



# EDB Postgres Distributed Always On: **The End of the Reign of Oracle RAC**

## AUTHORED BY:

**Celest Turner Hall**  
SVP, Product Marketing

**Kelly Poole**  
VP, Product Management

**Petr Jelinek**  
VP, Chief Architect

# Contents

<b>Introduction</b>	<b>03</b>
<b>1. What does it mean to be Always On?</b>	<b>04</b>
1.1 Who needs this?	05
1.2 Customer case study: ClickUp	07
1.3 Why is this so hard?	07
<b>2. Overcoming PostgreSQL HA limitations with EDB Postgres Distributed</b>	<b>08</b>
<b>3. Overcoming PostgreSQL HA limitations with EDB Postgres Distributed</b>	<b>12</b>
<b>4. The business case for making the switch</b>	<b>14</b>
<b>5. Conclusion</b>	<b>16</b>

# Introduction

'Always On' is a hallmark of digital transformation. Digital businesses never sleep. Mobile apps, websites, IoT systems, social media, telcos, and financial services are operating 24 by 7. Hour-long maintenance windows are a thing of the past.

Whereas outages used to cost money because of missed transactions, today they cost customers. Credit cards that are rejected because the service is not available move to the back of the wallet; payment gateways that are offline for more than a few seconds get replaced by vendors with better availability; content that doesn't load instantaneously all the time is soon forgotten; security systems that cannot generate instant access keys are unacceptable.

Databases are foundational for all of these services. Their 'Always On' mojo, also known as High availability (HA), is constrained by what the database can support. If the database can only achieve 99.99% availability, then the whole solution cannot exceed that service level.

This whitepaper takes the reader through a discussion of what it means to be 'Always On', what the obstacles are, and how EDB Postgres Distributed creates a Postgres®-based alternative to proprietary solutions, such as Oracle® Real Application Cluster (RAC). We end with a discussion of the business case for EDB Postgres Distributed.

# 1. What does it mean to be Always On?



# What does it mean to be Always On?

Not too many years ago, High Availability referred to technologies that protected software systems from hardware and network failures. Clustering solutions, for example, would automatically fail over when the underlying server malfunctioned. That reflected a very software system-centric point of view. Today, HA refers to service availability—today's definition is independent of why a software or hardware component has failed. HA measurement takes planned and unplanned maintenance activities into account, not just hardware and network failures. That means that HA solutions for the database need to be able to guarantee database service availability including the unavoidable maintenance windows for hardware, OS, and the database software.

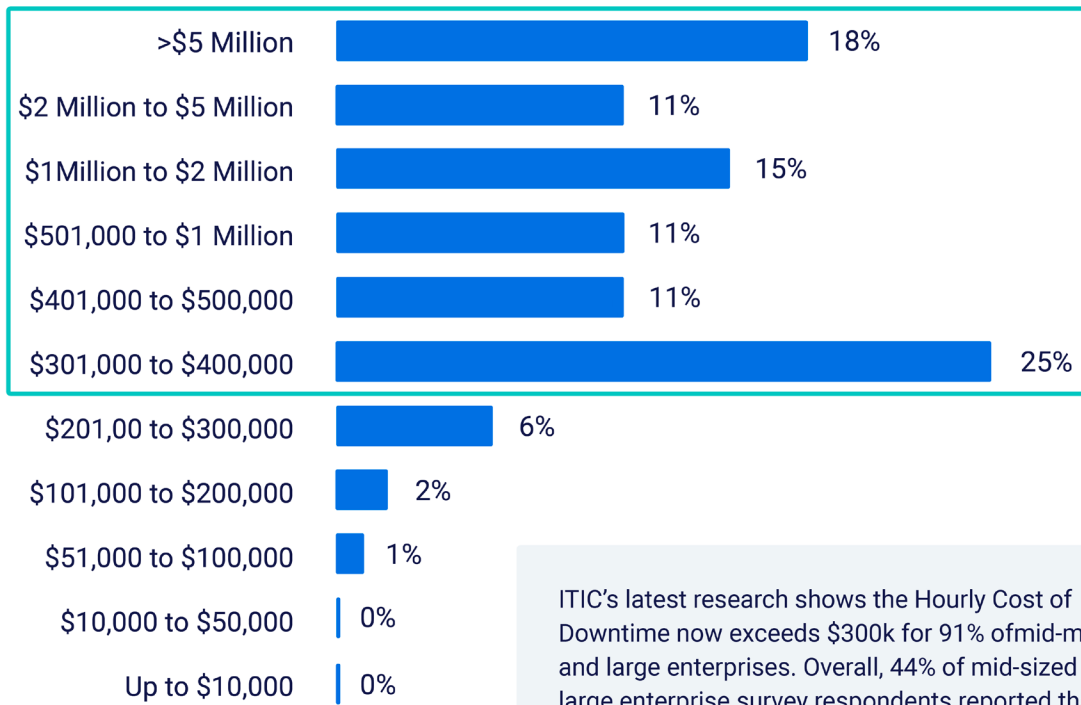
## 1.1 Who needs this?

The increasing convergence of consumerization and globalization of business has eliminated the concept of a planned maintenance window. Gone are the weekly downtimes when IT could upgrade or migrate a server. First the windows shrunk as businesses expanded globally, and then they disappeared completely when transactions, website visits, customer inquiries, and influx of IoT data started to happen continuously. Remember when businesses had paper-based credit card processing as a backup solution? Those machines are gone! Today credit card processing is expected to work all the time, and rejections or retries because of downtime have become unacceptable. Single Sign-On solutions (SSO) are another example. Because of ongoing security threats they have become the standard across the enterprise—and SSO needs to work all the time 24x7, otherwise business stops.

In that situation, even minute-long outages will cause a storm of help desk calls and can lead to a delay in business transaction processing.

ITIC's 2022 Global Server Hardware and Server OS Reliability Survey shows that HA requirements are growing dramatically. The cost of an hour-long outage is estimated at over \$100,000 and over 44% of organizations estimated that cost to exceed \$1 Million. Since 2021, only one (1%) percent of respondents said a single hour of downtime costs them \$100,000 or less.

## Ninety-One Percent of Firms Say Hourly Downtime Costs Exceed \$300K in 2022



Copyright 2022 ITIC All Rights Reserved  
 Source: ITIC 2022

ITIC's latest research shows the Hourly Cost of Downtime now exceeds \$300k for 91% of mid-market, and large enterprises. Overall, 44% of mid-sized and large enterprise survey respondents reported that a single hour of downtime can cost their businesses over \$1M.

We can easily illustrate this with an example: Imagine a brand that may have thousands of locations and eCommerce with payment processing of 100 business transactions per second. If the average transaction is \$85 and there is an outage or unplanned downtime of five minutes, then that business will miss out on 30,000 transactions—and be out \$2.5M in revenue. This disappoints 30,000 customers directly, plus it spreads to the friends and family they talk to.

## 1.2 Customer case study: ClickUp

**ClickUp**

ClickUp™ is a SaaS project management and collaboration company experiencing massive growth with customers spanning individuals to large enterprises supporting over 100,000 teams globally.

**EDB Postgres Distributed** (PGD) has contributed to the 99.99% uptime for 12 consecutive months of ClickUp’s seven-node global deployment.

### 1.3 Why is this so hard?

Two things make ‘Always On’ very difficult: Maintenance is necessary—the OS and the database have functionality and security patches, and those patches have to be applied. Even in the best of cases, one has to plan for approximately four such events per year. Innovative software, like Postgres, also has annual major releases and many users want to take advantage of the new functionality. One such maintenance operation can easily consume the majority of the downtime budget. On top of this, one has to budget for potential schema changes or data management operations that may require exclusive locks.

HA Rating	Downtime/month (days:hours:min:secs)	Downtime/year (days:hours:min:secs)
99%	00:07:18:00	03:15:36:00
99.5%	00:03:39:00	01:19:48:00
99.9%	00:00:43:48	00:08:45:36
99.99%	00:00:04:23	00:00:52:34
99.999%	00:00:00:26	00:00:05:15

# 2. Overcoming PostgreSQL HA limitations with EDB PostgreSQL Distributed





## Overcoming PostgreSQL HA limitations with EDB Postgres Distributed

This section gives a good overview of how Oracle RAC works in environments that demand four or even five nines of availability, its shortcomings, and where organizations may run into weaknesses of the shared disk/single database approach.

Up until now, Postgres mostly provided streaming replication-based HA solutions built on a primary/standby architecture, which are less capable of handling downtime-free maintenance and generally need a minimum of 20-30 seconds to address underlying hardware and network failures. These solutions are generally not suitable for above 99.99% availability requirements.

With Postgres Distributed, EDB has introduced a Postgres database extension that enables Extreme High Availability (EHA) for Postgres clusters with up to 99.999% availability. It uses logical replication of data and schema in a mesh-based multi-master architecture, plus a robust set of features and tools to manage conflicts and monitor performance. This means applications with the most stringent HA demands of 99.999% availability can be run with confidence on Postgres.

When deployed on premises, many Oracle RAC deployments require special hardware to get the benefits of RAC. In the cloud, RAC is only available on the Oracle Cloud (see Oracle Database Support for Non-Oracle Public Cloud Environments (Doc ID 2688277.1)). EDB Postgres Distributed however, can use commodity hardware, is deployable in every major cloud and can be deployed in every data center. This flexibility, combined with EDB Postgres' native compatibility with Oracle PL/SQL, makes EDB Postgres Distributed the only credible alternative to Oracle RAC on premises and in the cloud.

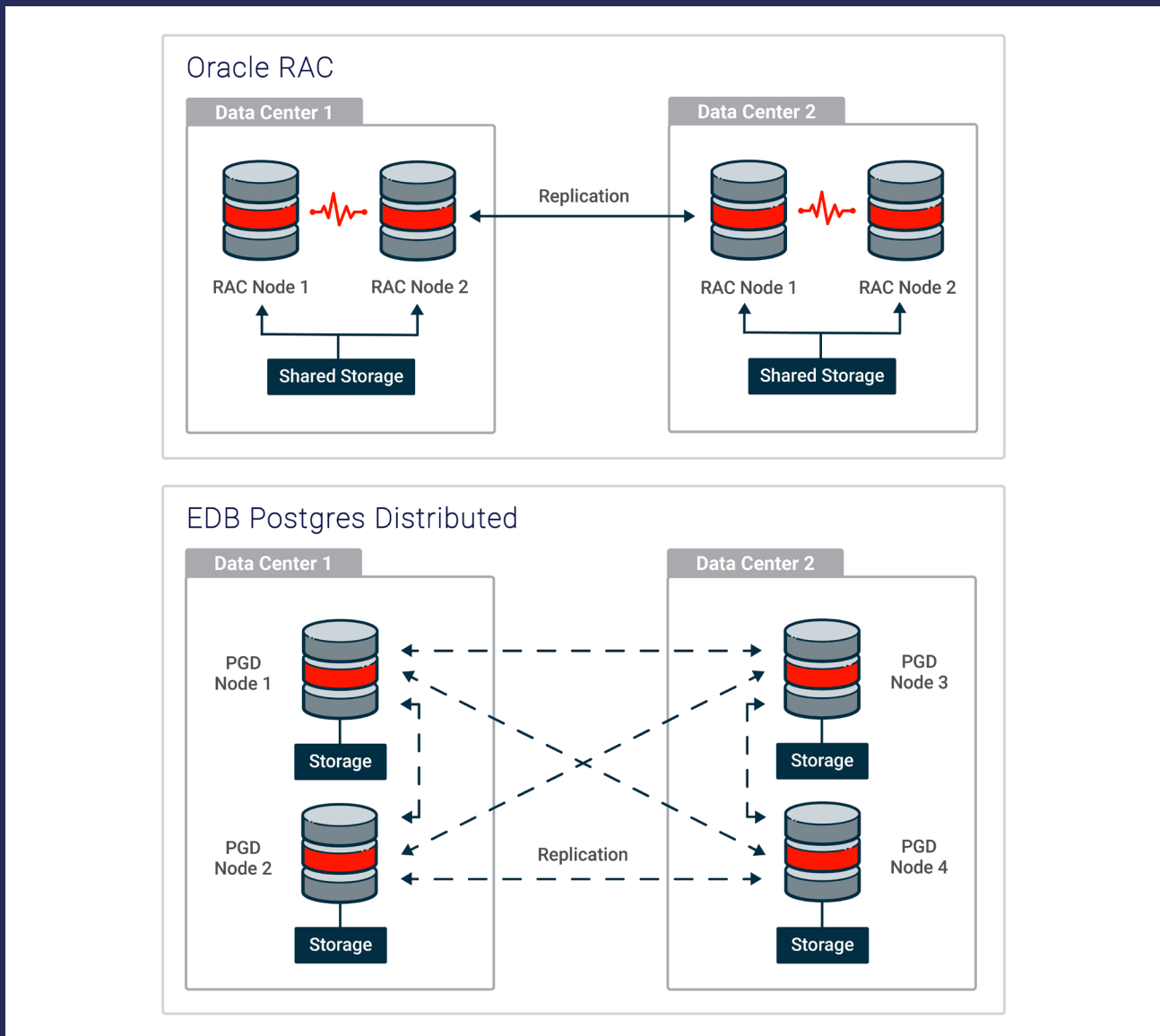


Figure 1: Comparing Oracle RAC shared disk architecture (top) with EDB Postgres Distributed **shared nothing** architecture (bottom) in a two data center setup

EDB Postgres Distributed can be deployed in multiple architectures, depending on the use case requirements for hardware redundancy, data center redundancy, and recovery time objective. Figure 1 shows a high-level comparison of an Always On configuration explicitly designed to achieve extreme high availability for the most demanding use cases, including Data Center redundancy. EDB Postgres Distributed combines hardware redundancy, logical replication, and mesh networking to achieve a 99.999% available database infrastructure.

## Key points:

- EDB Postgres Distributed's multi-master based replication does not require time consuming operations of failure detection and verification, followed by promotion and failover that are required by primary/ standby replication systems such as Postgres streaming replication or AWS Aurora. With EDB Postgres Distributed, the application can simply use another master, without delay.
- Multi-master architectures make it easy to apply upgrades to one node at a time, as logical replication works well across different database versions.
- During OS or hardware maintenance operations, traffic simply gets redirected away from the current node, and after the node has caught up, it can resume taking traffic.

This has convinced leading businesses to build on EDB Postgres Distributed for their mission-critical digital business applications.

EDB Postgres Distributed capabilities convinced Thales, a leading provider of security software, to combine SafeNet KeySecure and EDB Postgres Distributed to provide extreme high availability single-sign on capabilities for their enterprise customers.

ACI, a supplier of global payment solutions, leveraged EDB Postgres Distributed's underlying conflict handling and data consistency capabilities, to handle real-time processing of high-value financial transactions in a distributed highly available model complemented with elegant conflict resolution capabilities.

# 3. Overcoming PostgreSQL HA limitations with EDB Postgres Distributed



## Overcoming PostgreSQL HA limitations with EDB Postgres Distributed

Recent cloud-native approaches, such as AWS Aurora, still struggle to achieve availability ratings that are truly 'Always On'. As of this writing (February 2023), AWS states "If the DB cluster has one or more Aurora Replicas, then an Aurora Replica is promoted to the primary instance during a failure event. A failure event results in a brief interruption, during which read and write operations fail with an exception. However, service is typically restored in less than 120 seconds, and often less than 60 seconds." <https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/Concepts.AuroraHighAvailability.html> and the AWS SLA is limited to 99.99% (<https://aws.amazon.com/rds/aurora/sla/>). This means that three or four unplanned failovers can easily exhaust the time budget, prior to any planned maintenance operations on the schema or the data! Solutions like Aurora suffer from the same weakness that any single-primary (a.k.a. single-master) technology has: failures have to be detected and verified, before promotion and failover can happen. These steps take time, and during that time the service is not available.

# 4. The business case for making the switch



## The business case for making the switch

The costs for Oracle RAC licenses, annual maintenance, proprietary hardware, and enterprise grade SANs quickly add up. EDB Postgres Distributed which uses a subscription model and can run on commodity hardware that is available in every cloud, provides a significantly more cost-effective solution. EDB's internal ROI calculations indicate that even with extensive discounting, EDB Postgres Distributed is likely a tenth the cost of RAC when considering a three year cost of ownership.

This phenomenal ROI does not even take into account the increased deployment flexibility that customers derive from moving to Postgres. PostgreSQL, and EDB Postgres Distributed, run in every cloud and support most leading operating systems. While cost is key, and EDB Postgres Distributed provides a very cost effective solution, the ability to shutter data centers and to adopt new deployment technologies, such as containers, Kubernetes, and the public cloud are key differentiators for customers that are focused on innovation and differentiation.

# 5. Conclusion





## Conclusion

EDB Postgres Distributed is the first true way for businesses to run their Tier 1 applications on Postgres with up to 99.999% of availability.

Postgres Distributed from EDB runs on every major cloud, on all key operating systems and uses commodity hardware. This gives businesses a flexible and cost-effective way to modernize the application stacks for mission critical digital business applications.

[Learn more about EDB Postgres Distributed](#)



## About EDB

EDB provides enterprise-class software and services that enable businesses and governments to harness the full power of Postgres, the world's leading open source database. With offices worldwide, EDB serves more than 1,500 customers, including leading financial services, government, media and communications and information technology organizations. As one of the leading contributors to the vibrant and fast-growing Postgres community, EDB is committed to driving technology innovation. With deep database expertise, EDB ensures extreme high availability, reliability, security, 24x7 global support and advanced professional services, both on premises and in the cloud.

This empowers enterprises to control risk, manage costs and scale efficiently. For more information, visit [www.enterprisedb.com](http://www.enterprisedb.com).



# EDB Postgres Distributed Always On: **The End of the Reign of Oracle RAC**

© Copyright EnterpriseDB Corporation 2023  
EnterpriseDB Corporation  
34 Crosby Drive  
Suite 201  
Bedford, MA 01730

EnterpriseDB and Postgres Enterprise Manager are registered trademarks of EnterpriseDB Corporation. EDB, EnterpriseDB, EDB Postgres, Postgres Enterprise Manager, and Power to Postgres are trademarks of EnterpriseDB Corporation. Oracle is a registered trademark of Oracle, Inc. Other trademarks may be trademarks of their respective owners. Postgres