WHITEPAPER

Highly Available PostgreSQL with Physical (Streaming) Replication

1

Introduction

This white paper discusses the scenarios when designing a High Availability architecture for PostgreSQL. It looks at the desirable features and benefits of delivering High Availability in a business environment. Finally, it proposes an architecture that delivers on the above.

2

High Availability Database Clusters

Business applications require high reliability of their backend database cluster. Availability is critical for any business, whether for a planned switchover, or an unplanned failure. An architecture that can handle either situation is essential to any organization running their business on PostgreSQL. For situations that you can plan for, administrators require the appropriate controls to schedule and monitor changes in the cluster. In the event of an unplanned failure - in either a primary or standby node - a High Availability architecture must ensure the database remains available for your business applications. Finally, in the case of a complete data center failure, it must provide architectural redundancy at a secondary location.

3

High Availability Single Master Architecture

One way to deliver high availability is to utilize a single master with multiple standbys. PostgreSQL physical replication technology is used to copy data from WAL files on disk to the replicas.

The reference archictecture in figure 1 utilizes three open source PostgreSQL extensions: PgBouncer, repmgr, and Barman. The key characteristics of this architecture are as follows:

- It is deployed across two data centers (A & B)
- Each data center is an **availability zone**
- The following components are included:

Four PostgreSQL nodes (nodes 1-4)

  - One node designated as primary
  - Three nodes designated as Standby

Two PgBouncer nodes
(providing connection pooling)

Two Barman nodes
(enabling both backup and recovery of PostgreSQL nodes)

  - The node in the same data center as the Primary PostgreSQL node is configured as Primary
  - The one in the other data center is configured as Passive

Each PostgreSQL node runs repmgr
(assisting administrators with cluster management)

- Applications access the database layer through PgBouncer
- PostgreSQL Standby nodes are 2, 3, 4 are cloned from Node 1 using repmgr, and can replace Node 1 on failure
- The Primary Barman node takes backups from the current Primary PostgreSQL node
- WAL is transmitted using **Physical Streaming Replication(PSR):**
  - from primary PostgreSQL node to Standby PostgreSQL nodes 2, 3
  - from primary PostgreSQL node to the Primary Barman node
  - from Node 3 to Node 4, both located on the same date center
- WAL streaming is chosen because it provides a lower Recovery Point Objective (RPO)
- The Primary Barman node uses a **replication shot** to ensure that the upstream node does not remove unsent WAL

- The Passive Barman node fetches WAL and backups form the Primary Barman node using **Geo-redundancy** mode
- Standby PostgreSQL nodes do not use replication slots; if they fall behind due to WAL streaming being disrupted, they can restore any missing WAL directly from the local Barman node

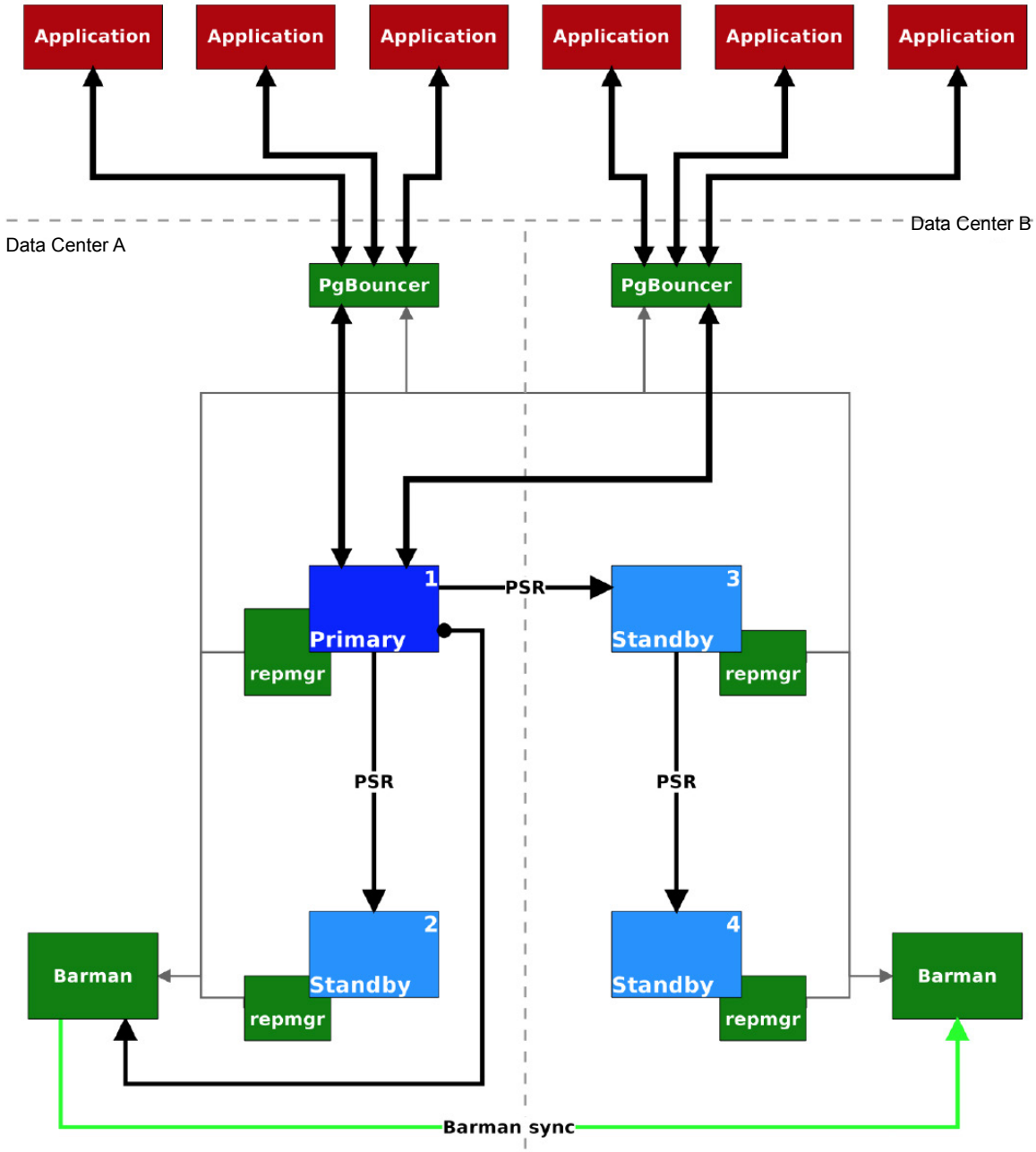


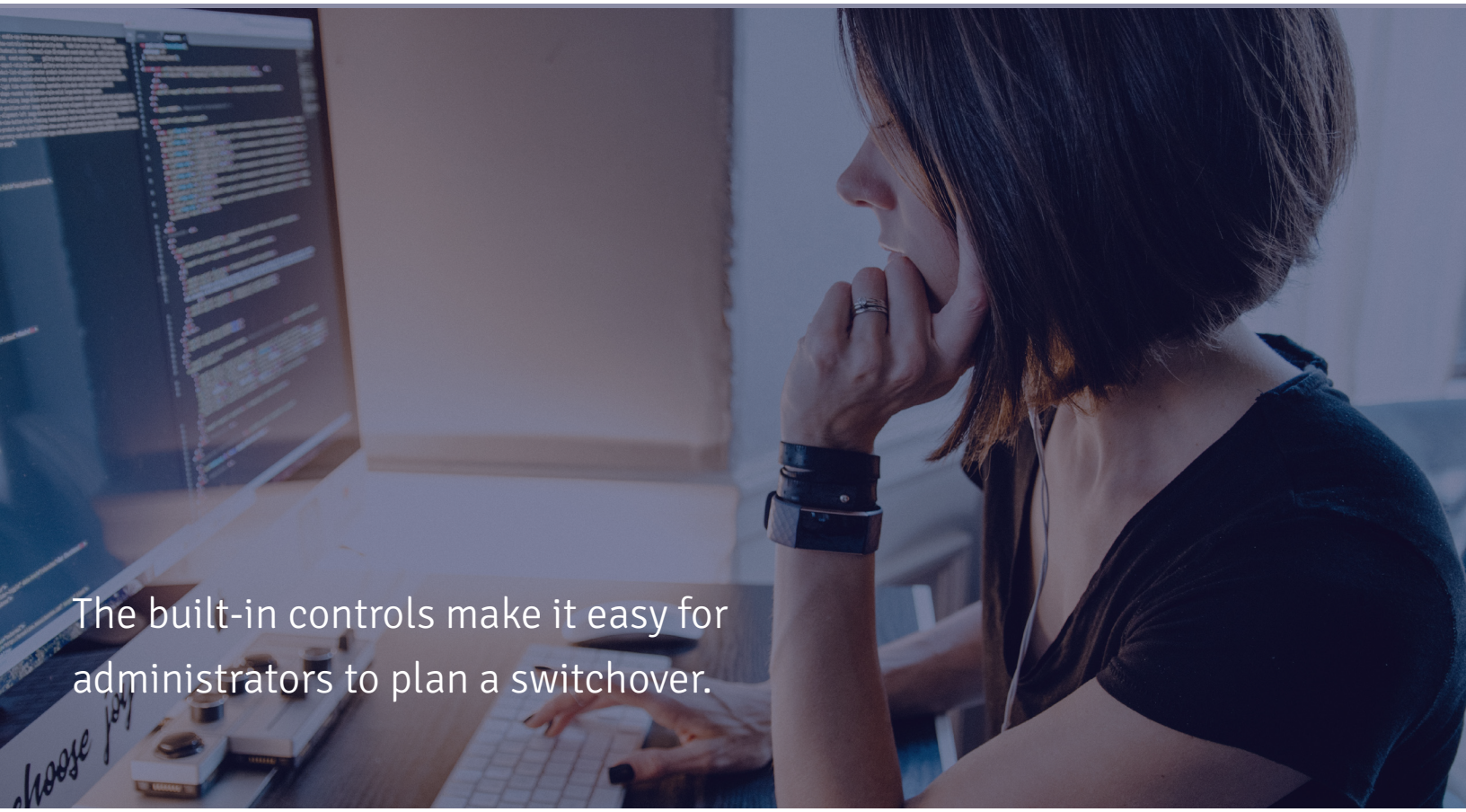
Figure 1. Highly Available Single Master Architecture

4

Delivering High Availability and More

4.1 High Availability

This reference architecture comes with the ability to handle either planned switchover, or an unplanned failure situation. The built-in controls make it easy for administrators to plan a switchover. For example, by pausing PgBouncer, shutting down Node 1, waiting until Node 2 is ready to be promoted, promoting Node 2, and finally reconfiguring PgBouncer and other nodes - including Barman - to work with the new primary.



In the event of an (unplanned) node failure, primary or standby, the cluster will failover to ensure the database remains available for your applications. For example, if the Primary PostgreSQL node becomes unresponsive or unreachable, its state must be monitored until it recovers, or until a predefined timeout. If the Primary node recovers, the failure is proven to be temporary and the cluster returns to normal state without requiring any action. Conversely, if the Primary node does not recover within the timeout, then it is considered permanently failed. On permanent failure, a failover procedure is initiated, and repmgr ensures that the failed Primary does not return to the cluster. The Failover procedure has the immediate goal of resuming write activities, by promoting the 'newest' standby to a primary node, and the subsequent goal of restoring the expected redundancy, by replacing the promoted standby node with a newly provisioned one.

> The Failover procedure has the immediate goal of resuming write activities, by promoting the 'newest' standby to a primary node, and the subsequent goal of restoring the expected redundancy, by replacing the promoted standby node with a newly provisioned one.

Finally, this architecture provides redundancy in case of complete data center failure. The failure of an entire data center can occur for several reasons, including power outage, planned maintenance, or simply the loss of connection between both data centers. The latter is interpreted by each data center as a failure of the other one. The configuration must also distinguish between the failure of a data center where the Primary PostgreSQL node is running (in our diagram 'Data Center A') and the case where the failed data center includes only Standby PostgreSQL nodes, (Data Center B in our figure).

In case of failure of 'Data Center B', the application can still continue working, as the Primary PostgreSQL node and Standby (node 2) are still functioning. Application connections will migrate gently: any application that was connected to the PgBouncer instance in 'Data Center B' will have been disconnected, and will then reconnect via the PgBouncer instance in 'Data Center A'. Once all application connections are migrated to 'Data Center A', a new data center 'Data Center B' can be created with the same components.

Conversely, if data center 'Data Center A' fails, then the only remaining option is failing over to data center 'Data Center B'. If the data center is fully lost, the application nodes running in 'Data Center A' will also be inoperable. This architecture supports a procedure that is able to restore operations after failure of the data center where the Primary PostgreSQL node is running. However, the resulting cluster does not have full redundancy capabilities, because it is vulnerable to a second data center loss. It is therefore important to restore this capability at the earliest opportunity by provisioning new nodes in another functioning data center. This procedure is very similar to the one described above in response to a failure of data center 'Data Center B'.

> Application connections will migrate gently: any application that was connected to the PgBouncer instance in 'Data Center B' will have been disconnected, and will then reconnect via the PgBouncer instance in 'Data Center A'

4.2 Backup Strategy and Disaster Recovery

The solution comes with the built-in ability to formulate a backup strategy and recover from disasters. Barman (leading tool for Backup & Recovery Management) is configured on two nodes. The Primary Barman node is configured to perform base backups and archive WAL files from Node 1, the Primary PostgreSQL node in 'DC A'. The base backup, together with the WAL archive, can be used to both restore a PostgreSQL node using Point-In-Time-Recovery (PITR) and provide Standby nodes with any older WAL files that have already been eliminated from their upstream nodes. The Passive Barman node does not connect to PostgreSQL. Instead, it fetches data from the primary Barman node to provide backup redundancy without additional I/O pressure on the database servers.

Conclusion

High Availability has two enemies, the planned and the unplanned incident. For the former, an architecture must present the features and controls that allow database administrators to seamlessly switchover from a Primary to a standby node without impacting the business that the database is supporting. Single Master with High Availability achieves this by a combination of: deploying multiple standby nodes; having redundancy in the connection pooling; utilizing Physical Streaming Replication of the WAL files; and supporting Geo-redundancy in its backup and recovery solution.

For unplanned node failure, the architecture supports a robust Failover procedure that has the immediate goal of resuming write activities (and hence not losing application data), by promoting the 'newest' standby to a primary node. Ultimately, to restore the expected levels of redundancy and High Availability, the procedure includes replacing the promoted standby node with a newly provisioned one.

The worst unplanned failure is a complete outage in a data center. This solution is deployed across two data centers and if failure happens to the secondary data center, then a smooth transition of application connections to the operational data center occurs, followed by a new 'standby' being provisioned. If it is the main data center that fails, then this solution supports a procedure that is able to restore operations in the secondary data center whilst achieving a low RPO.

About EDB

PostgreSQL is increasingly the database of choice for organizations looking to boost innovation and accelerate business. EDB's enterprise-class software extends PostgreSQL, helping our customers get the most out of it both on premises and in the cloud. And our 24x7 global support, professional services, and training help our customers control risk, manage costs, and scale efficiently.

With 16 offices worldwide, EDB serves over 4,000 customers, including leading financial services, government, media and communications, and information technology organizations. To learn about PostgreSQL for people, teams, and enterprises, visit EDBpostgres.com.

WHITEPAPER

Highly Available PostgreSQL with Physical (Streaming) Replication

EnterpriseDB Corporation
34 Crosby Drive
Suite 201
Bedford, MA 01730

EDB™

EDBPOSTGRES.COM