



Choosing your target: Community Postgres®, EDB PG AI or EDB PG AI Distributed High Availability

The architecture decision behind a MySQL® migration, driven by your availability topology and the capabilities you are replacing, including when the right answer is community Postgres.

A MySQL migration has a real choice of target

Not every source platform does. Migrating from Sybase, EDB Migration Toolkit targets EDB Postgres AI (EDB PG AI) only, so there is no decision to make. From MySQL 5.5, 5.7 or 8.0 it targets community PostgreSQL or EDB PG AI, which means the target is an architecture choice rather than a formality.

Treat the migration as a platform conversion, not a data copy. The target decision shapes the operational model you will live with for years, and it is much cheaper to make deliberately at the start than to revisit after cutover.

The answer is sometimes community Postgres, and we will say so

If what you need is the engine (better query planning, JSONB, extensibility, real constraint enforcement) and your team is comfortable owning high availability, backup and patching, then community PostgreSQL is a legitimate destination. Section 06 covers when reaching for EDB PG AI by reflex is the more expensive mistake.

Three targets, and what each one is for

TARGET	CHOOSE IT WHEN	WHAT YOU OWN
Community PostgreSQL	You need the engine's capabilities; your team has the operational maturity and wants to keep it; the capabilities you are replacing are in MySQL Community rather than Enterprise Edition.	HA design, backup and restore, patching, CVE response, tuning. Real work, done by your team.
EDB PG AI	You want enterprise features, EDB-supported tooling and a more opinionated operational stack around security, availability and manageability, particularly when replacing capabilities that sat in MySQL Enterprise Edition. One limit to plan for: the Enterprise Postgres (Oracle Compatible) edition, which is the server Migration Toolkit targets, doesn't currently support the enum data type, so MySQL ENUM columns move to CHECK constraints.	The workload. EDB supports the platform, the tooling and the migration path.
EDB PG AI Distributed High Availability	Your requirement is continuous write availability across regions: genuinely active-active, not single-primary with failover. The closest equivalent to Galera or Group Replication.	A distributed architecture, with the conflict-management design decisions that implies.

EDB PG AI Distributed High Availability should not be the default answer

It is the right answer when the business goal is active-active or geographically distributed write capability. It is the wrong answer for a straightforward single-primary replacement, where it adds architecture and conflict-handling complexity you did not need. If you are running Galera today, ask whether you use it for multi-primary writes or simply for availability: the answers point at different targets.

What you run today, and what it maps to

The topology you actually run points at a target before any product does.

MYSQL TODAY	POSTGRES EQUIVALENT	NOTE
Single instance, backups only	Single node, plus a real backup and PITR design	The migration is the moment to fix this rather than reproduce it.
Async primary / replica	Streaming replication, with EDB Failover Manager or repmgr for failover	The standard enterprise pattern. Build it first, migrate into it.
Galera or Group Replication, used for availability	Streaming replication with automated failover	If the multi-primary capability is not actually being used for writes, this is the simpler and better fit.
Galera or Group Replication, used for multi-primary writes	EDB PG AI Distributed High Availability	This is the case it exists for.
Binlog-driven downstream pipelines	Logical replication, or third-party CDC	Anything feeding analytics, search indexes or event streams needs a replacement design, on some estates a larger scope item than the database move itself.

No EDB tool reads the MySQL binary log

There is no EDB-side consumer for the MySQL binlog. EDB Replication Server and EDB Data Migration Service don't support MySQL as a source, so EDB has no first-party change data capture path from MySQL.

Inventory every binlog consumer before you scope the move. Each one needs its own replacement: Postgres logical replication after cutover, or a third-party CDC tool.

What are you actually replacing?

MySQL Community is genuinely free. Several capabilities enterprises rely on are in MySQL Enterprise Edition. Working out which edition you are on, and which of its features you actually use, is the most direct route to the target decision, and it is a packaging question rather than a judgment about the engineering.

Work through this list against your own deployment

Transparent data encryption

In use? Required by a control framework? This is frequently the single capability that decides the target.

Audit

Who reads the audit trail, how often, and what would satisfy an examiner on the new platform?

Enterprise backup

What is your current RPO and RTO, and are they met today or merely stated?

Monitoring and management tooling

What would your team lose on day one, and what replaces it?

Firewall and access controls

What sits between the application and the data today?

Commercial support

Who do you call at 3 a.m. now, and who would you call afterward?

The comparison worth running

Not free-versus-paid. Compare what Community MySQL gives you today against what an enterprise Postgres distribution includes. If you are already on MySQL Enterprise Edition, that is close to a like-for-like comparison. If you are on Community and have built around the gaps, the honest comparison includes the cost of what your team built and maintains.



Target first, migration second

Stand the target up as a production-shaped cluster before migrating into it. Cutting over into an architecture that is already resilient is materially less work than retrofitting resilience under a live system, and it means the migration is tested against the configuration you will actually run.

1 Build the cluster

Streaming replication with EDB Failover Manager or repmgr for primary/standby, or the EDB PG AI Distributed High Availability topology if section 02 pointed there. Production-shaped, not a single node you intend to fix later.

2 Stand up backup and DR on day one

Barman or pgBackRest. Configured before data arrives, not after.

3 Migrate into it

Per the technical playbook. The migration lands in the real architecture.

4 Prove three things

Before go-live, not after, check these three things in this order:

- a. That backups complete.
- b. That WAL archiving works.
- c. That a full restore plus point-in-time recovery has been tested on the target platform. "Tested" means someone performed a restore and timed it, and the duration is written down. A backup that has never been restored is a hypothesis.



Three failure modes worth naming

Choosing EDB PG AI by reflex

If your team is operationally strong, your capability list in section 04 comes back thin, and what you want is the Postgres engine, community PostgreSQL may serve you better, and EDB can still help you migrate. Paying for an operational stack you will not use is a real cost.

Choosing community Postgres by reflex

The mirror image, and more common. The migration is the easy part; the years afterward are where HA design, patch cadence, CVE response and backup verification actually live. Be honest about whether that capacity exists and will be protected, not whether it could exist in principle.

Reaching for EDB PG AI Distributed High Availability because the source was clustered

Galera in the estate is not by itself evidence of a multi-primary write requirement. Many deployments run it for availability. Check what the application actually does before designing for distributed writes.

Four questions, in order

1

Which MySQL edition, and which features?

Section 04. This single answer resolves most of the target decision.

2

What is the real availability requirement?

Section 03. Distinguish availability from multi-primary writes: they point at different targets.

3

Who will own operations after cutover?

Named team, protected capacity. Not an aspiration.

4

What consumes the binlog today?

If anything does, it needs a replacement design, and that design belongs in the estimate now.

Request an assessment

An EDB migration architect will work through these four questions with you and give you a target recommendation with the reasoning attached, including when that recommendation is community Postgres. Request a migration assessment at enterprisedb.com, or contact your EDB account team.

enterprisedb.com



Target first, migration second.

Bring the four questions to an EDB migration architect, and expect a recommendation with the reasoning attached, including when the answer is community Postgres.

Sources

- EDB Migration Toolkit documentation, supported platforms and databases: MySQL 5.5, 5.7 and 8.0 migrate to PostgreSQL or EDB PG AI; SAP ASE 15.7 and 16.0 to EDB PG AI only (enterprisedb.com/docs/migration_toolkit/latest/02_supported_operating_systems_and_database_versions).
- EDB Migration Toolkit FAQ: enum data type support and the CHECK constraint workaround (enterprisedb.com/docs/migration_toolkit/latest/11_questions).
- EDB PG AI (enterprisedb.com/docs/epas/latest), EDB PG AI Distributed High Availability (enterprisedb.com/docs/pgd/latest), EDB Failover Manager (enterprisedb.com/docs/efm/latest) and repmgr (repmgr.org) product documentation.
- EDB Replication Server (enterprisedb.com/docs/eprs/latest) and EDB Data Migration Service (supported databases, Hybrid Manager documentation): MySQL is not a supported source for either.
- MySQL Community Edition and MySQL Enterprise Edition feature packaging per Oracle's published product documentation.

This guide is architecture planning guidance and does not constitute a contractual commitment or a performance guarantee. Product capabilities reflect EDB documentation at the time of writing (October 2026); verify against current documentation before committing to a design. MySQL and Oracle are registered trademarks of Oracle Corporation. Postgres and PostgreSQL are registered trademarks of the PostgreSQL Community Association of Canada and used with their permission. EDB Postgres® AI is a trademark of EnterpriseDB Corporation. © EnterpriseDB Corporation 2026. All rights reserved.

EDB Postgres® AI is the first open, enterprise-grade sovereign data and AI platform, with a secure, compliant, and fully scalable environment, on premises and across clouds. Supported by a global partner network, EDB Postgres AI unifies transactional, analytical, and AI workloads, enabling organizations to operationalize their data and LLMs where, when, and how they need them. For more information, visit enterprisedb.com.

