



SHOULD YOU PAY FOR YOUR OPEN SOURCE DATABASE?

TABLE OF CONTENTS

- 3 **WHEN IS IT TIME TO MOVE FROM COMMUNITY TO ENTERPRISE EDITION?**
- 4 **WHY WOULD I PAY FOR OPEN SOURCE?**
- 5 QUESTION 1: DO YOU HAVE A TEAM OF DATABASE EXPERTS AND ANALYTICS EXPERTS ON STAFF?
- 6 QUESTION 2: WHAT IS YOUR UPTIME REQUIREMENT AND RISK TOLERANCE?
- 9 QUESTION 3: HOW CONFIDENT ARE YOU THAT YOUR DATA IS SECURE?
- 10 QUESTION 4: ARE YOU CONCERNED ABOUT TAKING YOUR DATABASE INTO THE CLOUD?
- 12 **SO, IS IT TIME TO PAY FOR YOUR OPEN SOURCE DATABASE?**
- 13 **FEATURE COMPARISON: MARIADB COMMUNITY (FREE) VS. ENTERPRISE (SUBSCRIPTION)**

WHEN IS IT TIME TO MOVE FROM COMMUNITY TO ENTERPRISE EDITION?

How do you know if it's time to move beyond the free, community-supported version of your open source database? If your business hinges on the success of an application, you know this much: You need a reliable, secure and highly performant database to power that app. A free open source database may work brilliantly for testing and QA, but is community-supported software your best bet once it's time to take your mission-critical application live?

Four Essential Questions

Your answers to four key questions will help you determine when it's time for an enterprise open source database subscription:

Question 1: Do you have sufficient database expertise on staff?

Question 2: What is your uptime requirement and risk tolerance?

Question 3: How confident are you that your database is secure (from both internal and external threats)?

Question 4: Are you concerned about taking your database into the cloud?

Let's look at each of these in detail, and consider a few scenarios from the real world.

WHY WOULD I PAY FOR OPEN SOURCE?

It might seem counterintuitive, but devoting budget to an enterprise open-source solution can actually save you money. Some potential costs associated with community-supported software are mitigated (or avoided entirely) by moving to an enterprise solution. If you're considering hiring experts, if your current experts are spending significant time troubleshooting, or if missing features are undermining your business goals, the writing is on the wall. The potential for these issues to negatively impact your bottom line is reduced or eliminated by the reliability, stability and level of support that an enterprise-class solution can offer.

For instance, MariaDB database offerings come in two editions:

MariaDB Community Server: This is the free, community-supported version, licensed under the [GPL](#). As an open source software vendor, we value community participation and innovation, and we're proud to offer free software. However, for mission-critical production deployments many businesses require more high availability features, hardened security, uptime-specific SLAs and 24x7 DBA support. That's why we also offer MariaDB Enterprise Server.

MariaDB Enterprise Server: This is an enhanced, hardened and secure version of MariaDB Server, available only as a subscription. When it comes to mission-critical applications running in production environments, businesses value reliability, stability, security and long-term support.

- **Reliability:** MariaDB Community Server is released first. MariaDB Enterprise Server then undergoes an extensive QA process, and is made available to customers only when it meets our exacting standards. We back it up with world-class support from the same company that built the database.
- **Security:** All non-GA plugins are disabled by default in Enterprise Server to reduce the risks incurred when using unsupported features. Further, the default configuration enforces strong security, durability and consistency.
- **Stability:** MariaDB Enterprise Server releases include a number of backported features, allowing customers to benefit from critical improvements without having to incur the risk and uncertainty of a major upgrade every year.

We believe the best way to serve both the community and our customers is to build separate community and enterprise releases. With MariaDB Community Server, the community benefits from faster innovation, more experimentation and greater collaboration. With MariaDB Enterprise Server, customers benefit from greater reliability, safety and stability. You can compare the versions at a glance in the [side-by-side comparison chart](#) at the end of this guide.

Question 1: Do you have a team of database experts and analytics experts on staff?

Experts. We all need 'em. That means either hiring them yourself or paying for their time through a service provider. With free software, if questions arise, you're on your own. But with an Enterprise version, expert support is included to keep you operating efficiently. Consider these factors in the context of your on-staff expertise:

- **Databases are complicated.** No matter how you slice it, powerful databases are complex. *You don't know what you don't know...and you probably don't have time to figure it out.* The support included in an Enterprise subscription saves time and money by ensuring your database is operating as it should.
- **Community is great, but it takes time.** Community-supported software is a wonderful thing – what could be better than a group of enthusiasts donating time to build software that benefits everyone? MariaDB Community Server, for example, has strong community engagement through the MariaDB forums, knowledge base, and sources like StackOverflow. But with community platforms, if you don't have a dedicated database expert, you'll need to devote time to becoming involved in the community – not just to find answers, but to surface feature requests you may have. A feature you want may be an edge case that's important to your shop, but would require a large number of community requests for it before it's added to the development roadmap.

With MariaDB Enterprise Server, customers get the benefits of Community Server, plus a direct line to our database experts; a subscription comes with world-class support from the source, with 30-minute responses to S1 issues. Better still, support engineers will spend all the time necessary to get your problem resolved. Support also includes consultative services to answer questions before problems arise or offer advice on setting up your database in the best possible way for you. Additionally, feature requests from Enterprise customers are prioritized in the product roadmap.

- **Knowing which features best suit your needs is important.** Your database of choice may have an incredible list of features, but if you don't have a sufficient number of database experts on staff, how do you take advantage of all those bells and whistles? Should you be using data-in-motion encryption, or data-at-rest encryption? Would a Galera cluster benefit you? How about a distributed SQL architecture? With an Enterprise subscription you get access to the experts you need to make these decisions.
- **You don't need one expert; you need many experts.** Even if you do have database experts, if it isn't a huge team they're likely focused on optimizing for the queries your developers are writing, or creating specific schemas. They don't have time to figure out what storage engines are best for them, how to architect for high availability and DR and how to ensure your data is secure from costly breaches.

“Leveraging enterprise MariaDB and the Remote DBA support team not only created a high availability system for our content but also gave us access to experts to monitor and maintain our databases.”
—IT Bureau Chief at a US state unemployment agency

Question 2: What is your uptime requirement and risk tolerance?

Getting Things Running vs. Keeping Things Running

There's a big difference between performing an initial setup that demonstrates a service works in theory, and keeping that service running at scale, under stress, with rock-solid reliability. With an enterprise open source database subscription, there are protocols, policies and personnel available to ensure mission-critical reliability and minimal downtime.

- **Robust QA.** It's the life blood of reliable service. Community-driven development has systems for neutralizing bugs, but who determines those priorities? How many people are working on bugs, and how many hours are they spending on that? And will the community prioritize the bug that's crippling your app? Maybe, but an Enterprise subscription is the safe way to go if you're concerned. For instance, MariaDB Enterprise Server benefits from twice as much QA as Community Server – from a dedicated QA team. Community users generally receive updates (and the new features they unlock) quicker, but it also means they experience bugs more frequently. Enterprise Server protects you from this risk by including only features that have been thoroughly QA'd to ensure reliability and stability for your mission-critical apps. We back that protection up with responsive support should any hiccups find their way into your database.

“Thousands of the world's largest organizations depend on the Now Platform to create great experiences and unlock productivity. Better quality assurance and stability of critical enterprise features are extremely compelling. At our scale and in production with 100,000 MariaDB databases, reliability is what matters most.”

— Pat Casey, SVP of Development and Operations, ServiceNow

- **Your data deserves a second chance.** Automatic, multi-transaction failover. Sounds pretty great, right? In practice, it means that – if something goes wrong – your transaction data is preserved (without data loss) so transparently that your app won't even know there was a problem. Transaction replay, a feature built into MariaDB Enterprise Server, will even preserve transactions that fail mid-transaction.

The alternative? Data loss. Products without these sophisticated protocols (whether paid or unpaid) can, for example, fail over from one primary to another. In such scenarios, your app disconnects and throws a series of errors before eventually connecting to the new primary. The results? One broken app and a stretch of lost data. Fortunately, with an enterprise open source database you can easily avert such disasters thanks to features like automatic, multi-transaction failover and transaction replay.

Some Disasters Can't Be Avoided

There are times when dedicated experts, meticulous QA and robust feature sets simply don't matter. If a hurricane wipes out your data center, a phone call can't fix that! Negotiating these scenarios comes down to one thing: What's your backup strategy? And can a free database provide all the layers of redundancy you require, with staff to support it all? Here's a sampling of failsafes and efficiencies to think about:

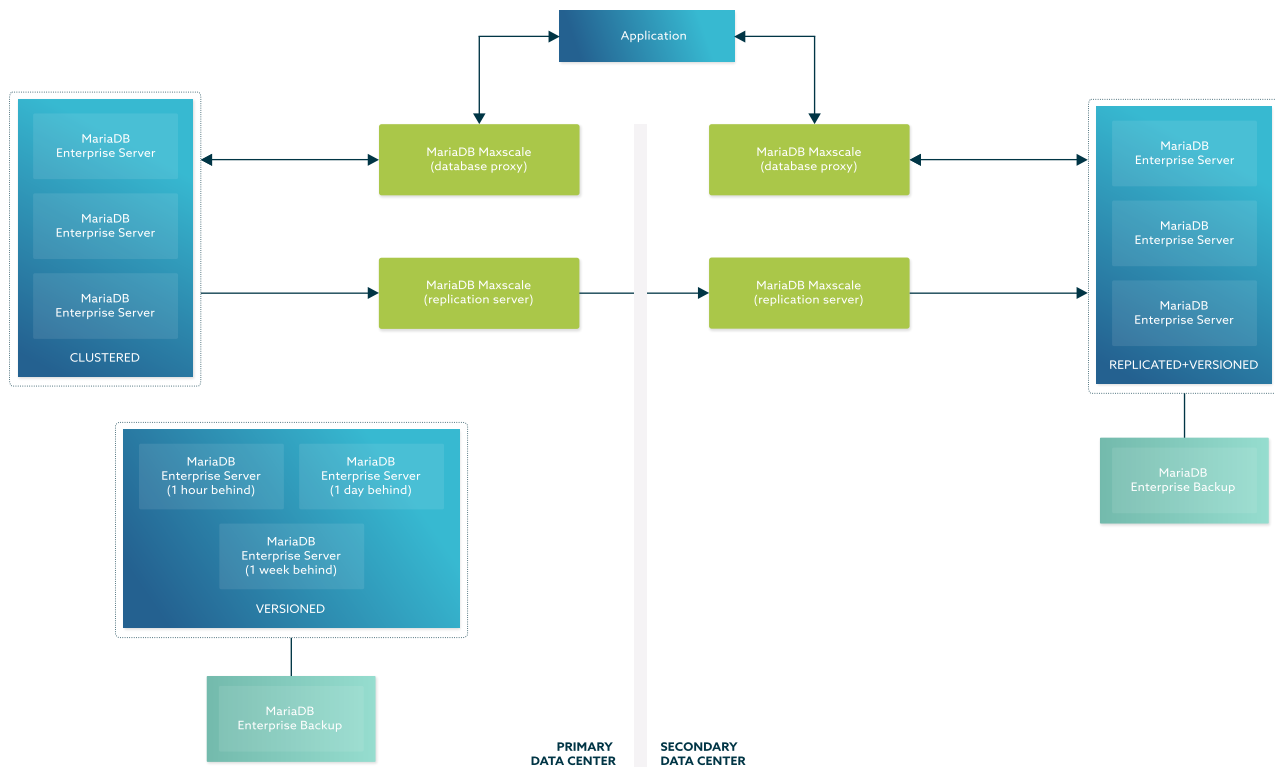
- **Non-blocking backups.** Historical backups usually come with a catch: You can't write data during the backup. Will your free database block writes, log tables, and statistics to prevent data from changing while a backup is being taken? Will it wait until all queries have completed before initiating a backup? These quirks can create significant problems with larger databases; what if there's an unexpected spike in traffic right before the backup is about to start – or worse, after it starts? Of course, blocking backups don't only block writes; they block *schema* changes, too. What if an urgent bug fix includes a schema change but a backup is still being taken? **Non-blocking backups** solve these problems by implementing a sequence of non-blocking backup stages that allows writes and schema changes *while tables are being backed up*.
- **Full backups.** Naturally, you'll want to start by creating a file-level backup of your data directory. This backup includes temporal data and the encrypted/unencrypted tablespaces of supported storage engines (e.g., InnoDB, MyRocks, Aria).
- **Incremental backup/restore.** Large data sets can make full backups both time-consuming and resource intensive. The solution? Incremental backups. When you perform an incremental backup, your backup service compares a previous full or incremental backup to what it finds on the database server. It then creates a new backup containing the incremental changes. Incremental restore applies the same resource-saving approach to data recovery.
- **Partial backup/restore.** All data is not created equal, so it makes sense to back up your different data sets at different frequencies. The partial backup/restore feature allows DBAs to back up subsets of tablespaces from the data directory. This allows you to focus your backup strategy on your most critical data without the resource burden of more-frequent full or incremental backups. Partial restore allows you to recover these critical data sets without taking the operational hit of a full or incremental restore.
- **Point-in-time recoveries.** Recovering from a backup restores the data directory at a specific point in time, but it doesn't restore the binary log. That's less than ideal. With point-in-time recovery, you can restore the binary log data to a specific point in time after restoring the data directory from a full or incremental backup.
- **Flashback + delayed replication.** Mistakes happen. If data has been wrongly deleted or modified recently, flashback-capable databases can roll back your latest transactions to restore the database to a point in time right before the data was deleted or modified. Think of it as an Undo button for your data. Of course, while this is going on you don't want that bad data replicating across your platform redundancies. This is where delayed replication comes in; it prevents recent, unwanted changes from propagating to one or more replicas, allowing one of them to be temporarily promoted to primary while the changes are rolled back on the previous primary.

“MariaDB’s Remote DBA is absolutely amazing. It’s an expert resource on tap – easy to contact, helpful and very responsive to where we are using it. Every dollar we invest translates into great value in helping us to be more agile as we strive to meet our business objectives.”
 – Steve Sharpe, System Architect, Teleplan

Understanding how to set all this up, maintaining it and paying for the associated infrastructure is a serious commitment. An enterprise open source database with robust consultative support and complementary remote DBA services make it exponentially safer and easier.

A Common Enterprise Database Architecture for Disaster Recovery

When it comes to disaster recovery, the ideal reference architecture depends on the recovery point objective and recovery time objective – as well as business and application requirements regarding high availability, performance and consistency. The reference architecture below, implemented by many MariaDB Enterprise subscribers, uses clustering in the primary data center for high availability and asynchronous replication to the secondary for performance. In addition, while the primary data center uses separate databases with delayed replication for backups and system versioning, the secondary uses a scaled-down deployment with a primary/replica topology and system versioning.



Question 3: How confident are you that your data is secure?

The need to protect your data is self-evident: The average total cost of a data breach is \$3.86 million,¹ and can have a far-reaching impact on trust in your brand, as well as your regulatory compliance standing.

Sophisticated features are required to protect valuable (and sensitive) data, but they typically aren't included with community-supported platforms. Let's review the above-and-beyond security features typical of (and professionally supported by) Enterprise-class platforms:

- **Secure by default.** Your database's default configuration should maximize security and durability protocols to ensure your production deployments are safe. With Enterprise Server, for example, security features are enabled automatically and all non-GA plugins are removed to limit exposure.
- **Database firewall.** A database firewall filter restricts data access and blocks SQL injection attacks by rejecting or accepting queries based on a combination of syntax, type, time and user. It should be configurable out-of-the-box and/or with custom rules, and must support both whitelisting and blacklisting queries. Additionally, database firewalls log rejected queries to provide DBAs with greater insight into suspicious activity.
- **Dynamic data masking.** Pseudo-anonymization via dynamic data masking is critical to preventing data breaches and meeting security standards and regulations such as the Payment Card Industry Data Security Standard (PCI DSS) and the General Data Protection Regulation (GDPR).
- **Data-at-rest encryption.** Encrypting tables makes it almost impossible for someone to access the original data. But this measure typically brings a 3-5% overhead demand, so you'll want configuration flexibility. MariaDB Enterprise Server, for example, provides several options: everything, including all tablespaces (with all tables); everything, excluding individual tables; and individual tables or log files.
- **Query result limiting.** A result-limiting filter is essential for preventing accidental or malicious queries from rendering a database unavailable – or exposing large amounts of data.
- **Enterprise Audit.** Compliance auditing conjures dread in most IT organizations. Having built-in tools that make proving compliance simple removes much of the burden. But the specifics of your auditing process matter. MariaDB Community Server, for example, includes basic auditing, but its procedures are insufficient to meet strict security standards like GDPR. Partly, this is because it offers inadequate visibility; for example, audit logs do not log changes to the auditing configuration itself, so nobody would know if the audit configuration was changed to log fewer events (or none at all). By contrast, MariaDB Enterprise Audit (which does satisfy GDPR requirements) logs all changes to its configuration and includes improvements like the ability to specify both user and host, and define rules per user.

¹ IBM. Cost of Data Breach Report 2020. <https://www.ibm.com/security/data-breach>.

Question 4: Are you concerned about taking your database into the cloud?

Gartner estimates that 75% of databases will be deployed or migrated to a cloud platform by 2022.² But many businesses simply don't know where to start.

For this reason, database-as-a-server (DBaaS) is among the fastest-growing cloud segments. Using a DBaaS makes it simple to deploy your database to the cloud, and the DBaaS vendor maintains, updates and supports your database as needed. If you're simultaneously considering an enterprise open source database subscription and thinking about moving your database to the cloud, choosing a DBaaS vendor that has extensive expertise in open source database architecture and administration for the enterprise lets you kill two birds with one stone. The right enterprise open source DBaaS can vastly simplify your tech stack, even removing the need to use separate database and data warehouse services (e.g., Amazon RDS + Amazon Redshift/Snowflake). Let's explore some of the potential perks:

"MariaDB has been a true collaborative partner for us in our journey to the cloud. With SkySQL, we don't have to bother with containers or managing the database. That's left to the database professionals at MariaDB. We also have the option of easily expanding our applications to leverage blended transactions and analytics when the time is right."

— Bryan Bancroft, Lead Database Administrator, FNI

- **Fast deployment and expert management.** Getting started is often the hardest part, so any DBaaS worth considering must radically simplify the process of moving your database into the cloud; it should take no more than a few clicks to deploy a production database in the cloud of your choice. Importing data and tweaking day-to-day operations should be similarly breezy, since ease of use is largely what you're paying for with a DBaaS. MariaDB's SkySQL, for example, reduces most admin processes to a series of straightforward GUI clicks via its SkySQL Portal while providing a broad array of options to ensure you get a database tailored to your specific needs. And SkySQL subscriptions include Enterprise Server.
- **Cloud and database expertise.** You now need cloud experts in addition to database experts. So make sure your DBaaS provider gives you the people you need at the pace you need them. SkySQL provides a solid benchmark: 24x7 support, deep troubleshooting and analysis, proactive monitoring and incident response, security audits, performance tuning, query and app optimization and more. Don't settle for less.
- **Improved security.** It may seem counterintuitive, but moving your database into the cloud can actually improve your security outlook. With a good DBaaS, admins never have to lift a finger (or even leave their home offices) to benefit from security features such as isolation, IP firewalling, data-at-rest encryption and data-in-transit encryption. And the best DBaaS providers go above and beyond those standard security measures to eliminate potential vulnerabilities, minimize your attack surface and enforce strict access protocols.

² Gartner. "Gartner Says the Future of the Database Market Is the Cloud." July 1, 2019. www.gartner.com/en/newsroom/press-releases/2019-07-01-gartner-says-the-future-of-the-database-market-is-the.

- **Bigger is better, but only when you need it.** Your DBaaS should allow you to scale both as your business grows and as-needed to meet spikes in demand (if, for example, your business is seasonal). By subscribing to a full-featured DBaaS, you forgo the capital expenditure of buying hardware, the delay of waiting for new systems to ship every time you need to scale up or scale out, and the overhead and opportunity cost of tuning, monitoring, and upgrading your database.
- **Cloud agnostic.** Cloud lock-in is a common (and understandable) fear for organizations moving to the cloud. The last thing you want is for the success of your business to be tied inextricably to a specific cloud vendor. But some DBaaS providers obligate you to deploy your database(s) to prescribed cloud services. In a world moving to multi- and hybrid cloud, that's just not necessary. And avoiding lock-in, in addition to keeping you in control, places you in a much better position when negotiating contracts with your preferred cloud provider.

“The team at MariaDB blew us away with their deep expertise, and we quickly realized the importance of a DBaaS from the source – no one knows how to run MariaDB better than MariaDB. Being on the cloud and moving to SkySQL gives us the ability to scale quickly to meet our growing business requirements.”
— Sean Clark, Co-founder and CTO, Whitebox

SO, IS IT TIME TO PAY FOR YOUR OPEN SOURCE DATABASE?

This is a question only you can answer. While there are many reasons to consider an Enterprise subscription, those who commit tend to have four goals foremost in mind:

- **Leave databases to the database experts.** If you prefer to focus on product innovation rather than data tables, you need an enterprise-grade open source database.
- **Never worry about losing data or business due to technical glitches.** Exhaustive QA, failovers, ACID compliance, etc. – without an extraordinarily large DBA team, these key features in a mission-critical database require the expert support of an enterprise subscription.
- **Peace of mind for data security.** Security isn't just important; it's mandatory. But hackers are endlessly clever. And your IT team's time is likely better spent on pursuits outside this game of cat and mouse. You need a database that's secure by default – which means an enterprise subscription.
- **A database that meets the ever-evolving demands of the cloud.** As the footprint and feature demands of your cloud implementation grow, you need a database that keeps pace. Your database needs to go wherever your apps take it, it needs to go there fast, and – ideally – somebody else should get it there for you!

If these goals resonate with you, it may be time to consider an enterprise open source database like MariaDB Enterprise. For additional info on the benefits of taking the plunge, get an overview from the table below, then [learn more about the value of a MariaDB Enterprise subscription](#).

FEATURE COMPARISON: MARIADB COMMUNITY (FREE) VS. ENTERPRISE (SUBSCRIPTION)

		Community	Enterprise
High availability	Replication and Clustering	✓	✓
	Automatic Failover		✓
	Transaction replay		✓
Performance	Thread Pool	✓	✓
	Query result caching		✓
Scalability	Table Sharding	✓	✓
	Schema Sharding		✓
	Read/write splitting		✓
	Casual reads		✓
Security	Transparent data encryption (TDE)	✓	✓
	Dynamic data masking		✓
	Database firewall		✓
	Query result limiting		✓
Analytics	Column storage	✓	✓
	Apache Spark connector		✓
	Pentaho Kettle plug-in		✓
Integration	Change-data-capture client	✓	✓
	Apache Kafka connector		✓
Tools	MariaDB Backup	✓	✓
	SQLyog		✓
	SQL Diagnostic Manager		✓
Services	Knowledge base	✓	✓
	Technical support		✓
	Consultative support		✓
	Product notifications		✓
	Security alerts		✓
	Limited indemnification		✓