



MySQL Implementation Essentials Bootcamp

MySQL Backup

John Kehoe
March 2025

Objectives



Backup Types

Backup Types: Advantages and Disadvantages

MySQL Enterprise Backup: Features and Benefits

Backup Types

Why back up?

Data is your most valuable IP. You must protect it from loss.

Backup:

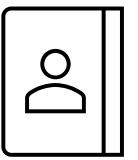
- > Ensures complete business continuity for disaster recovery
- > Helps in case of hardware, software, developer, or DBA errors
- > Is used in migration and upgrades (software and hardware)
- > Can be used to create replicas
- > Archives data (e.g., regulations that require to keep data for X number of years)
- > Moves chunks of data
- > Can be used to set up test environments

Backup and Recovery Can Be Trickier Than It Appears

- > Requires some forethought and planning
 - How do you back up your critical production systems?
 - Backups consume resources: When is the best time?
 - Down time is not acceptable
 - What is the proper backup storage?
- > Plan your backups, monitor backups, and verify their consistency
- > Plan to perform tests of your restore process
- > Integrate with existing document backup policies

Backup Types: Logical

- > Converts database and tables to SQL statements
 - Portable
 - Server must be running for export/import
 - Generally slower than physical backups



Tools:

- mysqldump
- MySQL Shell
- Data export

Backup Types: Physical

- > Based on the "raw image" of the DB file(s)
- > Require file stability during the backup
 - DB files must not be changed during the backup.
 - Otherwise, the restored server behaves as if it is restarting after a crash.



Tools:

- MySQL Enterprise Backup
- File system copies
- Snapshot
(LVM, HW, Virtualization)

Backup Types

> Others

- Activate and save binary logs for Point-In-Time Recovery (PITR)
- Use MySQL Replication for offline backups
- Use MySQL Replication with replica lag
- Use transportable tablespaces



Full and Incremental Backups

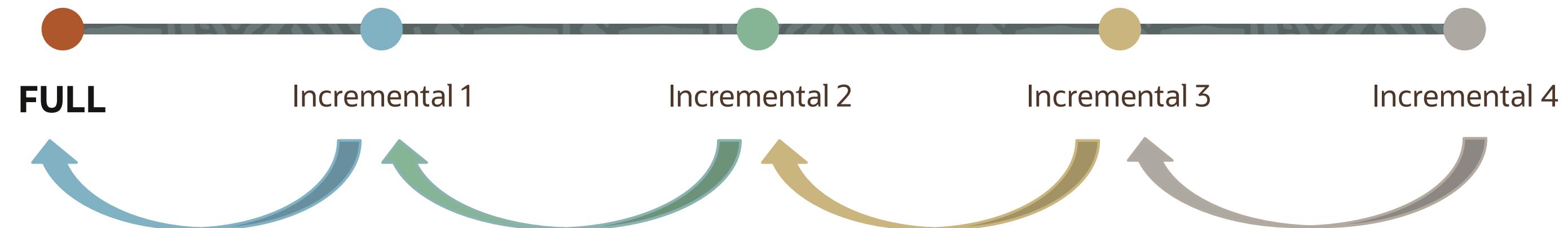


Full Backups

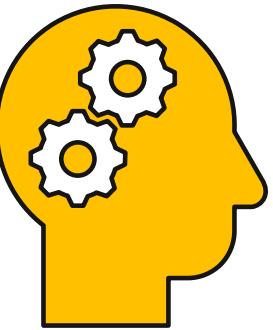
- A full image (physical/logical) of the database
- Restores a full copy of the DB

Incremental backups

- A backup image with the difference from the last point of backup that can be incremental or a full backup
- Restores **the last full backup** and **in the right sequence all the previous incremental backups (strict dependencies)** to the required restore point

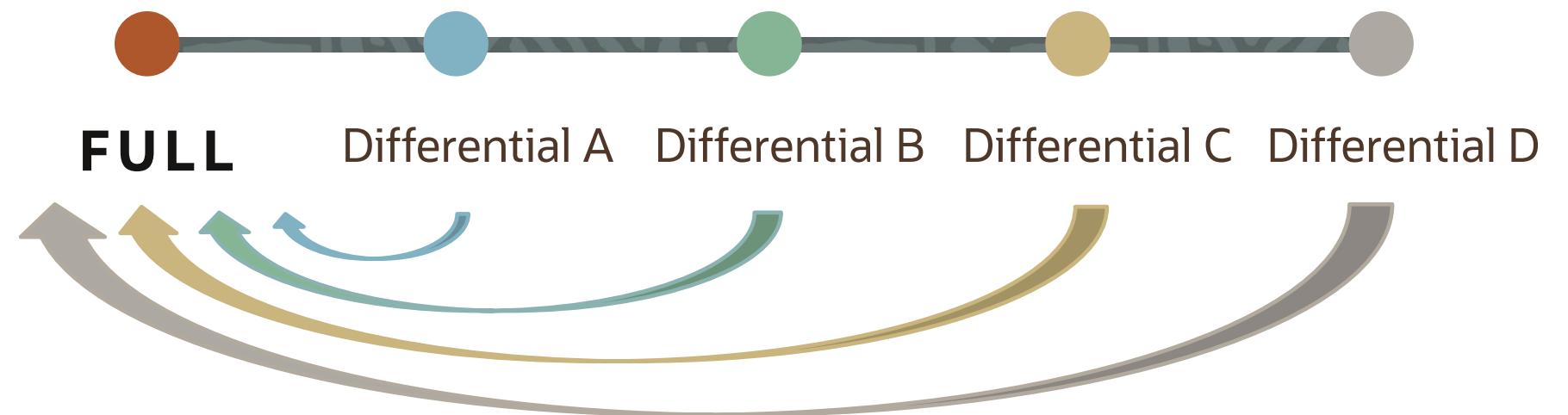


Full and Differential Backups



Differential backups

- A backup image with the difference from the last FULL backup
- Restores **the last full and the differential backups that you need**



How Does Backup Know Where to Start for Incremental & Differentials –It doesn't, it uses Binlogs and REDO

- > You must have a complete lineage of transactions from the last full backup for successful recovery.
 - **Binary Logs** Record all changes to the database and are required for PITR (be it Legacy Coordinates or GTID start/stop, as well the changes to the database)
 - **Redo Logs** Hold transaction state and position for InnoDB (LSN values) and record committed data changes, before writing to tables.
- **In order for successful, lossless recovery -The transaction information MUST exist.**
 - The Binary Logs contain their respective start and end points as well as the change vector
 - REDO Logs contain the committed data changed for InnoDB, that are unwritten to the table.



Backup Types: Advantages and Disadvantages – Part 1

Storage Snapshots



Advantages

- > Quick
- > Feature of Linux LVM, SAN, NAS, and virtualized environments
- > Good to use in conjunction with backups

Disadvantages

- > For consistency, it requires a service stop (or put it in read only)
- > It's a snapshot.
 - Still, a backup copy is required to be made – which can be “full” in size.
- > Performance may degrade with each concurrent snapshot.
 - Snapshots need to be released.
- > It has cross-file system limitations.

mysqldump

- > Create a SQL file that rebuilds the instance.
- > Advantages
 - **Good for small databases or tables**
 - **Produce a text file:** flexible and portable
- > Disadvantages
 - **Cannot back up an active instance** requires MVCC or table locks
 - **Not consistent across the backup** unless you use --single-transaction
 - **Not incremental or differential** requires a full backup every time
 - **Slower restore times than with physical backups**
- > Use mysql client to restore the backup.

<https://dev.mysql.com/doc/en/mysqldump.html>

mysqldump examples

- > Export of all databases for replication, including events and procedures:

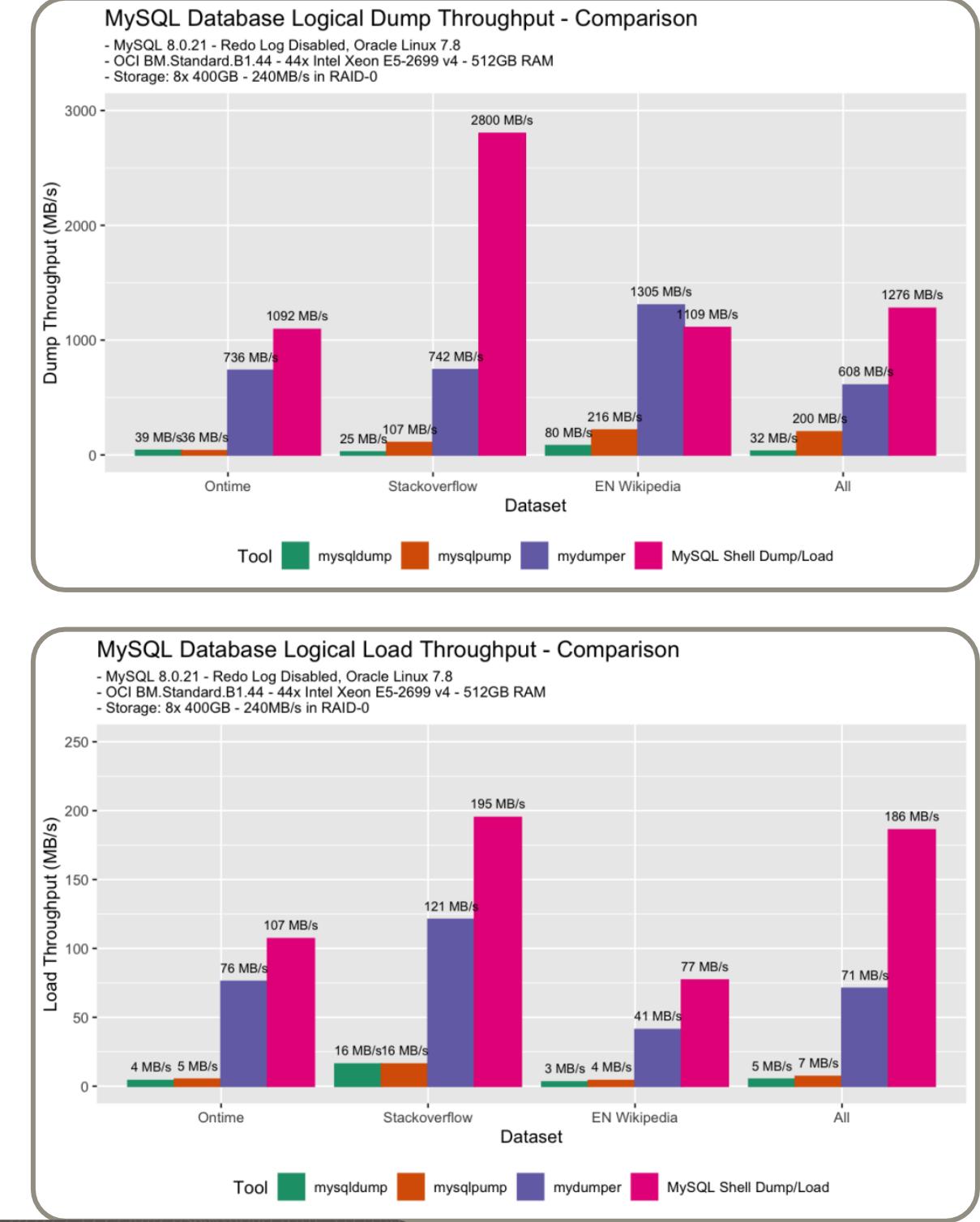
```
mysqldump [connection parameters] --all-databases \  
--single-transaction --routines \  
--events --source-data > full_backup.sql
```

- > Import of the dump:

```
mysql -u root -p -S /tmp/mysql_restore.sock < full_backup.sql
```

MySQL 8 Shell: Parallel Export/Import

- > Supports the dump and import of schemas or tables
 - Data streaming from remote storage
 - Parallel loading of tables or table chunks
 - Progress state tracking
 - Resume and reset capability
 - Add primary keys to tables
 - ...
- > Commands:
 - `util.dumpInstance()` : Dump an entire database instance, including users
 - `util.dumpSchemas()` : Dump a set of schemas
 - `util.loadDump()` : Load a dump into a target database



<https://dev.mysql.com/blog-archive/mysql-shell-dump-load-part-2-benchmarks/>

MySQL Replication and Backups

- > A replica is not a backup, but it can improve the backup policy.
 - Doesn't protect against human error if the error is propagated to the replica
- > Advantages
 - Rolling “snapshot”
 - Non-blocking
 - You can back up the replica to avoid overhead on the source instance
- > Disadvantages
 - Only latest point in time (point in time keeps moving forward)
 - Not historical
 - Not for archival purposes

Backup Types: Advantages and Disadvantages – Part 2

MySQL Enterprise Backup

Advantages

- > Physical backup and restore operations are very fast compared to logical backups
- > Flexible - many options
- > Archival
- > Scalable
- > Consistent
- > Supported

Disadvantages

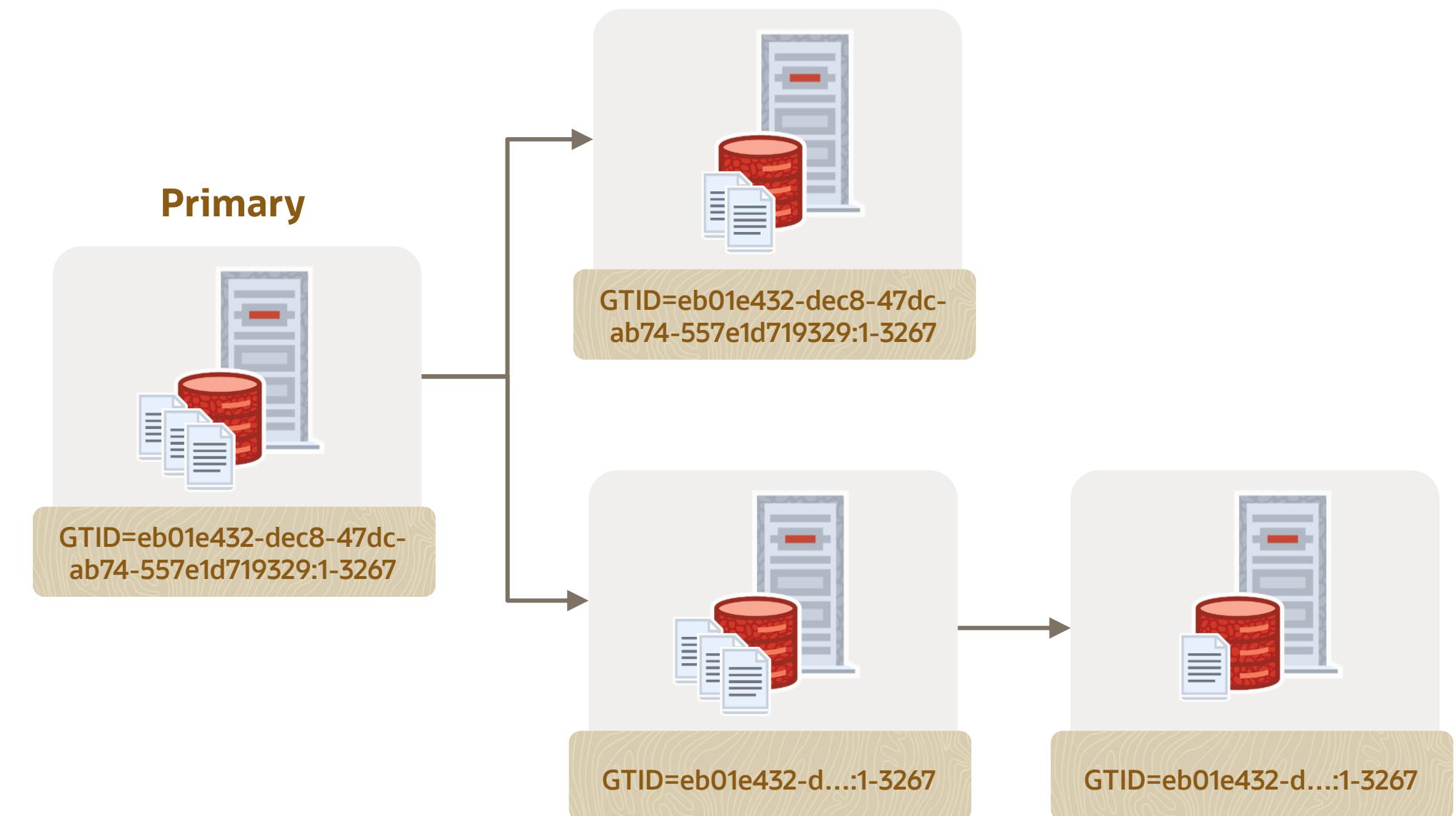
- > Cannot be used to perform upgrades
- > Not available in Community Edition

Binary Logs –Use Them!

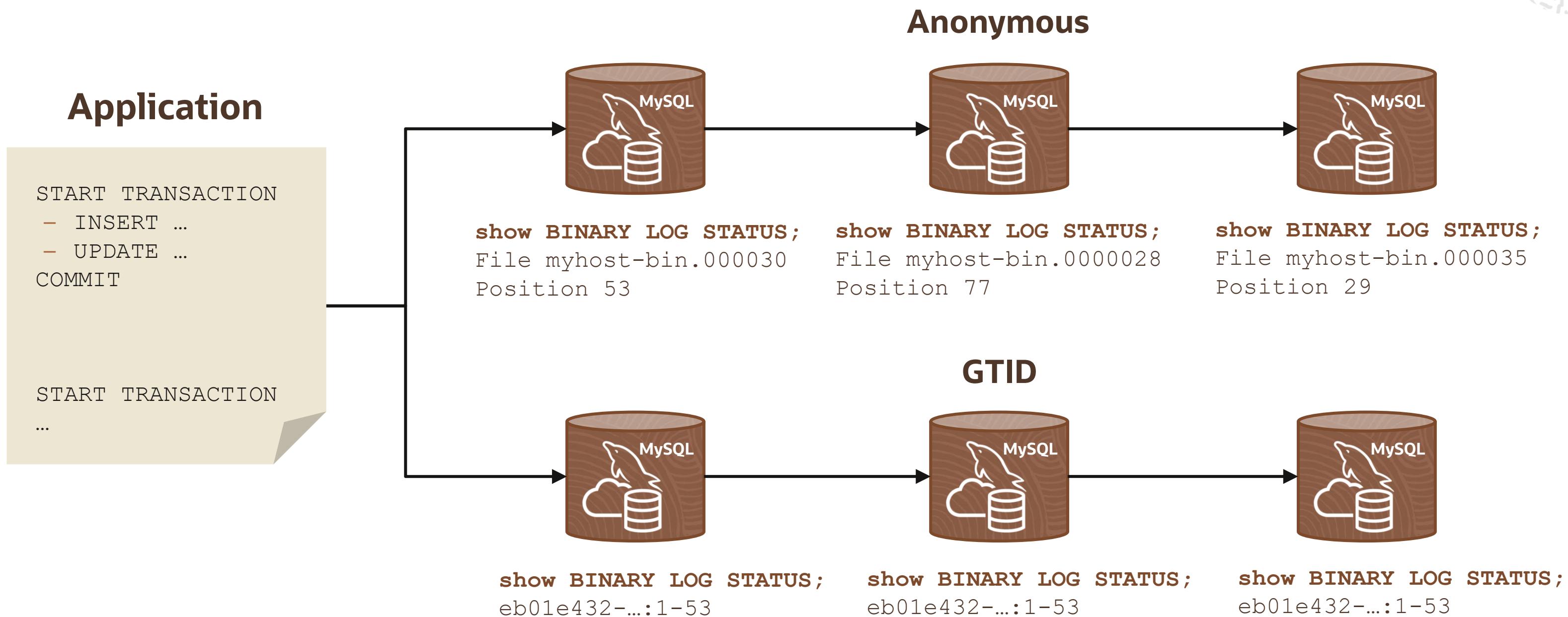
- > Set of numbered files containing changes
 - For example, binlog.000001, binlog.000002, etc.
- > Three possible formats
 - Statement-based replication (SBR): Propagation of SQL statements
 - Row-based replication: Logs column changes to individual rows
 - Mixed-format replication: SBR is default, but switches to row-based
- > Rotate/flush logs with FLUSH BINARY LOGS
 - Closes the logs but does not delete the logs
- > Purge can be manual or automatic
 - SHOW BINARY LOGS;
 - PURGE BINARY LOGS TO 'mysql-bin.000010';
 - PURGE BINARY LOGS BEFORE '2019-04-02 22:46:26';
 - binlog_expire_logs_auto_purge, expire_log_days, binlog_expire_logs_seconds
- > Can be non-transactional (legacy) or transactional (GTID)

Global Transaction IDentifier (GTID)

- > A unique identifier created and associated with each transaction committed on the server of origin (the source)
 - *<Server UUID>: <seq. number>*
 - The transaction has the same GTID in every node involved in replication topology.
- > Unique across all servers in a given replication topology



GTID vs. Anonymous Binary Log





Point-in-time Recovery: Process

- > Restore last backup.
- > Check the actual position (GTID/LSN).
 - mysql> SHOW BINARY LOG STATUS;
- > Execute events recorded in the binary using mysqlbinlog utility.
 - Specify all the binary logs in one command.
 - If possible, specify whenever recovery has to start or stop in terms of time or event position.
 - --start-datetime and --stop-datetime
 - --start-position and --stop-position
 - mysqlbinlog binlog_files | mysql -u root -p
 - As an alternative to mysqlbinlog, use replication to apply all available binary logs.
 - (KB 2009693.1 and KB 2277457.1)

How to Save Binary Logs

- > Make a physical copy of the files on a remote system:
 - Requires scripting: Ensure that the binary log file is closed when copied
- > Create a replica that has a copy of the binary logs using mysqlbinlog or replicas.
 - mysqlbinlog is not a daemon: monitor it!
 - <https://dev.mysql.com/doc/refman/8.0/en/mysqlbinlog-backup.html>
 - [KB 2180573.1](#)



Hands-On Labs

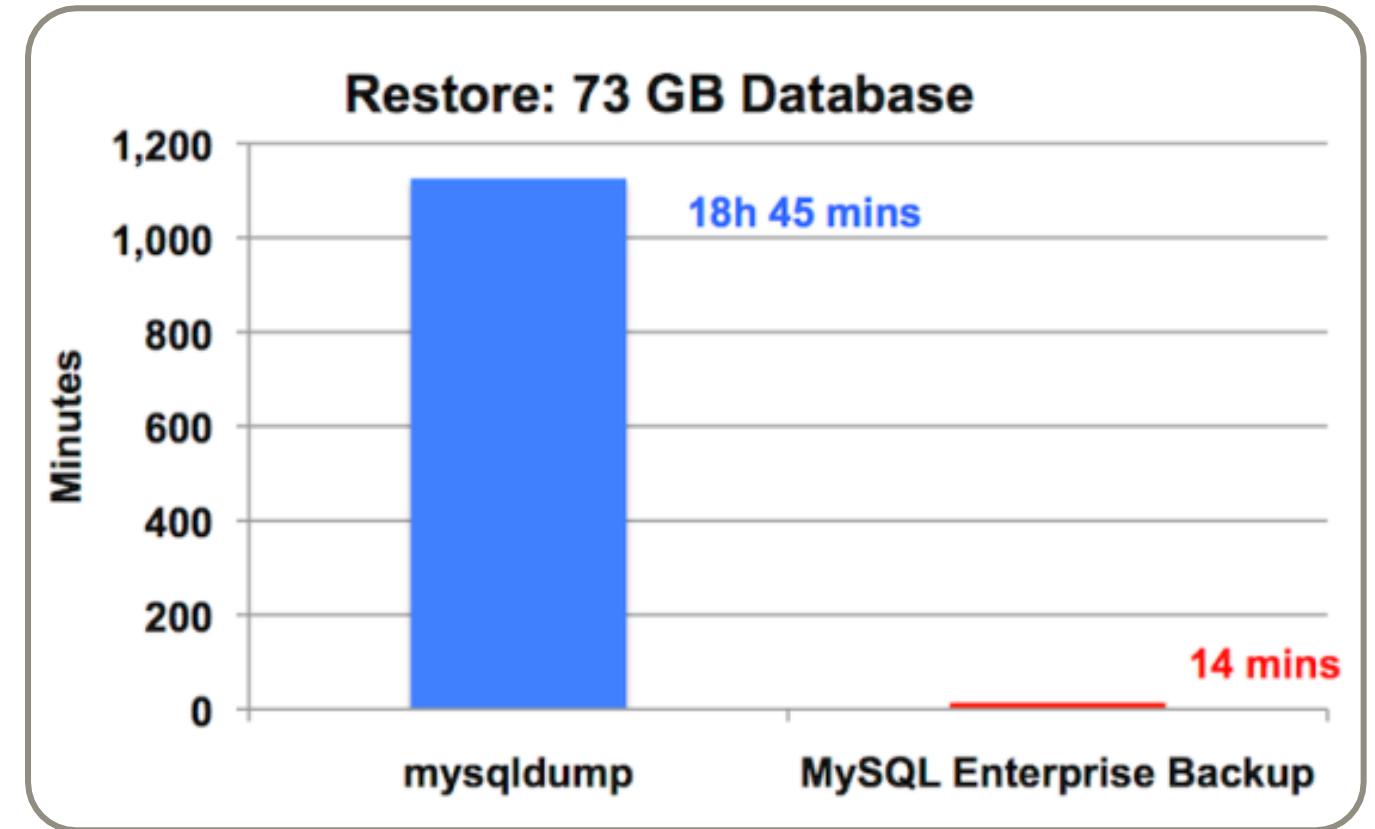
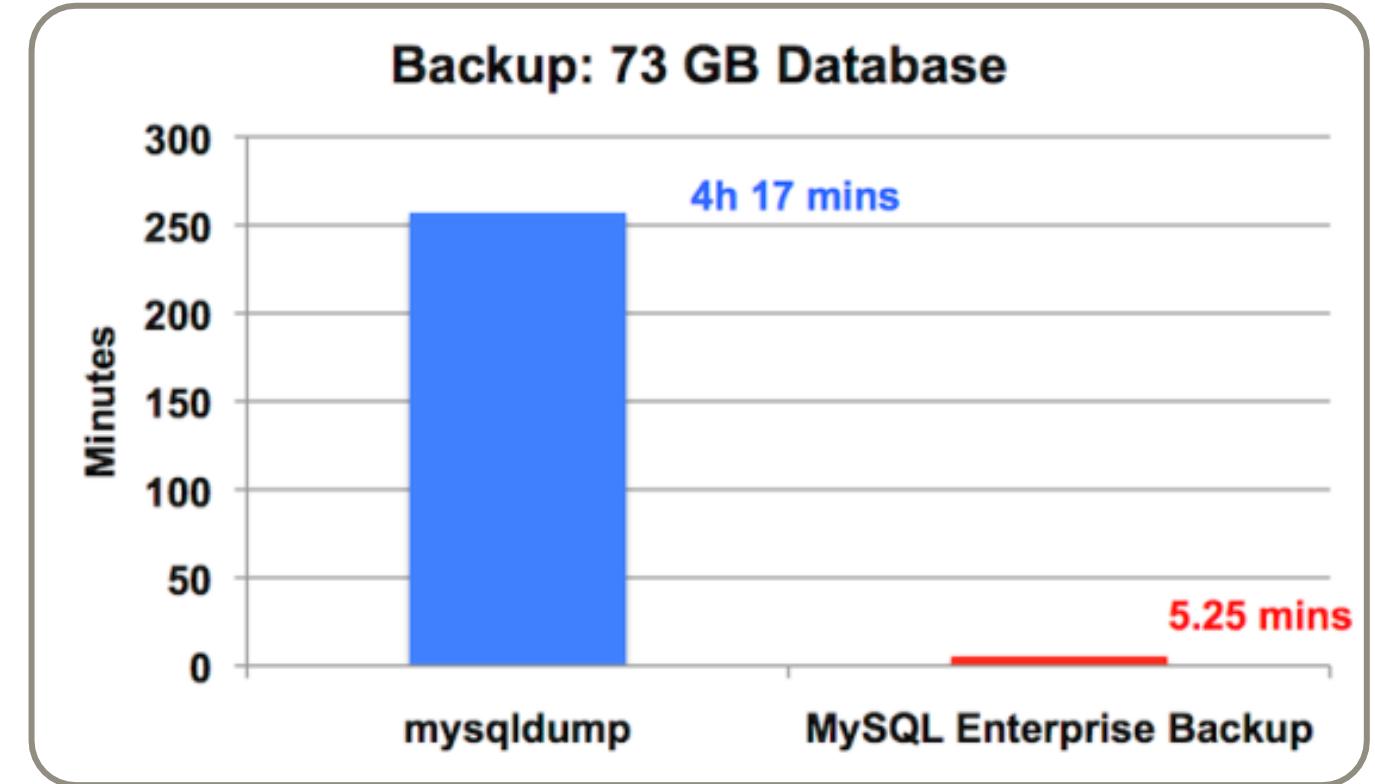


Lab 5a: MySQL Logical Backup

MySQL Enterprise Backup: Features and Benefits – Part 1

MySQL Enterprise Backup

- > Online, non-locking backup, and recovery
 - Complete MySQL instance backup (data and config)
 - Partial backup and restore
- > Supports direct-to-cloud storage backups
 - Oracle Storage Cloud, S3, etc.
- > Incremental backups
- > Point-in-time recovery
- > Supports compression and encryption
- > Supports *System Backup to Tape API (SBT)*
- > Optimistic backups



MySQL Enterprise Backup

- > Cross-platform (Windows, Linux, Unix)
- > Available as part of MySQL Enterprise Edition
 - Either with MySQL Server Enterprise Edition or as a separate package.
- > Download from:
 - <http://edelivery.oracle.com> (trial, only most recent release)
 - <http://support.oracle.com> (all releases)
 - And OTN for non-production use



MySQL Enterprise Backup: Instance Support

- > MySQL Enterprise Backup comes with both Long-Term Support (LTS) and Innovation Releases.
 - LTS: Supports only instances of the same LTS release
 - The last version available of MySQL Enterprise Backup is recommended.
 - Example: MySQL Enterprise Backup 8.4 (LTS) can back up MySQL Enterprise Edition 8.4, but not 8.0.37.
 - Innovation: Supports only instances within the same LTS release
 - Example: MySQL Enterprise Backup 8.7 (Innovation) can back up MySQL Enterprise Edition 8.4, 8.5, but not 8.0.

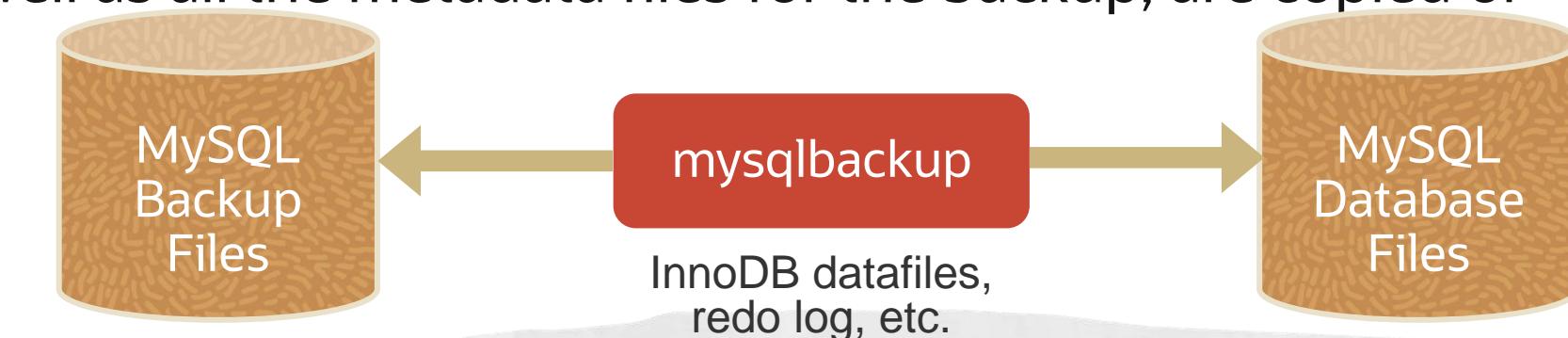


MySQL Enterprise Backup: Features and Benefits – Part 2

Backup Process

Short Summary

- > **step 1:** The InnoDB data files, redo log, binary log, and relay log files (except for the log files currently in use) **are copied into the backup**, while the database server operates as usual.
- > **step 2:** A **backup lock is applied** on the server instance.
- > **step 3:** The **FLUSH TABLES ... WITH READ LOCK** statement is applied on all **non-InnoDB** tables.
- > **step 4:** A **brief blocking** of logging activities on the server is applied, **to collect logging-related information** such as the current InnoDB LSN, binary log position, GTID, and so on.
- > **step 5:** The **read lock** on the non-InnoDB tables is **released**.
- > **step 6:** Using information from step 4, the relevant portion of the binary or the relay log file currently in use is copied.
- > **step 7:** The **backup lock** on the **server** instance is released.
- > **step 8:** The redo log files not yet copied before, as well as all the metadata files for the backup, are copied or created.
- > The backup operation is completed.



MySQL Enterprise Backup Files

- > MySQL Enterprise backup saves the datadir content.
- > Raw files are backed up with mysqlbackup.
 - InnoDB data
 - ibdata* files : Tablespace files
 - .ibd files : Per Table data files
 - ib_logfile* files : Log files
 - All other files
 - .MYD / .MYI : MyISAM data and index files
 - others : relay log, binlog, config files, etc. ...



Restore Process



1

Clean the destination directory.

2

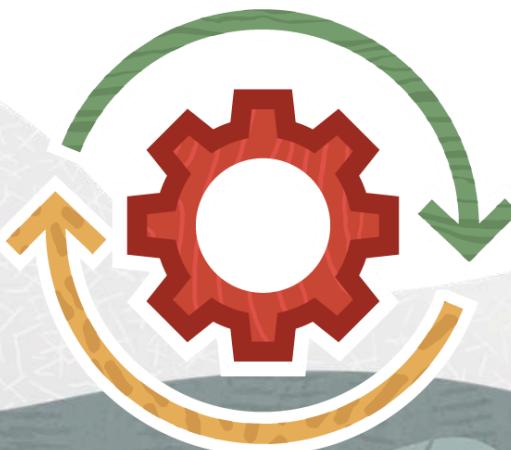
Restore the database using mysqlbackup with:

3

Depending on how you are going to start the restored server, you might need to adjust some files.

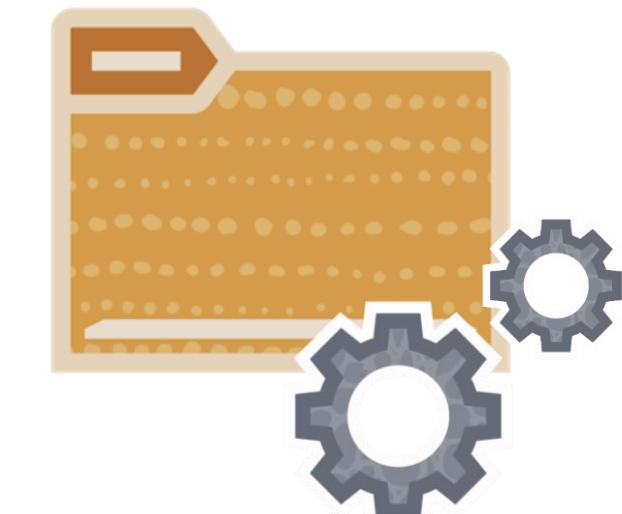
- > copy-back-and-apply-log or
- > copy-back

- > Ownership of the restored data directory
- > my.cnf/mi.ini configuration files
- > mysqld-auto.cnf
- > auto.cnf



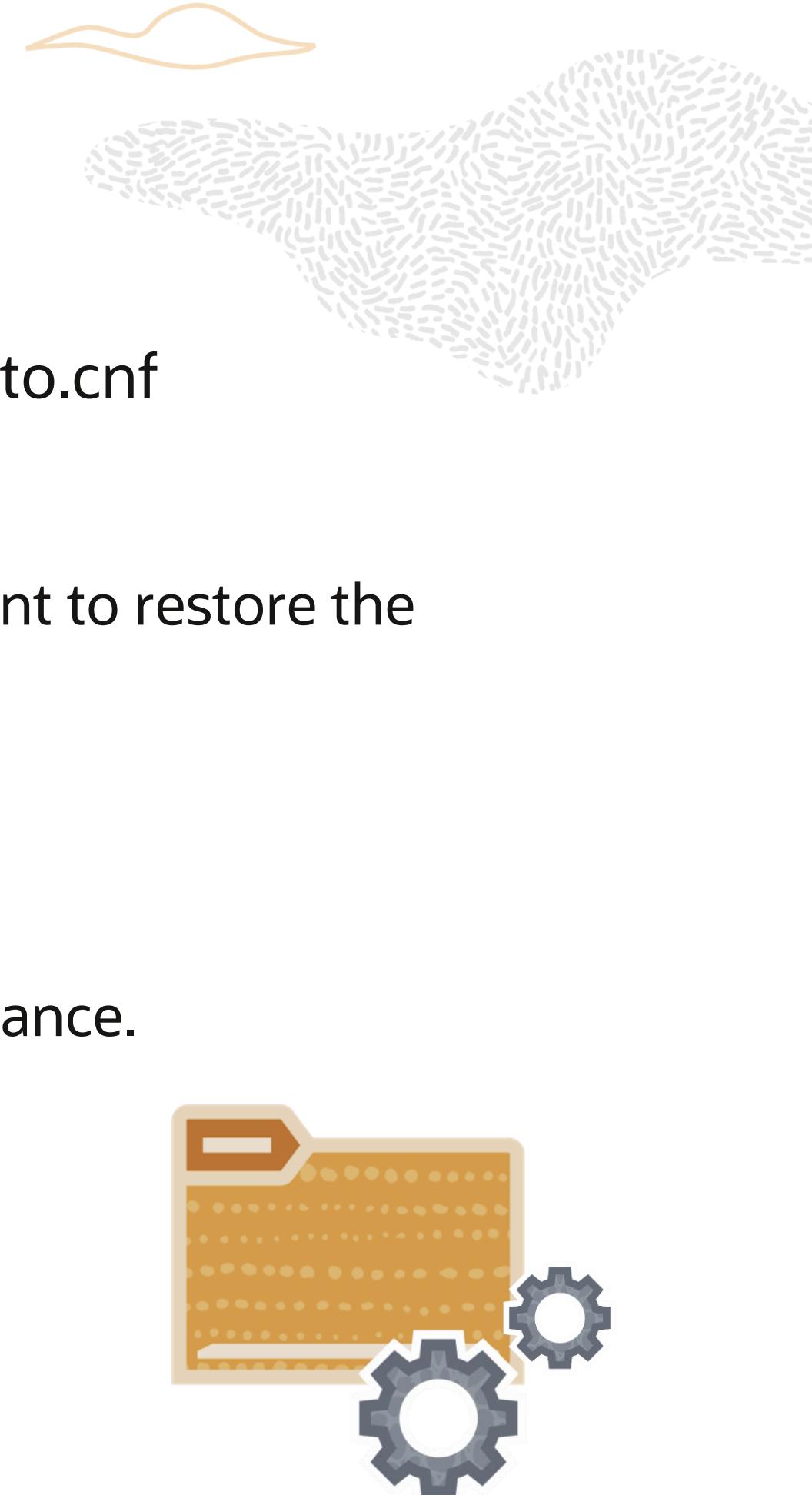
Configuration Files

- > It doesn't save the original my.cnf or my.ini or creates copies of the configuration file.
 - Use a standard OS backup for the original my.cnf and my.ini
- > It does creates configuration files for restoring.
 - backup-my.cnf: Records the crucial configuration parameters that apply to the backup
 - **server-my.cnf**: Contains values of the backed-up server's global variables that are set to non-default values
 - server-all.cnf: Contains values of all the global variables of the backed-up server



Configuration Files

- > Restore rename mysqld-auto.cnf to backup-mysqld-auto.cnf
 - It contains persisted variables.
 - It's important to restore the original name when you want to restore the original instance or create a replica.
- > Restore rename auto.cnf to backup-auto.cnf
 - It contains server UUID.
 - It's important when you want to restore the original instance.
 - Don't rename to create a new replica.



MySQL Enterprise Backup Options

Just some of the many options that can be used

> Encryption:

- --encrypt (--decrypt to restore)
- --key=<hex key string>
- --key-file=<keyfile>
- Generating key-file : e.g., openssl rand 32 -hex > mykeyfile
- It's only valid for images.

> Compression:

- --compress, --compress-level=[0 - 9]
- Only InnoDB data files are compressed.

> Using the SBT Interface:

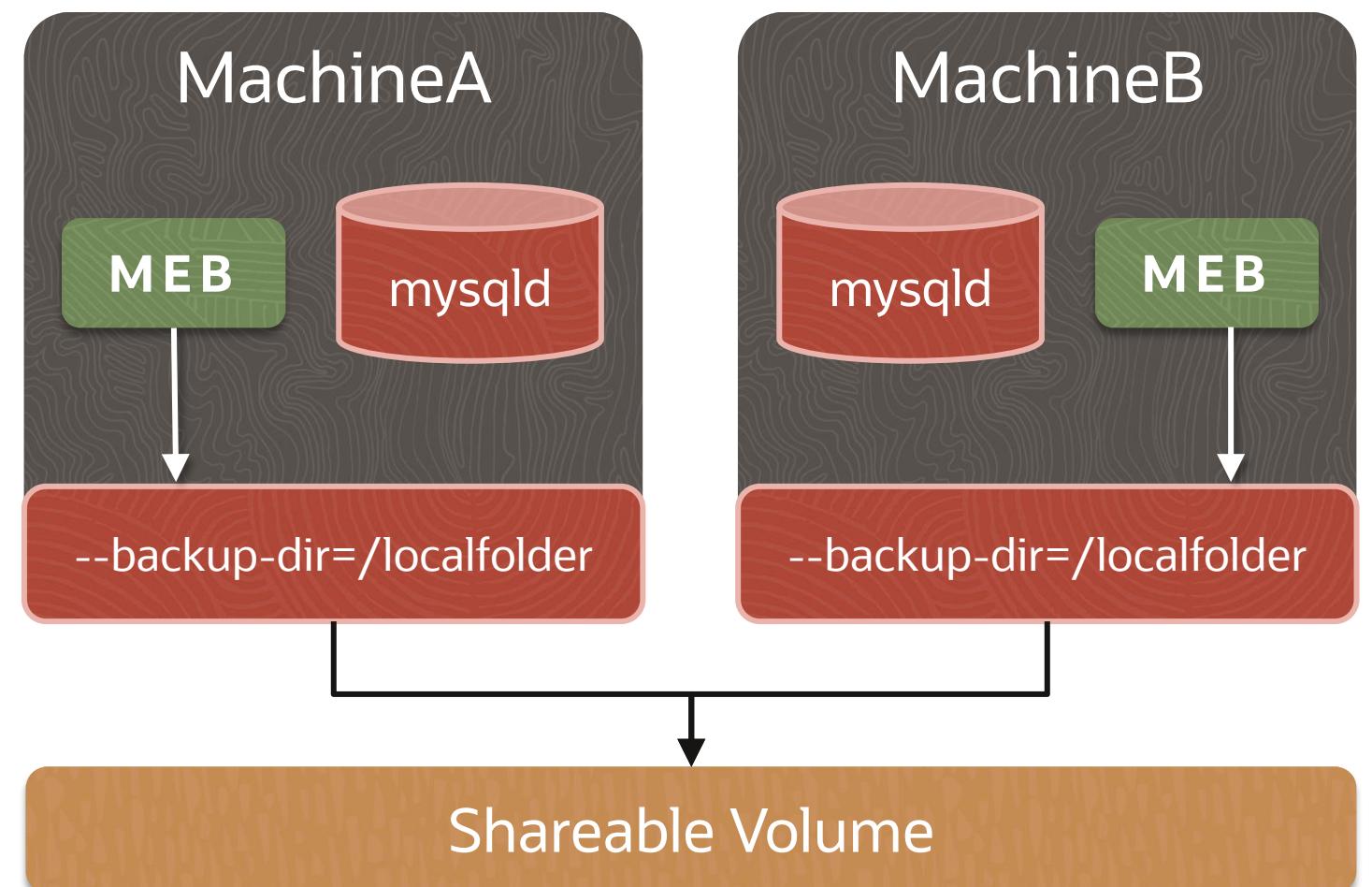
- Back up directly to your company's media management software (MMS).

More command line options

<https://dev.mysql.com/doc/mysql-enterprise-backup/9.2/en/mysqlbackup.usage.html>

MySQL Enterprise Backup Deployment – Example

- > Multiple MySQL Instances deployment
- > Back up to local folder/image file
 - Local folder: Specified with `--backup-dir`
 - The local folder can be mounted to external NFS or shared volume.
- > Timestamp
 - option : `--with-timestamp`
 - Create a folder with a timestamp and store multiple copies of a backup based on timestamp



Hands-On Labs



Lab 5b: MySQL Enterprise Backup

Summary

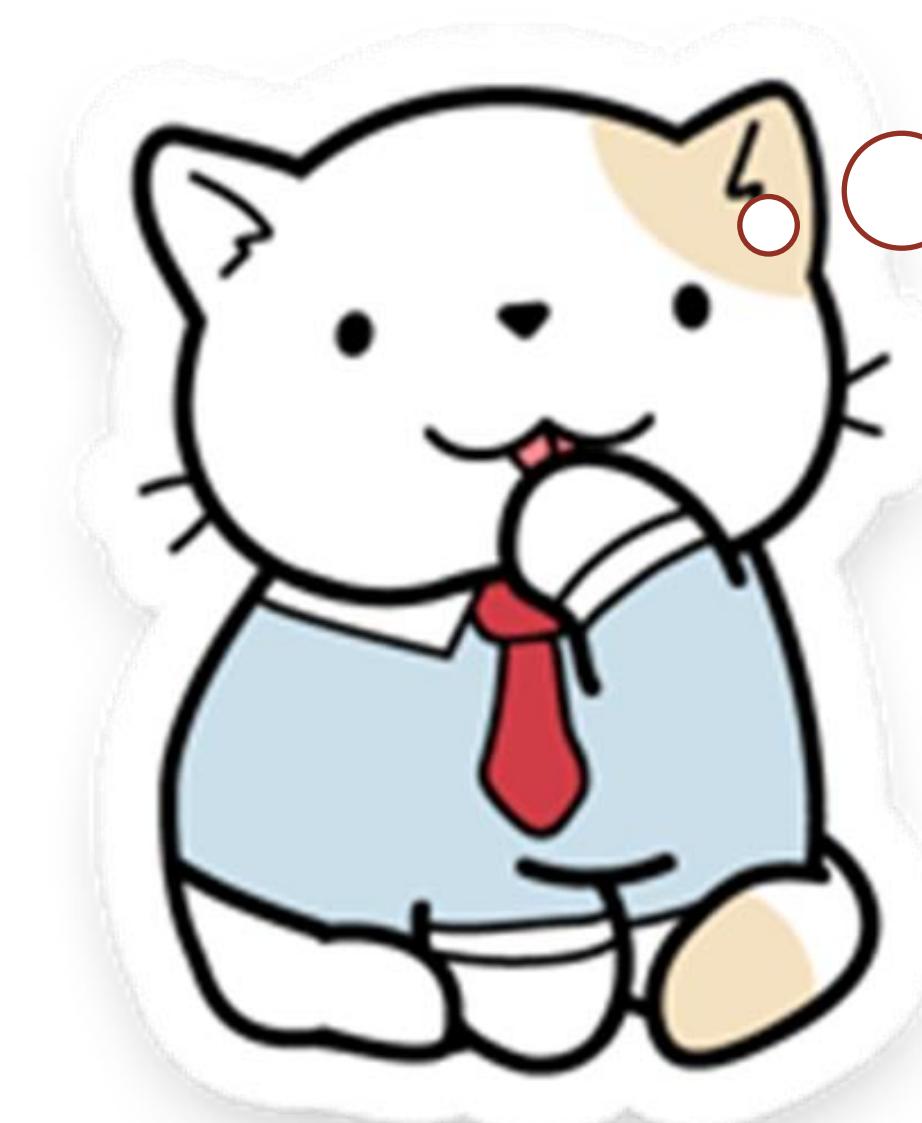


Backup Types

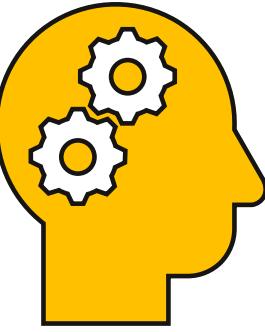
Backup Types: Advantages and Disadvantages

MySQL Enterprise Backup: Features and Benefits

Skill Checks



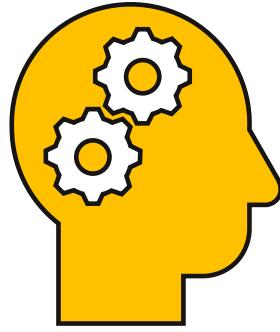
MySQL Backup – Skill Check



What makes a database backup effective ?

- Scheduling the backup
- Monitoring the backup for consistency
- Not exceeding limitations for the backup resources
- Being able to restore the backup data

MySQL Backup – Skill Check

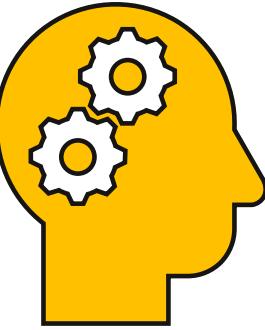


What makes a database backup effective ?

- Scheduling the backup
 - Monitoring the backup for consistency
 - Not exceeding limitations for the backup resources
- * Being able to restore the backup data**

Explanation: No backup is effective unless you can use it to restore your data. It is important not to just create backups, but to also regularly test the restoration process to ensure it works effectively.

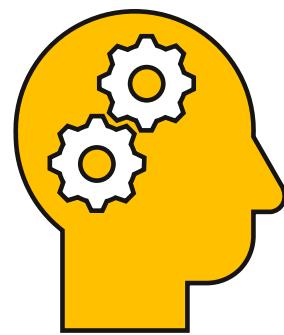
MySQL Backup – Skill Check



Which of the following statements is true about the mysqldump utility?

- mysqldump is a standard way to create physical backups
- mysqldump is excellent for backing up large databases
- mysqldump is fast and does not require locking tables
- mysqldump output is a human-readable text file with SQL statements

MySQL Backup – Skill Check



Which of the following statements is true about the mysqldump utility?

- mysqldump is a standard way to create physical backups
 - mysqldump is excellent for backing up large databases
 - mysqldump is fast and does not require locking tables
- * mysqldump output is a human-readable text file with SQL statements**

Explanation: The mysqldump utility has long been a standard way to create logical backups. It creates a script made up of the SQL statements that recreate the data and structure in a database or server. As a text file, it can be edited and managed by source code management systems.

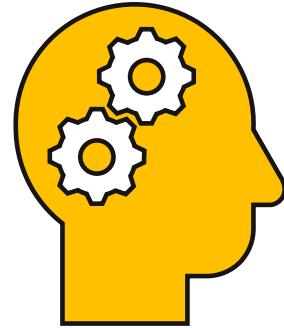
MySQL Backup – Skill Check



What is the MySQL Enterprise Backup utility designed for

- Create a snapshot of the MySQL storage medium
- Perform upgrades of MySQL systems
- Create Logical backup of MySQL systems
- Create Physical backup of MySQL systems

MySQL Backup – Skill Check

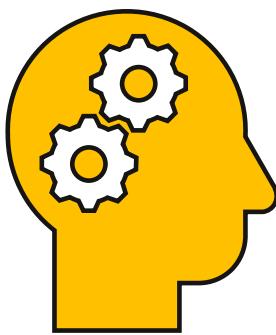


What is the MySQL Enterprise Backup utility designed for

- Create a snapshot of the MySQL storage medium
- Perform upgrades of MySQL systems
- Create Logical backup of MySQL systems
- * Create Physical backup of MySQL systems**

Explanation: MySQL Enterprise Backup is a utility designed specifically for backing up MySQL systems in the most efficient and flexible way. At its simplest, it performs a physical backup of the data files, so it is fast. However, it also records the changes that were made during the time it took to do the backup, so the result is that you get a consistent backup of the data at the time the backup completed.

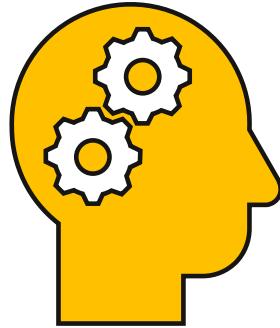
MySQL Backup – Skill Check



How does MySQL Enterprise Backup support optimistic backup?

- It locks up tables during the backup process
- It ignores data from busy tables
- It puts the database server in read only mode
- It records all data including data from busy tables

MySQL Backup – Skill Check

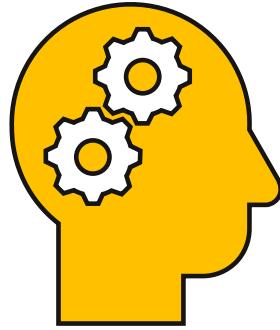


How does MySQL Enterprise Backup support optimistic backup?

- It locks up tables during the backup process
 - It ignores data from busy tables
 - It puts the database server in read only mode
- * It records all data including data from busy tables**

Explanation: MySQL Enterprise Backup supports optimistic backup. This process deals with busy tables separately from the rest of the database. It can record changes that happen in the database during the backup, for consistency. In a large dataset this can make a huge difference in performance.

MySQL Backup – Skill Check



What must you do to restore from a backup using MySQL Enterprise Backup ?

- Review the SQL script generated script to make sure it is valid
- Unzip the MySQL Enterprise Backup file
- Shutdown the MySQL database system
- Remove any previous files from the MySQL data directory

MySQL Backup – Skill Check



What must you do to restore from a backup using MySQL Enterprise Backup ?

- Review the SQL script generated script to make sure it is valid
 - Unzip the MySQL Enterprise Backup file
 - Shutdown the MySQL database system
- * Remove any previous files from the MySQL data directory**

Explanation: To restore from a backup using MySQL Enterprise Backup, you must first remove any previous files from the data directory. The restore process will fail if you attempt to restore over an existing system or backup.