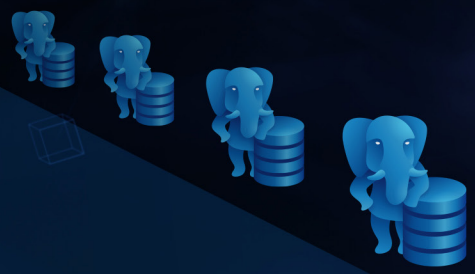
PERCONA

Achieving High Availability on PostgreSQL With Open Source Tools

A guide for businesses of all sizes – startup, medium, large, and globally dispersed enterprises



Why read this eBook?

Remember when you could schedule database downtime for hours to perform maintenance? Or when an unplanned outage was more of a nuisance and less of an immediate threat to your business's viability? Those were the good ol' days.

Now, **high availability (HA)** is table stakes. If your business's services are down, the consequences are significant. Sources vary, and data depends on a range of variables (industry, seasonality, size, etc.), but most reports estimate the average cost of unplanned downtime to fall somewhere in the ballpark of **$300,000 - $500,000 per hour**.

Of course, you know as much already. Chances are you've cobbled together an assortment of extensions and tools from hundreds of different sources in an effort to meet your HA goals. That's what most DBAs do when using PostgreSQL, as the community version does not come out of the box with the features and functionality necessary to achieve high availability.

But how confident are you in this piecemeal approach? Is your mishmashed database environment overly complex and all too brittle? Do you worry that a failure of your chosen components to work together could undermine your goals? Are you struggling with clustering and reliability, or are you forced to create workarounds for failover and switchover?

If so, this eBook is for you.

In it, we'll start with the basics of high availability. Then, we'll share Percona HA PostgreSQL reference architectures for businesses of every size (startup/small, medium/large, enterprise), which pull together tools and components to enhance and harden interoperability and ensure your confidence in high availability on PostgreSQL.

Each of these proposed architectures, which are built on years of Percona expertise and feature streaming replication, can offer an HA level of 99.999%, or five nines.

High availability basics

What is high availability?

At its simplest, high availability is created by eliminating Single Points of Failure (SPOF), making sure there is sufficient redundancy in all components, and ensuring there is a foolproof method of making those redundancies work automatically. Let's unpack that quickly.

SPOFs are defined as any component of your database environment that, if they were to fail, would result in downtime. The majority of SPOFs are a consequence of scale, where designs that were once adequate (excellent, even) no longer meet evolving, more complex requirements. Downtime due to SPOFs can also be attributed to bottlenecks from architectures designed for applications instead of databases. A non-purpose-built database environment, like the one described in the introduction, will result in a high number of SPOFs.

The second consideration is redundancy. Redundancy refers to when duplicate or mirrored data is stored in more than one place, such as a server. To qualify for high availability, redundant systems must be supported by failure detectability mechanisms and, in the event of an issue, failover (more on that later) must be handled automatically and seamlessly.

Addressing the aforementioned issues to achieve high availability takes place across four key areas:

Infrastructure

This is the physical or virtual hardware the database systems rely on to run, in this case, PostgreSQL. Without enough infrastructure (physical or virtualized servers, networking, etc.), there cannot be high availability.

Connection management

This is the software management related specifically to the networking and connectivity aspect of the database. Clustering solutions typically bundle with a connection manager. However, in asynchronous clusters, deploying a connection manager is mandatory for high availability.

Topology management

This is the software management related specifically to the database and managing its ability to stay consistent in the event of a failure.

Backup and continuous archiving

PostgreSQL asynchronous replication relies on the write-ahead log (WAL) stream, and it's vital to have centralized storage that all nodes can access. This is the location where the primary node will archive the WAL segments and the replicas will restore any segment that isn't present on the primary anymore. It is of extreme importance to keep the replication working if any replication delay happens and the replica node isn't able to work at the primary's pace. Those files can also be used for Point-in-time Recovery if any disaster occurs.

Measuring high availability

Generally speaking, the measurement of availability is done by establishing a measurement time frame and dividing it by the time that it was available. This ratio will rarely be one, which is equal to 100% availability. At Percona, we don't consider a solution to be highly available if it is not at least 99% or two nines available.

Availability %	Downtime per year	Downtime per month	Downtime per week	Downtime per day
99% *"two nines"*	3.65 days	7.31 hours	1.68 hours	14.40 minutes
99.5% *"two nines five"*	1.83 days	3.65 hours	50.40 minutes	7.20 minutes
99.9% *"three nines"*	8.77 hours	43.83 minutes	10.08 minutes	1.44 minutes
99.95% *"three nines five"*	4.38 hours	21.92 minutes	5.04 minutes	43.20 seconds
99.99% *"four nines"*	52.60 minutes	4.38 minutes	1.01 minutes	8.64 seconds
99.995% *"four nines five"*	26.30 minutes	2.19 minutes	30.24 seconds	4.32 seconds
99.999% *"five nines"*	5.26 minutes	26.30 seconds	6.05 seconds	864.00 milliseconds

Keep in mind, there is no solution that can guarantee 100% availability. A disaster can still happen, and the application should be able to handle some level of instability and/or service downtime.

Your high availability needs will vary and depend on numerous factors, including:

- **criticality of the applications**
- **customers and users impacted**
- **how quickly a database of application must failover**
- **acceptable limits on data loss**

For mission-critical applications, such as those used in e-commerce, four nines (99.99%) is generally considered the industry standard. As the above chart shows, four nines equates to 52.60 minutes of total unplanned and planned downtime per year. For non-critical applications and systems, two nines is the widely accepted benchmark, which equates to 3.65 days of total downtime per year.

The architectures proposed in this document are categorized as five nines, which equates to 5.26 total minutes of downtime per year.

High availability and failovers

Failovers are the primary way in which high availability systems are maintained. If properly planned and architected, a database node failure shouldn't affect the availability of the database infrastructure. **Aligning the business requirements for availability and uptime with failover methodologies is critical to achieving those goals.**

Below are the two main types of failovers that can occur in database environments.

Switchover

A switchover is a failover that has been scheduled in advance or occurs at a regular interval. There can be many reasons for planned failovers, including patching, large data operations, retiring existing infrastructure, or simply testing the failover strategy.

Unplanned failover

An unplanned failover occurs when a database unexpectedly becomes unresponsive or experiences instability. Unplanned failovers could also include emergency changes that do not fall under the planned failover cadence or scheduling parameters. Unplanned failovers are generally considered high risk operations due to the elevated potential for data loss, data corruption, or data fragmentation.

High availability and maintenance windows

Major vs. minor maintenance

Although it may not be obvious at first, not all maintenance activities are created equal, nor do they have the same dependencies. It is good to separate maintenance that demands downtime or switchover from maintenance that can be done without bringing down a node. When defining these maintenance dependencies, there can be a change in the actual maintenance process that allows for a different cadence.

Maintenance without service interruption

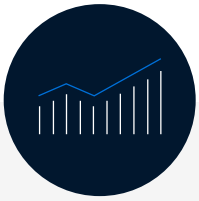
Using a rolling upgrade, it is possible to cover both major and minor maintenance without service interruption. The process is two-phased. In the first phase, a standby node is taken down and upgraded while the other standbys and the primary remain online. This process is repeated, one by one, until all standbys are upgraded.

In the second phase, the primary is upgraded. Here, the application is disconnected from the primary and connected to a standby. The standby is promoted to the new primary while the former primary is upgraded.

Measuring and monitoring your database infrastructure

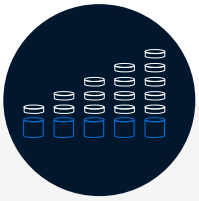
To ensure database infrastructure is performing as intended or at its best, it is necessary to measure specific metrics and alert when some of these metrics are not in line with expectations. A periodic review of these measurements also helps to promote stability and understand potential risks associated with the database infrastructure.

Below are the three aspects of database performance measurement and monitoring.



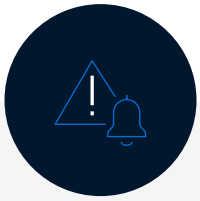
Measurement

To understand how a database infrastructure is performing, there is a need to measure multiple aspects of the infrastructure. With measurement, it's important to understand the impact of the sample sizes, sample timing, and sample types.



Metrics

Metrics refers to the actual parts of the database infrastructure being measured. When discussing metrics, more isn't always better, as too many can introduce unintentional noise or make troubleshooting overly burdensome.



Alerting

When one or many metrics of the database infrastructure is not within a normal or acceptable range, an alert should be generated so that the team responsible for the affected portion of the database infrastructure can investigate and remedy it.

Achieving high availability on PostgreSQL with open source tools

Multiple tools and architectures are available for managing the high availability of PostgreSQL. Each has its own pros and cons, but this eBook will describe only one approach, which will use continuous streaming replication as the replication method.

Let's break down what that entails.

Continuous archiving

PostgreSQL maintains a WAL that records every change made to the database's data files. While this log exists primarily for crash-safety purposes, it can also be used for replication. By packing up the primary server, creating another server with this backup, and shipping the generated WAL files to this new server (continuously archiving the logs), you can create a warm standby system. This replica will have a nearly current copy of the database and can be used for HA.

Streaming replication

Streaming replication is a technique where the replica connects to its primary and continuously receives a stream of WAL records as they're generated. This technique allows the standby to be up to date.

Synchronous streaming replication

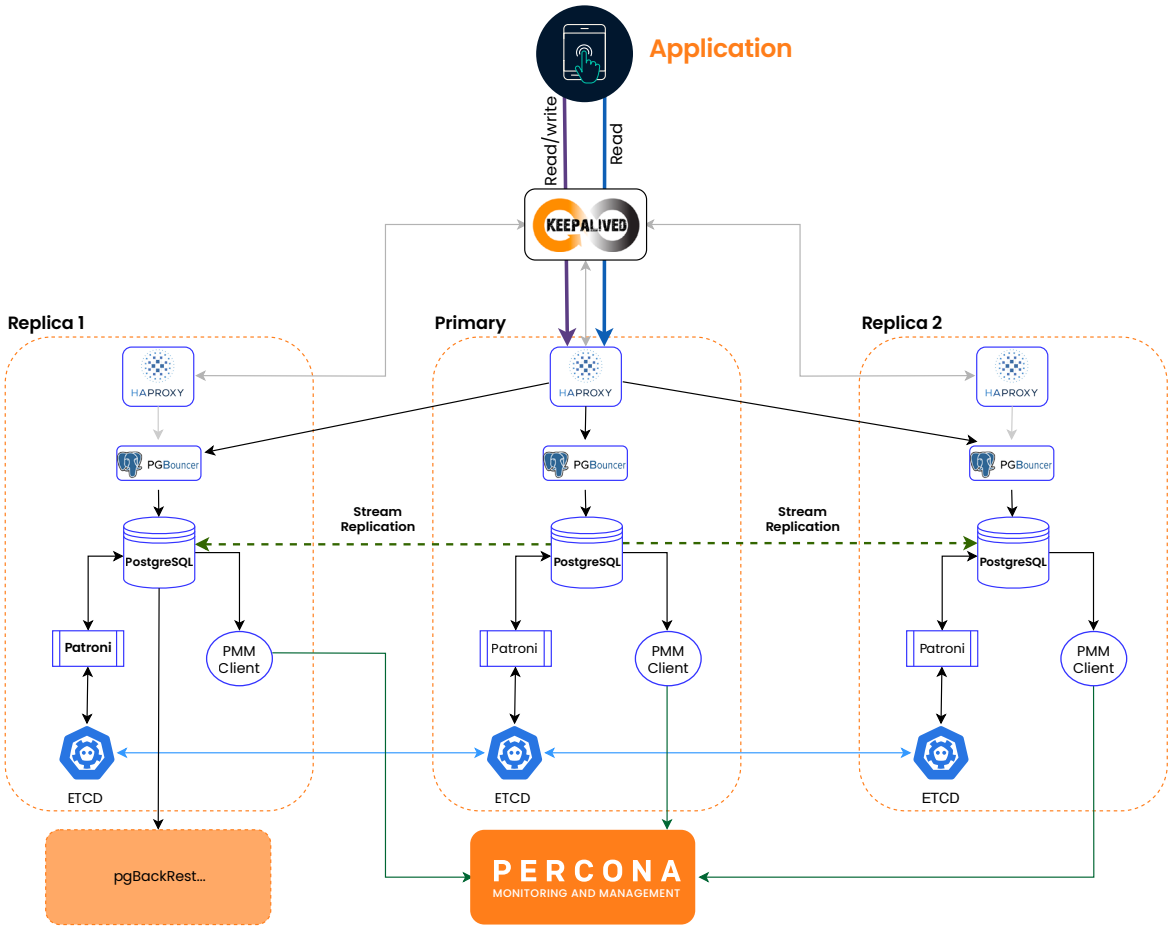
PostgreSQL streaming replication is asynchronous by default but can be configured to work synchronously by electing one or more replicas to be a synchronous standby. With synchronous replication enabled, you can extend the standard level of durability offered by a transaction commit, with the ability to confirm that all changes made by a transaction have been transferred to one or more synchronous standby servers.

The above approach underpins all three of the proposed architectures you'll find below. Each is tailored according to business size, and every one **offers an HA level of 99.999%, or five nines**.

High availability PostgreSQL reference architecture for startups and small businesses

There is an unfortunate misconception that high availability solutions are too expensive for startups and small businesses. Maybe you're of that belief. Perhaps you've been turned off by the necessity of managing more system resources. Or maybe you've run the numbers and believe that the cost of implementing a high availability architecture, often via expensive tools, outweighs the loss of revenue that could come from downtime.

But in using open source tools, as outlined in the below Percona architecture, high availability on PostgreSQL is achieved without an exuberant price tag or the need for an overly complex environment. In fact, it can be done by building the HA infrastructure within a single data center on your local node. This base reference architecture intends to keep the database available for your application in case the primary node goes down — whether it is Automatic Failover in case of a disaster or Planned Switchover in case of a maintenance window.



The architecture above is powered by streaming replication and built solely with open source tools:

- ETCD and Patroni as an HA controller
- KeepAlive as Connection Router
- Percona Monitoring and Management (PMM) for performance monitoring and database management
- pgBackRest for Backup and Continuous WAL Archiving

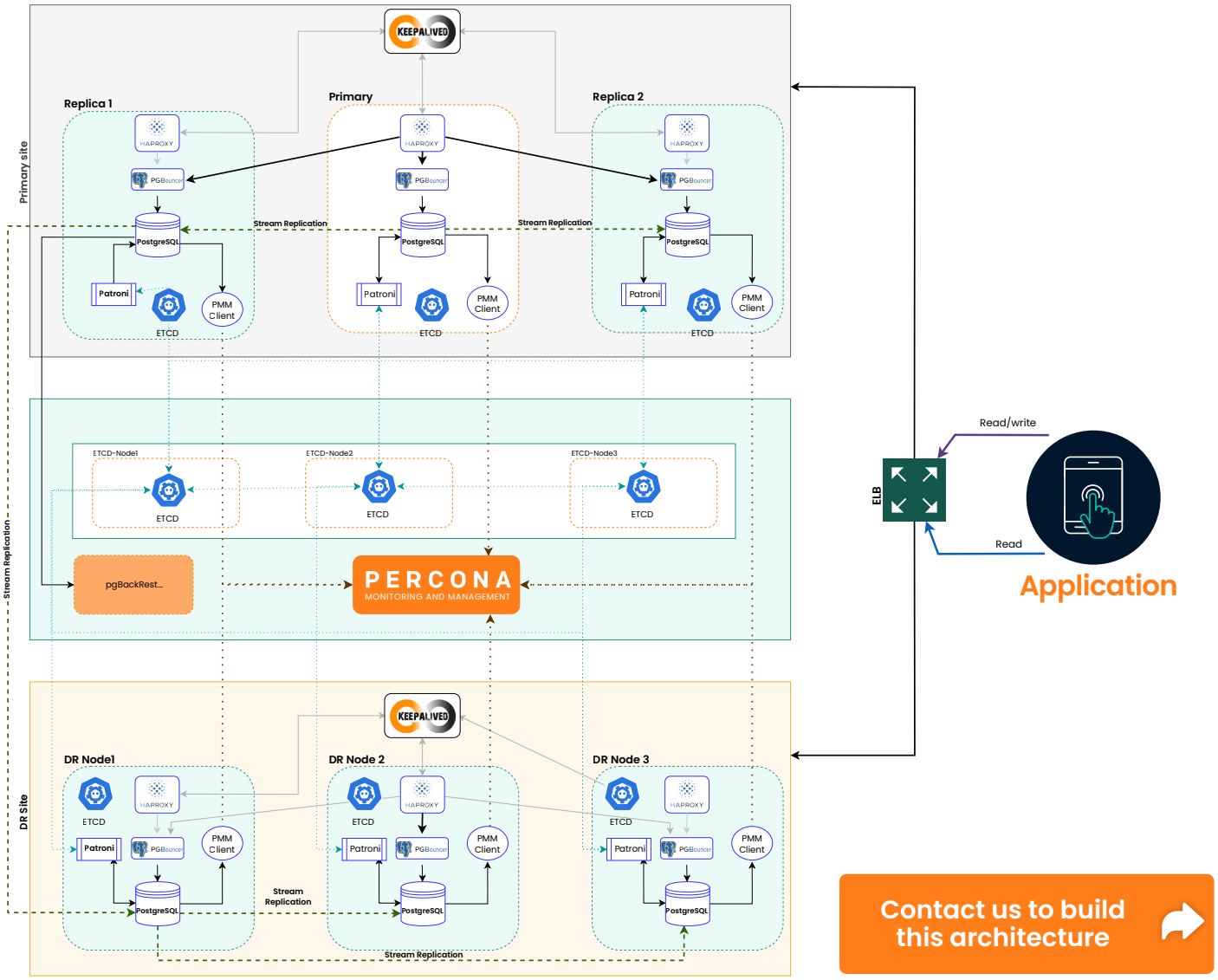
Build this architecture using our docs

High availability PostgreSQL reference architecture for medium and large businesses

As your business grows, so should your high availability architecture. At this size, the consequences of downtime, both in terms of lost revenue and erosion of customer sentiment, are significant. High availability requires more fault-tolerant, redundant systems and likely larger investments in IT staff. Still, in using open source tools, high availability can be achieved cost-effectively and without the threat of vendor lock-in that can come from paid enterprise SQLs.

In this architecture, **Percona spreads availability across data centers** to add more layers of availability to the cluster. This version of the architecture is designed to keep your infrastructure available and your data safe and consistent even in the event of a problem in the primary data center.

It features a disaster recovery site and an external layer of ETCD nodes. If the communication between the sites is lost, the ETCD node layer will act as a "source of truth" and decide which replica will be promoted as a primary, keeping the cluster healthy without creating a split-brain and the infrastructure highly available.

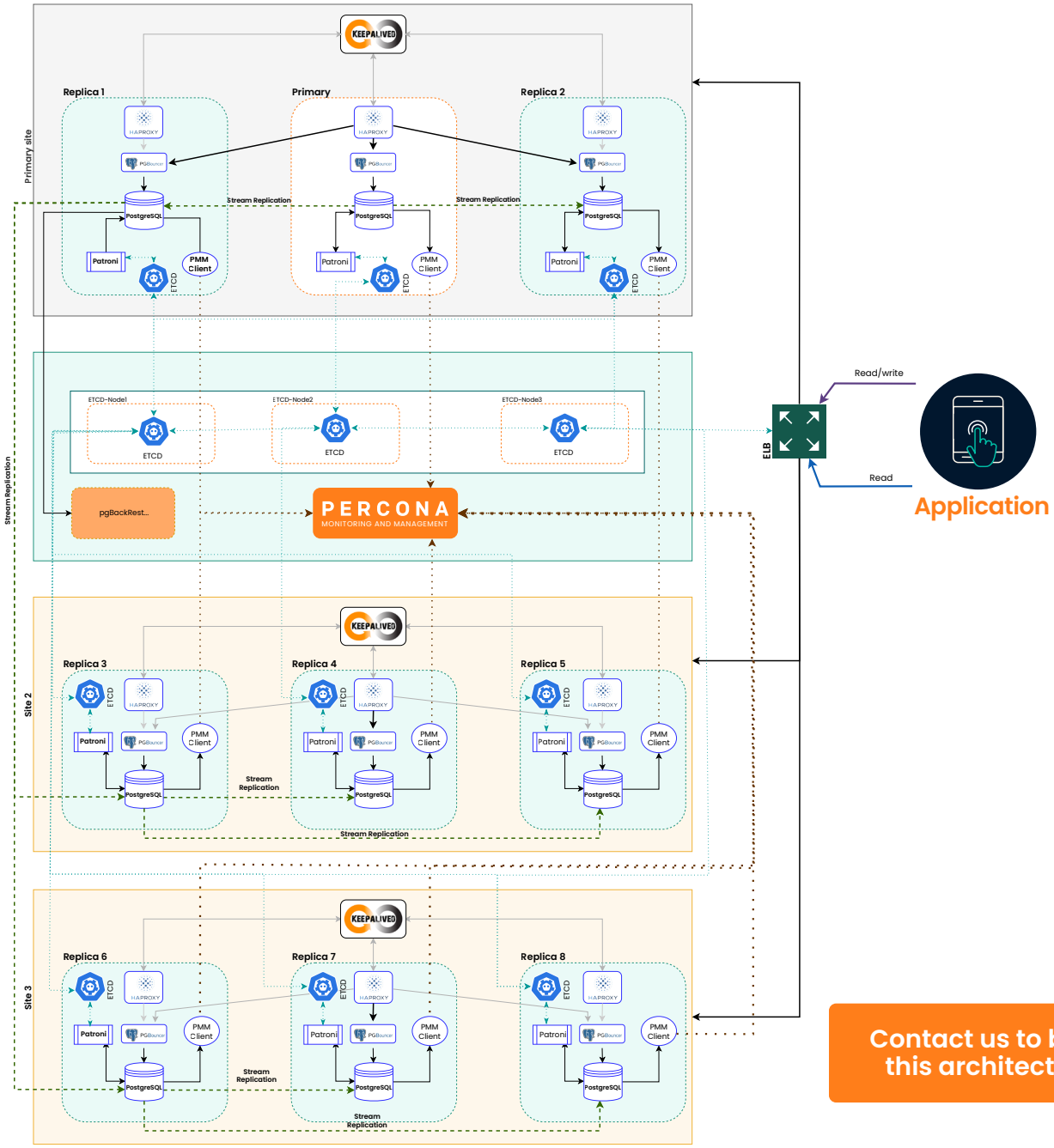


Contact us to build this architecture

High availability PostgreSQL reference architecture for globally distributed enterprises

The challenges and consequences mentioned in the medium and large business section are multiplied for enterprises. The proposed architecture reflects that, as it is an evolution of the architecture proposed above.

This architecture features two disaster recovery sites and adds more layers to the infrastructure in order to stay highly available and keep the applications up and running. This architecture, based on tightly coupled database clusters spread across data centers and availability zones, can offer an HA level of 99.999% when using synchronous streaming replication, the same hardware configuration in all nodes, and fast internode connection.



Contact us to build this architecture

Achieving high availability with Percona on PostgreSQL

PostgreSQL is one of the most used and trusted open source databases, and ensuring its uptime should be a major priority for businesses of any size. The architectures discussed in this eBook enable precisely that. With them, you can eliminate the complexity and brittleness of your mishmashed database environment and achieve an HA level of five nines, which equates to 5.26 total minutes of downtime per year.

Learn More

030223

About Percona

Databases run better with Percona. Percona is the only company that delivers enterprise-class products, support, and services for a range of open source databases, including MySQL®, MariaDB®, MongoDB®, and PostgreSQL across traditional deployments, cloud-based platforms and hybrid IT environments. The company is committed to supporting open source as an approach to software licensing, development, and deployment - its database management tools are used by millions of application developers, database administrators, and IT professionals worldwide.

Percona equips businesses with the freedom to choose, the freedom to create, and the freedom to make a difference — helping them scale with speed as they grow. The company supports global brands such as PayPal, Vimeo, RockStar Games, Duolingo, Fiserv, Slack, Cisco Systems, and Rent the Runway, as well as smaller enterprises looking to maximise application performance while streamlining database efficiencies.

For more information, visit percona.com.

For more information please contact us at:
+1-888-316-9775 (USA), **+44 203 608 6727** (Europe),
or via email at sales@percona.com