

## Unit 9 – Formative Activity

Migrating a MySQL database to an OpenStack cloud with Percona XtraBackup is an excellent way to ensure data consistency and minimal downtime. This method leverages XtraBackup's ability to create hot, non-blocking backups, keeping your database online during the process (Nallapareddy, 2025).

### Migration Steps

The process is divided into three key phases: backup, synchronization, and final cutover.

#### 1. Initial Full Backup

First, install Percona XtraBackup on your local machine. Run the `xtrabackup` command to create a full, physical backup of your database. This process copies all data files without locking tables, so your applications can continue to write to the database. Once the backup is complete, use the `--prepare` option to apply the transaction logs. This prepares the data for restoration and ensures it's in a consistent state. Note the Log Sequence Number (LSN), as it's crucial for the next step. Finally, compress the backup and transfer it to your OpenStack environment using tools like `scp`.

#### 2. Data Synchronization with Replication

In your OpenStack cloud, launch a new VM and install MySQL. Stop the MySQL service and restore the backup by copying the data files to the MySQL data directory.

Configure replication by making the new cloud database a replica of your local one. Use the LSN from the backup to tell the replica where to start reading the binary logs from the source. Start replication and monitor the lag to ensure the new database catches up with all the changes.

#### 3. Final Cutover

Schedule a brief maintenance window. The actual downtime will be a matter of seconds.

During the window, stop all write operations to the local database. Wait for the replication lag on the cloud instance to reach zero, confirming that both databases are in sync.

Now, promote the new cloud database by stopping replication on it. Update your application's configuration to point to the new cloud endpoint. Perform a final verification and go live.

### Reflection

This method is highly effective because it uses a physical backup, which is significantly faster than a logical backup for large databases. The replication phase is the core of the minimal downtime strategy; it handles the ongoing changes, so the final switch is nearly instantaneous (Li *et al.*, 2023).

A key challenge is version compatibility between Percona XtraBackup and your MySQL server. Networking is another hurdle, as transferring a large backup file can be a bottleneck. Meticulous planning and testing are crucial for success, especially when configuring replication and OpenStack's security settings.

Overall, this migration strategy is a robust and widely used approach that effectively balances speed, data integrity, and business continuity (Anumula, 2025).

**Word count:** 424

**References:**

Anumula, K. (2025) 'Enhancing Oracle Database Migration with Cutting-Edge Techniques', *International Journal of Scientific Research in Computer Science Engineering and Information Technology*, 11(2), pp. 1022–1034. Available at: <https://doi.org/10.32628/cseit25112424>

Li, Z., Long, Z., Yu, Z., Diao, B. and Liu, J. (2023) 'Analysis of Network Resilience for Logical-Physical Dual-Layer Networks in Configuring Backup Paths', *2023 9th International Conference on Big Data and Information Analytics (BigDIA)*., pp. 189–195. Available at: <https://doi.org/10.1109/bigdia60676.2023.10429384>

Nallapareddy, C. (2025) 'Database Cloud Migration Strategy for a Business-Critical Application', *Journal of Technology and Systems*, 7(2), pp. 22–32. Available at: <https://doi.org/10.47941/jts.2635>