BIG ANIMAL

The Ultimate Playbook for Moving Your PostgreSQL Database to the Cloud

POWERED BY EDB

EDB

CONTENTS

INTRODUCTION

For organizations looking to move databases into the cloud, PostgreSQL has long been a powerful ally—flexible, scalable, and dynamic. However, as they look to effectively leverage PostgreSQL and achieve a cloud-native, fully-managed database, they find themselves facing a range of challenges.

Effectively and efficiently migrating your applications to the cloud isn't easy. In the wrong hands, it can be near impossible to achieve all the mission-critical goals that you've set for your transition.

EDB prides itself on experience, know-how, and success in helping businesses across industries take advantage of PostgreSQL. Whether it's facilitating or advising, we strive to ensure that every enterprise that sees their future in the cloud is able to achieve that future, best as possible. It's for this reason that we've created this guide: a roadmap and manual to help you understand how you can reach a fully-managed PostgreSQL database in the cloud, how you can avoid pitfalls like overspending and lost assets, and how you can make the most of the cloud when you get there.

Is Your Cloud Provider the Best Option?

1 Is Your Cloud Provider the Best Option?

When **migrating an existing application to the cloud** or developing a new one, your database plays the same critical, strategic role it does in an on-premises environment. For a developer that has already made the decision to capitalize on PostgreSQL's robust community, it can seem like an obvious next step to embrace their preferred cloud provider's native PostgreSQL service.

Yet such an "obvious" decision can result in negative downstream effects if developers don't fully investigate the limitations of a cloud provider's PostgreSQL offering. Whether you've chosen Amazon Web Services, Microsoft Azure, or Google Cloud Platform, here are four things to consider before you choose.

a. Level of control

One of the most significant challenges when adopting a cloud provider's PostgreSQL database is how much control of your database, operating system, and software stack you cede.

Certainly, the advantage to a managed cloud database is manifold. By relying on the cloud provider to automate and manage such administrative tasks as provisioning capacity, installing and maintaining database software, and performing back ups, application developers free up resources to focus on feature development and innovation.

In the case of cloud providers, however, the abstraction of such administrative tasks can bring with it a parallel abstraction of certain advanced configuration options, such as the ability to choose where your data is located. In some cases, such control is unnecessary—a standard, off-the-shelf application, for example, may not require customized control—but for many developers, the inability to obtain root access to the database or super user access to pg_tables creates constraints, particularly when developing complex applications.

Ultimately, the cloud databases offered by cloud providers require blind faith in their PostgreSQL database instead of the ability to control the compute, storage, functionality, and database internals.

b. Ease of migration

Application owners migrating from an existing on-premises database to the cloud face different considerations depending on the situation. For those looking to move an existing PostgreSQL-based application to the cloud, the migration is comparably simple, though developers will want to confirm that their specific configuration can be replicated in the cloud.

Many cloud databases limit which PostgreSQL versions they support, size and scalability within a region, or the amount of storage or number of instances you can run simultaneously.

For applications that currently run on a non-PostgreSQL database, the migration is obviously more complex. While cloud providers like to tout ease of migration and may provide access to tools to simplify the process, the reality is that many of those tools are relatively basic open source tools that provide simple advice and execution processes, but fail to provide meaningful guidance and automation as it relates to application impact.

This is particularly true of **Oracle to cloud PostgreSQL migrations** due to the high number of Oracle-specific queries that typically get built into applications. Without a sophisticated migration tool that can help smooth over some of those challenges, application owners may find themselves suddenly in the midst of a complex and costly migration incurring expensive professional services fees to help with application changes.

c. Vendor lock-in and rising, unpredictable costs

One of the biggest considerations in working with a cloud provider's PostgreSQL database is flexibility if you decide to make a change or support a multi-cloud deployment. Unsurprisingly, all the cloud providers offer dramatic incentives to remain wholly on their platform. In the case of Amazon Aurora, for example, AWS has substantially modified PostgreSQL, making any future transition much more difficult.

Moreover, the lack of configuration options as outlined in the previous section can help drive costs up unpredictably given complexity across CPU, memory, storage, backup, data transfer, etc. In some instances, scaling storage down is disallowed, impacting customers' flexibility and cost control efforts.

d. Open source community engagement

One major benefit of using an open source database such as PostgreSQL is taking advantage of the commitment, responsiveness, and participation from the community. Cloud generalists, however, are typically unable to stay close to the database. They often lag with bug fixes and security updates, are unable to contribute new features that their customers need, and typically cannot keep up to date with the newest version.

Without a direct line to the open source project, it can be difficult to advocate for a desired feature or modification.

Once you make the choice to use an open source provider, you need to start considering what your cloud architecture will function.

Key cloud architecture considerations

2 Key cloud architecture considerations

More and more applications are being built in cloud environments, for example on orchestrated deployments of virtual machines, load balancers, and other services, and increasingly as microservices deployed as containers in Kubernetes and similar orchestration platforms.

While it's clear that deploying and using technologies such as Kubernetes requires a different approach to a traditional "bare metal" deployment, in reality that remains true of any cloud deployment. Engineers need to understand how to make proper use of the services provided by their cloud vendor, and structure their applications and deployments to ensure data remains secure, highly available, and quick to access. Let's look at some of the key aspects of cloud hosted PostgreSQL instances and what you should consider when designing your deployments.

Consideration is given to DBaaS services from the cloud vendors and those provided by database vendors, as well as "home rolled" deployments on virtual instances or **in Kubernetes.**

a. How do you keep your database secure?

In a traditional database deployment in a privately hosted environment a lot of emphasis is placed on the fact that the entire network and hosting facilities are not public and can only be accessed by authorized personnel (or applications) from within the organization. Trust is placed heavily on the physical security of the servers, and controls on the network such as border routers with strict intrusion prevention rules (firewalls) and intrusion detection.

When we deploy in the cloud we're running on someone else's servers, in someone else's data center, on a network that's designed to be accessed via the public internet.

Physical Security

The **physical security of the systems** is largely out of our control. We must rely on the cloud provider to take appropriate measures to ensure their facilities and the equipment within them are secure. When selecting a cloud provider, be sure to research and understand the work they've undertaken to ensure the security of your data. **System Organizational Control** (SOC) 1, 2, and 3 reports are commonly seen; these are audit reports defined by the American Institute of Certified Public Accountants that assess five "Trust Service Principles" (security, availability, confidentiality, processing integrity, and privacy) at different levels. Other standards to look out for include ISO/IEC 27001, 27017, and 27018.

Network Security

When running in the cloud, we cannot rely on our own border routers (or internal firewalls) to prevent connections to our database servers. Most large cloud providers provide services that can be used to provide equivalent levels of security, if utilized appropriately.

Virtual Private Cloud or VPC is a term originally coined by Amazon in AWS. In the early days of AWS, when you created a virtual instance it was effectively just a virtual machine connected to the internet. In order to provide additional security (as well as to help minimize use of IPv4 addresses) AWS created VPCs. These are essentially virtual private networks hosted in the cloud that can consist of one or more IPv4 or IPv6 subnets, spread across one or more availability zones—think datacenter—in a single region.

VPCs can have virtual instances and other services such as Kubernetes clusters and Database as a Service (DBaaS) instances deployed in them which are not directly accessible from the internet. Access to specific services with a VPC can be managed through elastic IP addresses, load balancers, VPN endpoints and other similar services, as required. A typical architecture might have one or more subnets running web servers, with public access to them being possible only through a carefully secured load balancer. The web servers might access a database instance hosted in a separate subnet within the VPC, with security rules to restrict access to only the web servers, and only to the database server port.

When deploying in the cloud, make sure that you understand how VPCs, subnets, routers and so on operate in the environment you choose. Isolate your database servers from as much as you possibly can, allowing access only from the servers that run your application. If your application doesn't need to be accessed by the general public, configure VPN endpoints so the VPC acts like an extension of your internal network, and block all other ingress.

Most cloud providers offer virtual firewalls that can be used to secure services and VPCs. Wherever this is presented as an option, make use of it to lock down access to the resource. Follow the principle of least privilege; deny all access except that which is required for operation. Don't forget to leave a secure path for your system administrators to login as needed to perform routine maintenance and fault diagnostics. This should be kept as independent from any other access paths (e.g. for users visiting a website) as possible; one option might be to only allow SSH access to the VPC from a dedicated VPN endpoint which only the administrators can connect to.

When your applications connect to the database server, consider using certificate based authentication. When using PostgreSQL this can be configured to provide mutual authentication in which both the client and server validate each other's certificates. This can help prevent man-in-the-middle attacks and "spoofing" of the database servers, should an attacker gain access to the VPC somehow. Use of certificates for application authentication is also arguably more secure and convenient than using password authentication.

Finally, it's worth noting that security is not just about keeping unauthorized users out of your application and database; it's also about ensuring your data remains available and accessible. Ensure you make use of the services provided by your cloud provider to perform regular backups—and make sure you test them! If you don't test your backups regularly, you should assume you do not have backups. It is also highly recommended to make use of database replicas as another type of backup (albeit, not one that can replace normal backups). Most cloud providers organize their infrastructure into multiple availability zones (data centers) within each geographical region. By keeping a replica of your database in multiple locations, you maximize the chances of being able to stay online in the event of an outage in any one location.

b. How do you keep your database performant?

Performance tuning PostgreSQL in the cloud is largely the same as in other environments at the highest level; however, the differences become greater the further down the stack you go—and they are quite dependent on the type of deployment you create.

If you use virtual instances to self-build and manage a deployment, then the opportunities are the greatest. You have full access to the operating system and kernel configuration, storage layout, and provisioned IOPs to guarantee minimum performance (subject to what block storage options are available), the PostgreSQL installation and configuration, and, of course, the database schema. You can also select from a much wider range of machine instance types, with the characteristics best suited to your needs. You can make that access available not just to your own team, but also to dedicated database support vendors. However, this type of deployment requires far more design work and effort to get up and running and manage than DBaaS solutions.

If you choose a DBaaS solution, your options are far more limited. Cloud vendors typically offer only a few instance types to choose from, give you no opportunity to tune the operating system or kernel, or to utilize different disks to optimize speed vs. cost for hot or cold data, and frequently only allow limited tuning of the database server configuration. While this level of control might be suitable for some applications, it may not be enough for very large or very high velocity applications.

DBaaS solutions built by third parties on top of the infrastructure provided by the cloud vendors may offer additional options, as well as the opportunity for the vendor to assist with more complex tuning needs than the big cloud vendors will.

Deployments on Kubernetes can also vary wildly in the options available. The Kubernetes deployment itself may be provided as a service, or deployed manually onto virtual instances. The latter will give you far more control over the underlying operating system of course (such as the ability to select a specific kernel scheduler), but you still may be limited in what can realistically be tuned to meet the needs of demanding workloads—especially if other components of your application must share the same Kubernetes infrastructure. The deployment may also get particularly complex if you want to make use of tablespaces with different IO characteristics.

c. What failures do you need to tolerate?

Failures are inevitable, whether due to issues with the database software, operating system, or underlying virtual or physical hardware on which it all runs. Cloud environments make it very easy to have geographically diverse replicas, which can also be utilized for offloading report workload or load balancing read-only queries. More complex systems such as **EDB's BDR (Bi-Directional Replication)** can offer both read and write servers in multiple locations, as well as "always on" fault tolerant architectures.

For the most simple deployments, a single database instance with backups taken at a suitable interval—optionally with transaction log archiving to allow point-in-time recovery—may suffice. If your requirement for recovery time (RTO) is low, this may be a relatively cheap option. If your requirement for recovery point (RPO)—how recently in time data can be recovered to—is long, even more can be saved by not archiving transaction logs.

For deployments where RTO is higher, replicas are essential; and if it's crucial that no committed transactions can be lost, synchronous replication to at least one node of a three node or larger cluster can be used.

Geographic redundancy (GRO) is also important in some environments. In the cloud, clusters can often be spread over multiple availability zones within a single region; you might think of this as having replicas in different buildings, in the same city. This protects against localized failures such as loss of power or connectivity to one of those buildings.

In deployments with a very high GRO, you might consider having one or more replicas in an entirely different region; typically a different state, country, or continent. This can protect against widespread failures of an entire cloud region from natural disasters, such as an earthquake.

d. What are the cost implications?

It's relatively easy to estimate the cost of running a PostgreSQL instance in the cloud; providers often supply a pricing calculator to help with this. However, it can be common to overlook or underestimate the cost of the storage required which can be surprisingly high, especially when provisioned IOPs are allocated. Make sure that you keep this in mind, as well as the fact that the storage costs may differ between DBaaS services and plain virtual instances with attached block storage that could be running PostgreSQL.

Data transfer between DBaaS instances and virtual instances within the same region is often free of charge, however, cross-region may not be, and internet egress usually isn't (although ingress is often free). Consider what the data flows will look like in your application infrastructure, and weigh the expected costs into your choice of cloud provider and application design. Network costs for database use are typically very low compared to the storage and compute costs (whether DBaaS or virtual instances), but it can get expensive if you have very high volumes of egress to the internet or cross region.

If you have one or more replicas, expect to multiply the cost for compute and storage by the number of nodes. While some providers may offer discount rates for replicas, these are often very small. If you're building your cluster yourself and have cross-region replicas, the network traffic from the primary to the replica may also incur additional costs.

Backup storage increases costs as well, which can be dependent on the size of your database as well as the amount of change since the previous backup. This can be extremely difficult to estimate, though thankfully the costs are usually relatively small.

Finally, if you're building your own cluster using virtual instances, you may need to "Bring Your Own License", depending on your choice of engine. Licensing costs are typically included in DBaaS offerings.

e. How will you support your database?

Support can be critical with production systems; even with an expert team on staff, it's unlikely they'll have the in-depth knowledge of the database engine of a specialist database company with a deep understanding of the code.

As we discussed in the first chapter of this eBook, when using a DBaaS service, you will typically have limited access to key components of the operating system and virtual hardware. This makes it difficult to diagnose and fix both major and minor issues. The major cloud providers also have limits to the support they will offer; typically, ensuring the database is up and running and you can connect to it is about as involved as they will get. Specialist database companies such as EDB may offer more options , such as remote DBA services, however they too will be limited due to the lack of access to the underlying platform.

If you build your own cluster on virtual instances then you have far more access; you can login to the instances, modify kernel parameters and disk configuration, and change any aspect of the Postgres configuration you like, with or without the help of a third party support vendor.

This makes third party DBaaS options from vendors such as EDB an ideal solution. These offer all the convenience of a DBaaS service, plus you or the vendor (or both) will be able to access the operating system of the servers on which the database is running, allowing the vendor to provide a significantly higher level of support, and flexibility.

But even once you've answered all these questions, the journey to your ideal new environment can be a tricky one. It helps to have a guide.

The importance of a database specialist when moving to cloud

3 The importance of a database specialist when moving to cloud

80% of EDB's customers are planning to run their databases in the cloud; over 40% use EDB Postgres in the cloud today. Gartner has forecasted that 75% of all databases will be deployed in the cloud by 2022. In the fray to get there, RDS, Azure Database, and CloudSQL appear to rule the roost, and we are often asked if database specialists—such as Oracle, Percona, MongoDB, or EDB—still matter when the world of databases is moving to the cloud. Why do you need database software expertise, if somebody else runs the databases for you in the cloud?

Moving to a managed cloud service

Let's analyze what happens when an enterprise moves their databases to a managed database cloud service, such as Amazon's RDS. **Traditionally, successfully setting up a database required multiple organizations to participate:**

1. Compute infrastructure — that is the network, the CPU, the storage and the OS — had to be configured and made available. Typically, this was done by the Sys Admin Team. In some companies, those roles can be further subdivided, often making things complex and slow.

2. The database infrastructure — that is the database software, storage configuration with tablespaces, high availability, backup/recovery, geographic redundancy, security, etc. — had to be deployed. This is the job of the Infrastructure DBA.

3. Data models, tables, indexes, queries, stored procedures, etc. — these have to be designed, maintained, and continuously tuned to support changing business requirements. Tables, and indexes have to be mapped to the right storage devices to guarantee cost effective performance. Traditionally, this is done by the Application DBA.

#1 and #2 are focused on infrastructure, and are at the heart of the Cloud Service Provider's (CSP) strengths. Setting up and running reliable and resilient infrastructure is the core of what they do. This is what makes the move to the cloud so attractive: no more dealing with infrastructure, networks, backups etc, or what a CIO recently called "non value-add repetitive drudgery" during a discussion panel focused on migration to the cloud.

When database expertise matters

But the Application DBA work of data modeling and performance tuning remains, and the Application DBA will need support and help—especially when dealing with new technologies or making significant architectural changes, such as moving from the well understood on-premises performance environment to the cloud, or when they encounter an unexpected discrepancy with expected behavior, a.k.a. a bug.

This is when **partnering with a database expert** matters:a company that can answer difficult questions, share best practices, or address bugs and security issues quickly.

The CSPs operate the database software for their customers. They make sure it is up and running, and that the application can connect to it. Support for the software, how-to questions, best practices, or security patches are either handled through third parties, or in the case of open source software such as PostgreSQL, the customer has to wait for the community to get around to addressing the problem, and for the CSP to apply the patch. These are the situations where database software expertise matters.

It is a common misconception that there is no difference between RDS or other CSPs running a database in the cloud, and a database software specialist like EDB doing the same thing. But the differences are fundamental. For one, the CSP does not provide support for the software—they only operate the software as outlined above. In case of a problem, they refer the customer to a third party, or they wait for the open source community to address the issue.

The second key difference is more fundamental, and matters mostly to enterprises who view data as a differentiating business asset. The database software specialist, who is invested in the software, will work with their customers to understand their needs, and make the software better—the CSP is invested in the platform only, not the software. The CSP has a roadmap for the platform; the software specialist has a roadmap for the database capabilities.

Let's explore two real-life examples illustrative of why a customer-driven software roadmap matters for businesses who see data as a differentiating capability.

1. A northern European bank approached EDB to address a specific limitation in Postgres: at high logging levels, user passwords that are stored in the database are written to the log file when the password is being changed using the ALTER USER command. This inhibited the use of Postgres for certain classes of applications at the bank. EDB, acting in our role of a database software specialist, modified EDB Postgres Advanced Server to provide a special control flag to address the issue and redact the password in the log files.

2. One of the largest credit card companies started leveraging Postgres for very high transaction volume applications on exceedingly powerful hardware. The customer encountered several situations where Postgres approached or even crossed the transaction wraparound thresholds, and systems had to be taken offline for maintenance. EDB took a leading role in the community to improve vacuum and bloat handling, which allowed this customer, and others, to leverage Postgres for new levels of uses that previously were problematic.

That kind of partnership is only possible with database software specialists, who are invested in the underlying software and have the resources to drive a roadmap.

What are the business risks when you trust the cloud provider to be your database partner?

- **Bad design decisions** — The CSPs expertise and market differentiation are focused on the infrastructure, not the database software.

- **Lower SLA when it really matters** — If there is a bug or a disruption of critical service because of a software issue, that is when expertise really matters. Only database experts can have an expert engineer on the phone in minutes to help. Operators focused on running the software cannot do that.

- **Your data, your strategic differentiator, is treated like a commodity** — The CSPs roadmap focuses on the platform, not on driving more value from new database software features.

What are the business outcomes when you partner with a database expert?

- Accelerate time to market by avoiding costly design mistakes
- Make optimal infrastructure and software decisions, as the CSP and the database expert each play to their strengths
- Build strategic relationships by helping the CSP and the database provider shape their roadmaps to help you maximize the strategic value of having chosen a specific platform and a specific software.

Database specialists may be more important than ever. They help customers get to the cloud quickly, and operate successfully in the cloud. The platform provider and the software specialist each have their own important roles in support, bug fixing, and in the customer focused roadmap for the platform and the software. Trusting a cloud platform provider with the software is just as foolish as trusting a software provider with the cloud platform.

Which brings us to the BigAnimal in the room.

EDB's BigAnimal is designed to help you achieve cloud success

4 EDB's BigAnimal is designed to help you achieve cloud success

Recently, **we announced the launch of BigAnimal**—the first fully-managed cloud database service with industry-leading Oracle database compatibility and backed by a world-class team of PostgreSQL experts, all packaged up and running in your Microsoft Azure account. BigAnimal supports both PostgreSQL and **EDB Postgres Advanced Server** and is designed with the needs of enterprises in mind.

Now, you might be asking "So what?" Or, "Why now?" This chapter explores why we built BigAnimal, and why we think it matters.

a. Go with the customers

Throughout EDB's history we've continually focused on customers' needs. Over time, this has run the gamut from old-school bare metal machines in on-premises data centers and colocation spaces, to early virtual machine deployments, to the cloud, and to containerization. Each step in this evolution has come with its own opportunities and challenges, and it's been EDB's mission to meet customers where they are—taking advantage of opportunities and overcoming the challenges.

Fully managed database-as-a-service products are perhaps the biggest shift in this evolution. In some ways, with a managed database service your database is no longer "yours"—in exchange for having a third party manage your database, you give up control over how that database is configured and operates.

This is a big deal: many of EDB's customers are advanced users of their databases and rely on extremely high performance, sophisticated behaviors, and advanced configurations to make their businesses run. For many of them, existing managed database services just aren't an option—they're too restrictive. For others, perhaps they're willing to make compromises, but they'd really prefer not to have mundane technical problems impact their business decisions.

Finally, and perhaps most importantly, we're still in the early days of cloud migrations. Cloud database service usage has exploded over the past few years, but there is still a vast array of on-premises databases that haven't yet made the leap. Many of these databases are Oracle-based, and support for Oracle in major managed database services is limited at best; others have more stringent high availability requirements than current database services are prepared to meet. We need innovation in cloud database services to get these databases and the applications they support to the cloud.

b. Elevating PostgreSQL in the cloud

We built BigAnimal to solve these problems. EDB is a database company to the core; we know Postgres, and we think we have the talent and the tech to bring something new to the table in database-as-a-service. We like to frame our core approach along three lines: expertise in Postgres, compatibility with Oracle, and "continuous" availability. Let's dive in to each of these:

Real Postgres expertise, applied

This one is easy—EDB has a decent claim to being the Postgres company in 2021: a significant fraction of all open-source Postgres development comes from EDB employees, and we employ Postgres fanatics across the board to work on tooling, automation, support, and consulting. Expertise alone doesn't make a great product, though—what matters is applying that expertise! With BigAnimal, we're committed to supporting customers on the latest and greatest Postgres versions, providing break-fix support assurance, and offering more technical control than can be had with other cloud Postgres services. On top of that, we're layering on our support and professional services offerings such as Cloud DBA to raise the bar in terms of access to expertise. And finally, we're taking what we're learning back into the community to help make Postgres even more rock solid, reliable, and manageable than it already is.

The impact of EDB's Postgres culture on the BigAnimal product is difficult to overstate. We think that databases are too important and too complicated to be left in the hands of generalists, and we're here to make that possible in a managed cloud service.

Oracle-compatible PostgreSQL

It's 2021, and there are two ways to get an Oracle-compatible database in a managed service on a major cloud: buy AWS's Oracle RDS - with all of the licensing complexity that comes with it - or buy it from Oracle Cloud. Azure and GCP customers are basically out of luck.

EDB is no stranger to supporting sophisticated migrations away from Oracle. Helping customers migrate from Oracle to Postgres has been our bread and butter for years. With BigAnimal, we're taking the native Oracle compatibility built in to EDB Postgres Advanced Server to a fully-managed service, allowing customers to take their Oracle databases straight to a cloud managed service without a painful multi-stage migration process.

Always-On Postgres

Postgres' tried-and-true streaming replication implementation is at the heart of many mission-critical applications. It's solid, reliable, and flexible enough to solve a lot of **high availability** problems. Most of the major Postgres database-as-a-service offerings have it at the core of their availability architecture.

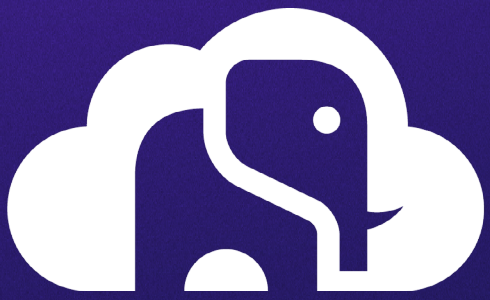
Streaming replication is somewhat limited, though, when it comes to more sophisticated high availability topologies. Failure detection and replica promotion times are long—even AWS's Aurora suggests that "most failovers complete in 120 seconds"—which limits availability targets of most cloud services to 99.99% and excludes a wide range of applications that might otherwise be a good fit for the cloud. High availability based on streaming replication simply isn't an option for systems that need the availability of, for example, Oracle RAC.

EDB's BDR is a logical replication based solution that aims to solve these problems, and BigAnimal is bringing it to a fully managed database-as-a-service. With BDR-based managed Postgres, customers will be able to support the needs of more stringent applications by taking advantage of BDR's capabilities in active/active replication and fast failover. These capabilities are unmatched in open-source relational database services today and open up new possibilities for bringing mission-critical applications to fully managed cloud services.

c. BigAnimal

The cloud is complicated, and databases in the cloud are more complicated. Your most demanding applications need a cloud Postgres service that's purpose-built for the requirements of the most stringent enterprise applications. With BigAnimal, we're working to bring our expertise and technology to bear to solve these problems and aid our customers in their cloud journey.

Learn more about BigAnimal



Your future in the cloud starts here

With all of the opportunities that the cloud presents, it's no surprise how eager businesses are to migrate their databases and applications. In order to take full advantage of these benefits, however, there are myriad factors to consider. Each component of your journey has the power to affect your total outcome. You want to thrive. You want to do it right.

We hope the insights shared here give you a better sense of not only what your enterprise can achieve, but how best to achieve it. With our years of experience helping organizations of all sizes leverage the power of PostgreSQL, we've seen just what a company like yours can reach for. And we want to help.

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 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**.

The Ultimate Playbook for Moving Your PostgreSQL Database to the Cloud

POWERED BY EDB