MySQL Backup and DR - From Strategy to Execution
Speaker Bio

- **Name** - Shiv Iyer
- **Occupation:**
  - Founder and Principal of MinervaDB
  - Database Architect, Database Engineer, DBA, Data SRE and Data Ops. Geek
  - Open Source Database Systems Infrastructure Operations Expert
- **Technology focus:**
  - Open Source Database Systems - Performance, Scalability, High Availability and Database SRE
  - MySQL
  - MariaDB
  - InnoDB
  - RocksDB
  - PostgreSQL
  - ClickHouse
Database Systems Backup and DR

- Create a copy of data that can be recovered in the event of a primary data failure
- **Primary data failure events:**
  - Hardware or software failure.
  - Data corruption due to human error or application failure eventually causing data deletion.
  - Virus or other malicious attacks.
- **Backup policy:**
  - Frequency of Backup and DR efficiency - The more time passes between backup copies, the more potential for data loss when recovering from a backup.
- **Backup validation policy:**
  - Confirm backups are not corrupted, truncated or deleted.
  - Using MySQL replication with `pt-table-checksum`
MySQL Backup Strategies
Full Backup

● Full backup backs up the whole database.

● **Transaction log is included in full backup** - You can do full database recovery after the completion of full database backup restoration.

● Durability and reliability of the database full backup is up to the time backup finished.

● Storage intensive and time consuming.

● If you have a larger database, Combine full backup with differential backups.
Differential Backup

- Differential backup is based on the most recent and previous full data backup.
- Differential backup captures only the data that has changed since that full backup.
- Full backups, except for copy-only backups, can serve as the base for a series of differential backups, including database backups, partial backups, and file backups.
- The differential backups are most recommended when the subset of a database is modified more frequently than the rest of the database.
Incremental Backup

- Both differential and incremental backup does only backing up changed files.

- An incremental backup only includes the data that has changed since the previous backup, a differential backup contains all of the data that has changed since the last full backup.

- Incremental backup is the fastest backup type since it only backs-up increments.

- Expensive full restoration - Full database restore operations are slow. Because, you need the first full backup and all increments since then.
Why differential backup is faster compared to incremental backup?

- Restoration of differential backup never requires more than two backup sets. Incremental backups, on the other hand, could require a great number of backup sets.
- Faster recovery time. In differential backup, you need only a full backup and the latest differential backup to restore the entire data repository.
Most commonly used MySQL Backup tools and strategies

- `mysqldump`
- `mysqlpump`
- MySQL Enterprise Backup
- Percona XtraBackup
- MySQL Replication for HA and Disaster Recovery.
**mysqldump** - MySQL Logical Backup Solution

- `mysqldump` is a MySQL client utility which can be used to perform logical backups.

- The `mysqldump` generate output in SQL (default and most commonly used to reproduce MySQL schema objects and data), CSV, other delimited text or XML format.

- Flexible backup solution - You can tweak the output file before restoration. `mysqldump` is not a scalable DR solution larger MySQL databases, Replaying SQL is too expensive from both disk ops. and index building perspective.
Backup all MySQL databases with `mysqldump`

```bash
mysqldump --all-databases --single-transaction --quick --lock-tables=false > full-backup-$(date +%F).sql -u root -p
```

**mysqldump options used in the above script**

- `--single-transaction` when used in `mysqldump` will issues BEGIN SQL statement before dumping data from the server, dumps the consistent state of the database at the time when `START TRANSACTION` was issued without blocking any applications.
- `--quick`: To enforce dumping tables row by row, Recommended for very large MySQL databases
- `--lock-tables=false`: Do not lock tables for this backup session
Backup a specific MySQL database with mysqldump

```bash
grep -A 15 /usr/local/mysqldump-5.7.27/bin/mysqldump
```

**mysqldump** options used in the above script

- `--single-transaction` when used in mysqldump will issues BEGIN SQL statement before dumping data from the server, dumps the consistent state of the database at the time when `START TRANSACTION` was issued without blocking any applications.
- `--quick`: To enforce dumping tables row by row, Recommended for very large MySQL databases
- `--lock-tables=false`: Do not lock tables for this backup session
Backup a specific MySQL table with `mysqldump`

`mysqldump -u username -p --single-transaction --quick --lock-tables=false MDBMASTER tab1 > MDBMASTER-tab1-$(date +%F).sql`

**mysqldump options used in the above script**

- `--single-transaction` when used in mysqldump will issues BEGIN SQL statement before dumping data from the server, dumps the consistent state of the database at the time when `START TRANSACTION` was issued without blocking any applications.
- `--quick`: To enforce dumping tables row by row, Recommended for very large MySQL databases
- `--lock-tables=false`: Do not lock tables for this backup session
Restrictions of `mysqldump`

- `mysqldump` does not dump `performance_schema` or `sys schema` by default. To enforce dumping or logical backup of any of these schema objects, you have to explicitly mention them with the `--databases` option or if you want to just dump `performance_schema` use the `--skip-lock-tables` option.

- `mysqldump` does not dump the `INFORMATION_SCHEMA` schema.

- `mysqldump` does not dump the InnoDB `CREATE TABLESPACE` statements.

- `mysqldump` does not dump the NDB Cluster `ndbinfo` information database.

- `mysqldump` includes statements required to recreate the `general_log` and `slow_query_log` tables for dumps of the `mysql` database. But, log table contents are not dumped.
mysqlpump - Faster MySQL Logical Backup

- MySQL introduced a parallel logical backup utility with MySQL 5.7.8 - mysqlpump
- By default, mysqlpump uses one processing queue with two threads.
- To increase the number of threads, You can tune system variable --default-parallelism.
- Much better orchestration possible – You can backup selected databases, tables, stored programs and user accounts etc.
- By default mysqlpump will not backup performance_schema, sys schema, ndbinfo by default, You have to name them with --databases or --include-databases option
- mysqlpump does not dump INFORMATION_SCHEMA schema.
- Faster secondary indexes creation, The indexes created only after inserting rows!
Plain simple backup using mysqlpump:

[root@localhost mysqlpump2018-06-23-25-49]# mysqlpump -u root -p employees > employeebakup$(date '+%Y-%m-%H-%M-%S').sql

Enter password:
Dump progress: 1/4 tables, 0/630999 rows
Dump progress: 2/6 tables, 541250/3919384 rows
Dump progress: 4/6 tables, 1306627/3919384 rows
Dump progress: 5/6 tables, 2128435/3919384 rows
Dump progress: 5/6 tables, 3081685/3919384 rows
Dump completed in 5309 milliseconds

[root@localhost mysqlpump2018-06-23-25-49]#
Using mysqlpump based backup with 6 threads:

[root@localhost mysqlpump2018-06-23-25-49]# mysqlpump -u root -p employees --default-parallelism=6  > employeebackup$(date '+%Y-%m-%H-%M-%S').sql

Enter password:
Dump progress: 0/5 tables, 250/3477363 rows
Dump progress: 2/6 tables, 606250/3919384 rows
Dump progress: 3/6 tables, 1272103/3919384 rows
Dump progress: 5/6 tables, 2028185/3919384 rows
Dump progress: 5/6 tables, 2932185/3919384 rows
Dump progress: 5/6 tables, 3864185/3919384 rows
Dump completed in 5503 milliseconds

[root@localhost mysqlpump2018-06-23-25-49]#
Using mysqlpump to backup only selected databases, spawned 5 threads to backup employee and sakila database...
[root@localhost mysqlpump2018-06-23-25-49]# mysqlpump -u root -p employees --parallel-schemas=5:employees,sakila --default-parallelism=6  > backup$(date '+%Y-%m-%H-%M-%S').sql

Enter password:
Dump progress: 1/6 tables, 0/3919384 rows
Dump progress: 2/6 tables, 635250/3919384 rows
Dump progress: 3/6 tables, 1354353/3919384 rows
Dump progress: 5/6 tables, 2219935/3919384 rows
Dump progress: 5/6 tables, 3066185/3919384 rows
Dump completed in 5279 milliseconds
[root@localhost mysqlpump2018-06-23-25-49]#
Using mysqlpump to backup selected database and schema:

[root@localhost mysqlpump2018-06-23-25-49]# mysqlpump -u root -p --databases employees.titles > emp.titles$(date '+%Y-%m-%H-%M-%S').sql

Enter password:
Dump completed in 437 milliseconds
[root@localhost mysqlpump2018-06-23-25-49]#
Restore the backup from mysqlpump

Both mysqldump and mysqlpump generate MySQL logical backup in .SQL file so restoration is quite an straightforward process:

```bash
mysql -u root -p < backup.SQL
```
MySQL Enterprise Backup

- Proprietary hot / online backup tool for MySQL from MySQL Enterprise Edition of Oracle
- Transparent page compression for InnoDB.
- Backup history available for all members of Group Replication by making sure backup_history table is updated on primary node after each mysqlbackup operation.
- Storage engine of the mysql.backup_history table on a backed-up server has switched from CSV to InnoDB.
- MySQL Enterprise Backup 8.0.20 supports encrypted InnoDB undo logs.
- MySQL Enterprise Backup 8.0.20 supports high performance incremental backup by setting page tracking functionality on MySQL (`set –incremental=page-track`).
- Much better MySQL Enterprise Backup 8.0 troubleshooting with now mysqlbackup prints a stack trace after being terminated by a signal.
- Selective restores of tables or schema from full backup for Table-Level Recovery (TLR)
Percona XtraBackup - Most popular open source hot / online backup solution for MySQL / InnoDB
What makes Percona XtraBackup obvious choice for MySQL Backup and DR?

- Hot backup solution for InnoDB without blocking / locking transaction processing.
- Point-in-time recovery for InnoDB.
- MySQL incremental backup support.
- Percona XtraBackup supports incremental compressed backups.
- High performance streaming backup support for InnoDB.
- Parallel backup and copy-back support for faster backup and restoration.
- Secondary indexes defragmentation support for InnoDB.
- Percona XtraBackup support rsync to minimize locking.
- Track Percona XtraBackup history with Backup history table.
- Percona XtraBackup supports offline backup.
Percona XtraBackup - Best practices

- Databases really get really big, Plan compressed backup from day one (being conservative about storage is a good habit for a DBA / SRE)
- Percona XtraBackup uses qpress for compression, generates * .qp files

#Compressed Percona XtraBackup
xtrabackup --backup --compress --target-dir=/prod-idb-data/data-backups/

#Decompress the backup from Percona XtraBackup
xtrabackup --decompress --target-dir=/prod-idb-data/data-backups/

#Remove .qp files (2.3.7+ / 2.4.6+)
xtrabackup --decompress --remove-original --target-dir=/prod-idb-data/data-backups/
Percona XtraBackup - Incremental Backup

#Sunday Full Backup
xtrabackup --backup --target-dir=/prod-idb-data/data-backups/Sunday-full

#Monday Full Backup
xtrabackup --backup --target-dir=/prod-idb-data/data-backups/Monday-inc \
--incremental-basedir=/prod-idb-data/data-backups/Sunday-full

#Tuesday Full Backup
xtrabackup --backup --target-dir=/prod-idb-data/data-backups/Tuesday-inc \
--incremental-basedir=/prod-idb-data/data-backups/Monday-inc
Percona XtraBackup - Best practices

Generate metadata to restore the backup efficiently by adding --print-param option

# This MySQL options file was generated by XtraBackup.

[mysqld]
datadir = /infra2/databuild/mysql/
innodb_data_home_dir = /databuild/innodb/
inodb_data_file_path = ibdata1:10M:autoextend
innodb_log_group_home_dir = /databuild/innodb-logs/

P.S.- We recommend you to redirect this output into a file in the target directory of the backup
Maximum Database Availability and Reliability Solution Architecture for MySQL Infrastructure Operations
MinervaDB
Contact me

Email - shiv@minervadb.com

MinervaDB Toll Free - (844) 588-7287

Follow me on social media:

- LinkedIn - https://www.linkedin.com/in/thewebscaledba/
- Facebook page - https://www.facebook.com/DataOpsGeek/
- Twitter - https://twitter.com/thewebscaledba