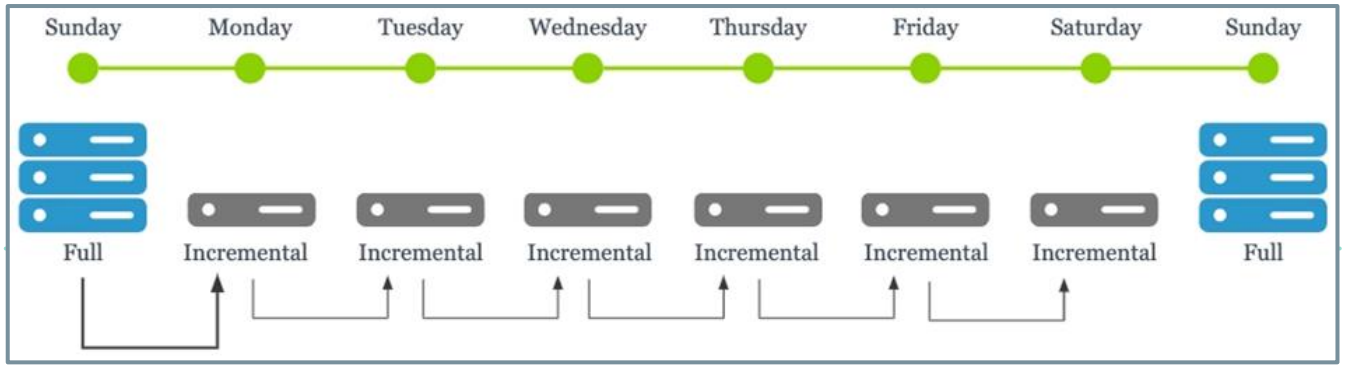
MariaDB

Partial Restore from Physical Backups

- You can import a tablespace from a MariaDB Backup made with `--export`
- May not be consistent with data in other tables
- Can be done in SkySQL

Incremental and Differential Backups

- Records difference since last backup
- Allows smaller backups, ie full backup on Sunday and backup changes since the last backup daily
- Risky if one incremental goes corrupt
- Available with MariaDB Enterprise Backup (physical backup) only
  - Default in SkySQL
- Negatively affects RTO
  - Take longer to restore



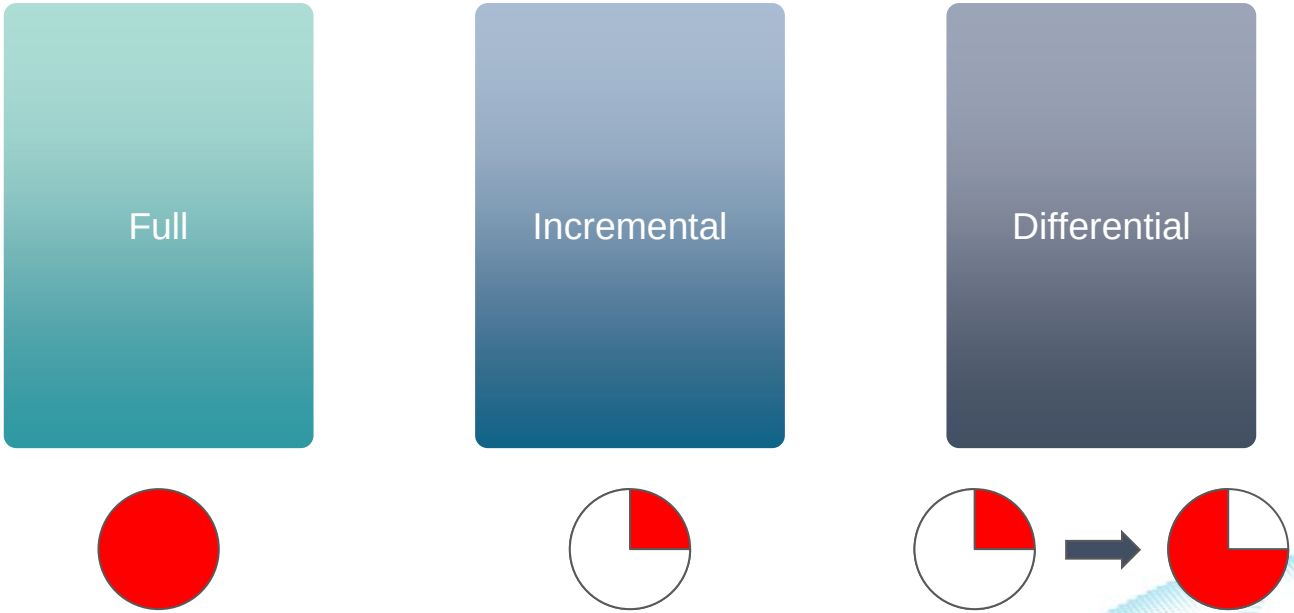
MariaDB

Monitoring

- Alerts on fails of backups
- Monitor duration and system impact of backups
- make recovery procedures visible

MariaDB

Backup Time



MariaDB

Restore Time

Full

Incremental

Differential

MariaDB

RECOVERY

Backups need recovery tests

- theoretically no backup available when not tested
- make sure this is done regularly

MariaDB

Optimal recovery test

- recovery on a similar target platform
- logical application test runs against the recovered system
- ensure availabilty of keys and passwords if encrypted

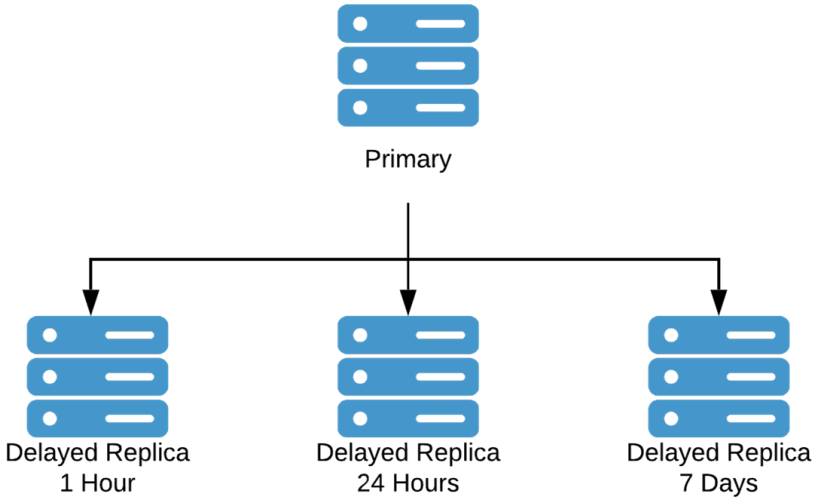
POINT IN TIME RECOVERY

Binary Logs

- Addresses RPO
- Logs should be rotated and backed up
- Combine with a Binary Backup to allow Point in Time recovery
- If you backup logs every hour, your RPO is 1 hour in case of a failure
- Can be reduced, but not to the point where the i/o and network traffic interfere with database operations
- Can stream logs for real-time RPO
- Adding complexity increases recovery time due to more complicated recovery procedures

Delayed Replicas

- A delayed replica should lag behind the primary by (at least) a specified amount of time
- Like a “time machine” with a database in a state as it was at a point in time to provide for fast RTO
- Downloads changes to local relay log but delays applying them
- Without a delayed replica, RTO of less than a few hours cannot be met easily.



MariaDB

Delayed Replica Types

- Including multiple intervals allows faster recovery of data within various points in history
  - A one hour delayed replica is critical when issues can be found quickly and rolling a 24 hour delayed replica forward may take longer than RTO allows.
  - Many times, a data error or other form of corruption is not caught within the first hour, this makes the 24 hour delayed replica critical.
  - Another option is to split the difference and replace both with a 6-12 hour delayed replica.
  - A multi-day delayed replica is also critical in cases where an error may not be fatal and may be caught much later. In those cases, it becomes necessary to recover data from days ago. Another example of this need is a failure late Friday that is not caught until Monday morning.

Delayed Replica Usage

- In Galera, each replica is an individual one node cluster, ready to become a primary of a multi-node cluster
- Delayed replication is enabled using the MASTER_DELAY option to CHANGE MASTER:
  - `CHANGE MASTER TO master_delay=3600;`
- When there is any production issue–regardless of if it is known to be database related–delayed replicas should be stopped immediately by issuing the `STOP SLAVE;` command on all delayed replicas.
- Once a production issue is resolved as not data related, `START SLAVE;` can be issued on delayed replicas to resume their operation.
- Ensure `relay_log_purse=OFF`

Delayed Replica Usage

- If a production issue is found to be a data related issue–for example corrupt, altered, or missing data–a delayed replica within the correct time frame should be selected.
- Use mariadb-binlog to find the date and time or binlog position of a bad statement
- `START SLAVE UNTIL …` command to have the replica resume until right before the data integrity was damaged
- A skip statement with `SET GLOBAL sql_slave_skip_counter = 1;` can be used to skip bad statements and then `START SLAVE SQL_THREAD;` can be issued after `CHANGE MASTER TO MASTER_DELAY=0;` to catch up to current transactions.
- The replica can be promoted to primary after issuing `STOP SLAVE; RESET SLAVE;.`

MariaDB

SkySQL solution

- skysql-backup
- recovery with backup manager
  - recovery on same
  - or another cluster
- PITR (point in time recovery)
  - possible if binlogs are part of the backup


MariaDB