

Moving Oracle workloads to the cloud: 5 key decisions

Highlights

- Choosing your workloads
- Choosing your database
- Choosing your cloud
- Choosing your tools
- Choosing your partner

5 key decisions to make when moving your Oracle workloads to the cloud

If your organization is considering moving your Oracle workloads to the cloud, it is likely part of a broader initiative to move enterprise workloads to the cloud. Many organizations are actively exploring ways to deploy new or upgraded applications to the cloud whenever possible. A trusted enterprise partner such as IBM or EDB can make the difference between a productive transformation and a convoluted ordeal.

Choice 1: Your workloads

While there is no shortage of applications you might want to move, not every workload is equally suited for cloud deployment. You must consider the operational characteristics of your applications in determining how well and how easily a potential migration might proceed. For highly sensitive or mission-critical applications that are operating well in their current environment, a move to the cloud may not be worth the risk.

Expectations around transaction processing must be taken into account. If your organization is coming from a bare metal environment, where the norm is 8,000 transactions per second (TPS) or more in an Oracle environment in your data center, it will not be the same in the cloud.

On-premises, you may have the luxury of a massive, fiber-channel-connected box where there is an average of 150,000 input/output operations per second (IOPS). You won't see that kind of high-end performance on an individual system in the public cloud. The IOPs might be only half of your organization's required capability, even in an ideal scenario with no overhead.

To properly prioritize workloads, experts recommend segmenting applications into different tiers based on their business importance and fitness for the cloud. In the end, IT must ensure that the organization makes a decision that takes the benefits and trade-offs into full consideration.

Choice 2: Your database

Once you select your workloads, you need to choose your database. It's important to realize that database migrations to the cloud should not be treated as a "lift and shift" exercise. Technologies in the cloud do not operate in the same way as comparable technology on-premises. For example, Oracle's database technology was built to run best on bare metal. It is very well tuned for scale-up scenarios, especially on large servers. In addition, Oracle's RAC architecture is geared toward optimizing use of shared disk storage. These factors generally make Oracle databases ill-fitted for the cloud.

While the Oracle Cloud has introduced a number of proprietary features to cope with this inherent mismatch, many organizations are reluctant to become further entangled with Oracle in their move to the cloud. In fact, cloud migration projects are a great opportunity to explore alternative database approaches, such as Postgres.

Because of this, it is not surprising to see such a rapid growth of interest in Postgres as a strong Oracle alternative. The PostgreSQL database management system gained more popularity in the [DB-Engines.com ranking](#) within the last year than any of the other 341 systems they monitor. This led DB-Engines to name [Postgres the DBMS of the Year for 2017 and 2018](#).

There is more than one way to deploy Postgres in the cloud, however, and it is important to understand the benefits and limitations of each Postgres option in order to forge a comprehensive migration strategy. A few approaches to consider include:

Option 1: Self-supported Postgres

Perhaps the most straightforward option is simply to run a copy of the PostgreSQL server on a compute instance in a public cloud. This approach is analogous to running the software in your own data center. You can configure the software as you like, use any extensions your application may require, log into the virtual machine directly, and tune any and all database and operating system parameters for your workload. This option provides maximum flexibility. However, it doesn't provide the kind of "self-service" experience that a DBaaS platform would provide or the level of enterprise support from a commercial database vendor that you are likely accustomed to receiving from Oracle.

Option 2: Amazon RDS and Aurora

Amazon RDS and Aurora provide a self-service alternative to provisioning Postgres-compatible databases in AWS. To simplify deployment of RDS and provide the elasticity of Aurora, Amazon has modified core PostgreSQL code to optimize their use of the AWS cloud infrastructure. Of course, the fact that these are Amazon proprietary versions of PostgreSQL leveraging AWS capabilities means that they are only available on Amazon's cloud. And because Amazon Postgres DBaaS offerings are forks of PostgreSQL, many common extensions are not available.

These DBaaS options are appealing to developers in particular, offering fast, easy deployments without the need for operational IT support.

This "hands-off" approach does come with some downsides. Amazon DBaaS offers the same basic Postgres configurations to everyone. It simply must be this way in order for Amazon to operate AWS economically at scale. It also creates rigidity and strips away some of the flexibility and extensibility that makes Postgres so capable. DBAs and developers are limited in the configurations and tuning they can perform on the database to optimize the system to meet performance, storage, and availability requirements. You can't even log into the machines where your database servers are running. And if you can't tune the Postgres engine to optimize for an Oracle migration or take advantage of features that enhance Oracle compatibility, you may be left with a failed migration.

Option 3: EDB platform

IBM Business Partner® EDB offers the EDB Postgres Advanced Server (EPAS)—one of two databases available as part of the EDB Platform. EPAS enhances the capabilities of the open source database PostgreSQL, to deliver many enterprise features not available in self-supported PostgreSQL, some of which are shown in the table below.

The EDB Platform's EPAS also extends PostgreSQL with Oracle-compatible functionality that can be deployed either on-premises or in the public cloud. It also comes with a complete set of tools to manage the database, deliver high availability, manage backups, etc. that will be familiar to Oracle DBAs. EDB provides compatibility with native PL/SQL, the programming language that runs on Oracle database products, and includes automated migration tools to help save time and minimize uncertainty and risks in the migration process.

	Self-supported PostgreSQL	EDB Postgres
Password management	Not available	EDB Password Profiles
Authorization	PostgreSQL RLS	EDB Virtual Private Database
Auditing	Limited audit capabilities	EDB Postgres Advanced Server (EPAS) audit with DML auditing for INSERT, UPDATE, DELETE, TRUNCATE by user and database, syslog integration, etc. Manage audit logs separately from the server logs.
SQL injection attacks	Not available	EDB SQL/Protect
Encryption at rest	Do-it-yourself	Proven full-disk encryption procedure Extension of pgCrypto to support secure key management
24/7 support	Do-it-yourself	Enterprise-level SLA support with direct access to Postgres community leaders
HA/DR	Multiple open-source tools	EDB Management Tool Suite: EDB Failover Manager EDB Backup and Recovery EDB Enterprise Manager
Data redaction	Do-it-yourself	Custom data views EPAS 11: Built-in data redaction
Secure configuration best practices	Do-it-yourself	EDB Postgres Advanced Server secure technology Implementation guidelines (http://iase.disa.mil)

EDB Postgres Advanced Server can be provisioned a number of ways:

1. Manually, as with the self-supported PostgreSQL.
2. Through the EDB Ark DBaaS platform for a self-service experience similar to RDS.
3. Through the EDB Managed DBaaS Service, a fully managed, “white glove” DBaaS service for Postgres delivered in the Amazon public cloud.

If you are currently considering moving your Oracle workloads to the cloud, Amazon RDS and Aurora may be familiar. However, the [EDB Managed DBaaS option](#) allows you to experience the same ease of deployment and elasticity as with Amazon, but without the trade-offs in performance tuning, configurability, customization, and control.

Finally, the EDB Platform is supported by the EDB team of Postgres experts 24x7 and backed up by EDB’s professional development organization, staffed with many PostgreSQL community members.

Together, these capabilities make migrating to EDB substantially less risky and time consuming for existing Oracle users whether migrating to on-premises, cloud, or DBaaS environments.

In the end, it is important to understand the benefits and limitations of each Postgres option in order to forge a comprehensive Postgres strategy that best meets all your business goals and objectives.

Choice 3: Your cloud

Certainly, Oracle has a cloud, but even though Oracle does a good job of building a cloud designed to run Oracle databases well, there are other cloud platforms to consider. AWS, Microsoft Azure, Google and IBM each have their strengths. Compare the options to see how they stack up.

Option 1: Oracle cloud

It is natural to consider the Oracle Cloud as an option for moving Oracle workloads. As mentioned above, it is well optimized for running Oracle database software. Of course, the database is only one piece of the overall application environment to be moved to the cloud and typically not the piece that is leading the decision for that move. The availability of application development tools, regions in which the cloud operates, pricing and any number of other considerations usually come ahead of optimizing database performance. In addition, if you are interested in reducing your dependence on Oracle as a vendor in your migration to the cloud, moving from on-premises Oracle to Oracle Cloud will only increase your commitment.

Option 2: AWS

AWS offers a portfolio of services for almost anything you might want to use on a cloud, and has a strong culture of creating APIs for all of it. If you want to script the cloud, orchestrate, and connect services, AWS makes that more accessible. They are rapidly innovating. Of course, many of these services are only available on AWS so you need to be careful in choosing which APIs you adopt if you want to avoid getting locked in. AWS also features data transfer fees that some users find surprising.

Option 3: Microsoft Azure

A strong alternative to AWS, Microsoft Azure might be a better fit for organizations already built on a .NET ecosystem. They have a large network of partners who can help you architect and build solutions. Microsoft also has a long history of working with the enterprise, offering quality of service, terms and conditions, and SLAs that can meet the expectations of SMBs and larger enterprises consistently.

Option 4: Google Cloud Platform (GCP) vs. AWS

Google's infrastructure offers massive scale and responsiveness, which they have created for their own purposes to enable search and support video. When operating in Google's cloud, you'll benefit from their investment from a price-to-performance standpoint.

They may over-provision you beyond what you pay for. They do have fewer services, but the services they offer are innovative, such as AutoML, TPU, and Video Intelligence for machine learning, and in the data space, Google Genomics, Spanner, and Bigtable. Also, unlike Amazon and Microsoft, they have a long history of contributing to open source. Because of this, many of their services can be replicated on other environments using the projects they've created, like Kubernetes, Google Cloud Spanner, and CockroachDB, reducing the risk of lock-in.

Choice 4: Your tools

Once you've decided where you're going, it's time to figure out how to get there. Migrating a database application to a new platform can be challenging and it takes significant forethought. This is especially true when moving to a very different environment like the cloud. Because of this, it can be useful to approach it in a number of phases to make the task manageable. Luckily, there are various tools available to support your efforts in each of these steps.



Migrate schema

When migrating between different types of databases, you may need to change the schema in some way to accommodate the new database. In moving between different relational databases, this is not generally a complex process, since most of the data types will be common between them and many that are not will have simple mappings. There may be exceptions; for example the way geospatial or GIS data is represented is typically specific to a particular relational database. In some cases, certain database vendors simplify this task even further by providing a data dictionary, which is compatible with another leading vendor, (e.g. MariaDB and MySQL or EnterpriseDB with Oracle.) Of course, when you move out of the relational world, for example to a NoSQL system, this mapping can become significantly more challenging as the fundamental data models can be very different and may require a full application rewrite.

A popular tool currently for helping migrate a database schema from one relational database to another is the [AWS Schema Conversion Tool \(SCT\)](#). AWS SCT provides a project-based user interface to automatically convert the database schema of your source database into a format compatible with a target Amazon RDS instance. If schema from your source database can't be converted automatically, AWS SCT provides guidance on how you can create equivalent schema in your target Amazon RDS database.

Of course there are limitations in the schemas that AWS SCT can handle. Oracle has a very rich set of capabilities that go beyond the standard SQL supported by PostgreSQL. In fact, if you're only using standard SQL, you may not even need a full blown schema migration tool. A simple text editor would likely be sufficient to replace any Oracle-specific types with PostgreSQL equivalents.

Beyond AWS SCT, you might look to [Microsoft's Database Migration Service](#), but SQL Azure is the target of their migrations. Similar to Oracle, Microsoft seeks to keep you in their ecosystem.



Migrate data

Once the schema has been mapped, data can be transformed and transported into the new data store. The actual transport of the data can also be quite challenging, especially in cases where there is a large volume of data to be moved and where the data needs to be constantly available with minimal downtime during the move. Automated migration tools can be a huge timesaver here and heterogeneous replication can help reduce or eliminate downtime when cutting over to the new database.

In this area, there are a lot of potential tools to choose from, including a few offerings from AWS. Like SCT, AWS offers a data migration tool called [Amazon Database Migration Service \(DMS\)](#). This provides a way of streaming data into your AWS-based data stores such as RDS, DynamoDB, Aurora, and RedShift. If your data set is too big for a real time migration, Amazon offers off-line tools like [Snowball](#) for petabyte-scale data sets and [Snowmobile](#) for even more extreme data sets.

Outside of the AWS ecosystem, you could look to some of the same tools that you may be used to using on premises. There are real time replication tools like [HVR's Data Replication](#), Attunity Replicate, or even [Oracle GoldenGate](#). EnterpriseDB offers [EDB Replication Server](#) and the [Migration ToolKit \(MTK\)](#) for this task as well. IBM offers Lift CLI for rapid network transfers and the Mass Data Migration appliance for larger datasets.

For extremely large data sets that require batch mode operations, traditional ETL comes into play. There are a lot of options from the long time market leaders, [Informatica](#) to the open source leader, [Talend](#) and lots of options in between from too many vendors to try to list here.



Migrate application logic

This is a step that many overlook when thinking about migrating a database. This may not be difficult if the migration is between relational databases and the application uses only standard SQL interfaces to access and manipulate the data. Moving from Oracle to a NoSQL database, on the other hand, likely means a full rewrite of the code that interacts with the database. Even in the case of a move from Oracle to another relational database, however, it can be challenging if the application was built to take advantage of non-standard proprietary features in PL/SQL.

Unfortunately, this is probably the area that offers the least help to people migrating databases while at the same time having the potential to have the biggest impact on the success of the overall migration project. The AWS SCT provides a bit of help in this area by recommending changes to SQL statements, but can't help much in translating Oracle stored procedures, implementing specific Oracle features or converting the use of Oracle specific connectors like OCI or PRO*C.

EDB can provide some relief in this area with its database compatibility for Oracle including PL/SQL execution, OCI and Pro*C compatible connectors, and numerous other features. The EDB Platform provides compatibility with native PL/SQL, the programming language that runs in Oracle database products, and includes automated migration tools to help save time and minimize the risks of rewriting database code that may have taken thousands of man-hours to perfect and has been running without error for years.

The [EDB Migration Tool Kit](#) migrates Oracle, SQL Server and MySQL data to PostgreSQL providing online/offline schema/data migration with flexible customization and fast parallel data loading.



Test, test, test

Once you've migrated the schema, data, and application, you'll want to run a thorough set of tests to ensure that the application behaves in the same way it did before the move. When doing this, don't forget the operational characteristics of the application such as performance, availability, scalability, network speeds, and security. As described above, the cloud operates in a very different way than on-premises machines. This leads to very different operational characteristics. Any migration should take these changes into account in determining the cloud environment for deploying the application to ensure that it meets the expectations of the existing users.

Choice 5: Your partners

Moving your Oracle workloads to the cloud is an achievable goal and an experienced partner can be invaluable in getting there. It can be complex, but with the right approach, you'll mitigate risk, modernize your architecture, and be better prepared for the future.

IBM Business Partner EDB has helped many of their customers achieve this goal, and their knowledge on this subject has been earned through real-world experience [inside diverse customer environments](#). They can help make your transition from Oracle to Postgres less stressful and more successful at every stage of the journey. Here are some examples of how they can help.

Gauge your readiness

Start by identifying the applications and databases that you seek to move to the cloud. Then, with help, conduct a production readiness assessment and [Architectural Health Check](#). With a deep understanding of how you currently scale, ensure security, and manage your databases and applications, EDB can make sound recommendations about which workloads are best suited for the cloud, or how you'll need to adapt them for the cloud.

Make a plan

Rank your workloads and help you build a plan to stage your migrations. By selecting low-risk, high-value workloads first, you can accelerate your move to the cloud and earn some early wins.

Tune your workloads

EDB can provide ongoing, valuable support and services to help ensure the ongoing health and wellness of your cloud deployments. They offer [pre- and post-deployment performance optimization services](#). Expert consultants remotely evaluate your environment, review your PostgreSQL database parameters, and can make adjustments according to your workload and use case. They can guide you in using the proper instance type and server size to meet your current needs for cost and performance, and help you plan for the future.

Leverage premium management services

With the [EDB Managed DBaaS Service](#), you can extend Postgres in the public cloud with Oracle-compatible functionality. EDB Managed DBaaS allows you to experience the same ease of deployment and elasticity as with Amazon, but without the trade-offs in performance tuning, configurability, customization, and control.

The EDB Managed DBaaS Service is a fully managed, “white glove” DBaaS service for Postgres offering proactive 24x7x365 database and infrastructure monitoring by certified Postgres specialists—offering a level of support that AWS does not.

The EDB Managed DBaaS service combines a RemoteDBA offering with on-demand database provisioning and management capabilities. Additionally, the remote DBaaS operations team manages the AWS cloud infrastructure on your behalf. EDB provisions and manages a dedicated DBaaS management environment in AWS for each DBaaS customer.

Your users get the benefit of having the EDB team deploy fully managed instances for them and can also choose to perform self-service provisioning of unmanaged database instances, for example for development use cases. Additionally, EDB's consulting services are included for onboarding activities and migration assessments.

Learn more about EDB Postgres [here](#).

The journey to migrate Oracle workloads to the cloud might be complicated, but help is available. Choose solutions from the EDB Platform—such as the EDB Postgres Advanced Server—to enhance the capabilities of open-source PostgreSQL. Or choose Oracle services from IBM—recently named a leader in the 2020 Gartner Magic Quadrant for Oracle Cloud Application Services.¹ IBM offers a full range of modern capabilities and solutions for Oracle customers using advanced analytics, blockchain, AI, cognitive services and more.

Both EDB and IBM have the experience and expertise to help you configure and tune your cloud deployments, assist with implementing your migration, and then provide the long-term support and services needed for your success.

For more information, visit: [Oracle Application Management IBM Services for Oracle](#).



© Copyright IBM Corporation 2020

IBM Corporation
New Orchard Road, Armonk, NY 10504
Produced in the United States of America
August 2020

This document is current as of the initial date of publication and may be changed by IBM at any time. Not all offerings are available in every country in which IBM operates.

It is the user's responsibility to evaluate and verify the operation of any other products or programs with IBM products and programs. THE INFORMATION IN THIS DOCUMENT IS PROVIDED "AS IS" WITHOUT ANY WARRANTY, EXPRESS OR IMPLIED, INCLUDING WITHOUT ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND ANY WARRANTY OR CONDITION OF NON-INFRINGEMENT. IBM products are warranted according to the terms and conditions of the agreements under which they are provided.

The client is responsible for ensuring compliance with laws and regulations applicable to it. IBM does not provide legal advice or represent or warrant that its services or products will ensure that the client is in compliance with any law or regulation.

Statement of Good Security Practices: IT system security involves protecting systems and information through prevention, detection and response to improper access from within and outside your enterprise. Improper access can result in information being altered, destroyed, misappropriated or misused or can result in damage to or misuse of your systems, including for use in attacks on others. No IT system or product should be considered completely secure and no single product, service or security measure can be completely effective in preventing improper use or access. IBM systems, products and services are designed to be part of a lawful, comprehensive security approach, which will necessarily involve additional operational procedures, and may require other systems, products or services to be most effective. IBM DOES NOT WARRANT THAT ANY SYSTEMS, PRODUCTS OR SERVICES ARE IMMUNE FROM, OR WILL MAKE YOUR ENTERPRISE IMMUNE FROM, THE MALICIOUS OR ILLEGAL CONDUCT OF ANY PARTY.

1. <https://www.gartner.com/doc/reprints?id=1-1YZNA09H&ct=200508&st=sb>

EWDPJZDQ

