



Migrating Oracle Exadata Workloads to AWS

INTRODUCTION

MIGRATION STRATEGIES

KEY EXADATA PRE-MIGRATION CONSIDERATIONS

MIGRATION TOOLS: CHOOSING THE RIGHT TOOL FOR THE RIGHT JOB

COMMON QUESTIONS ABOUT MIGRATING EXADATA WORKLOADS TO AWS

GET STARTED TODAY

Content

| | |
|---|-----------|
| Introduction: Modernization Opportunities for Oracle Exadata Customers | 3 |
| Migration Strategies | 4 |
| Rehost | 4 |
| Replatform | 4 |
| Refactor | 5 |
| Migration patterns when using refactor | 5 |
| Key Exadata pre-migration considerations | 6 |
| Nature of the workload (transactional vs. analytical) | 6 |
| Database editions and versions | 7 |
| Exadata feature usage | 7 |
| Performance and resource requirements | 8 |
| Additional migration considerations | 9 |
| Migration Tools: Choosing the right tool for the right job | 10 |
| Common questions about migrating Exadata workloads to AWS | 12 |
| Get started today | 13 |

Introduction:

Modernization Opportunities for Oracle Exadata Customers

The world of databases is undergoing a major transformation with the explosion of data and shift to cloud services. From an estimated \$38.6B in revenue in 2017, the DBMS market has added \$40B - doubling in 5 years, and the biggest DBMS market story continues to be the enormous impact of revenue shifting to the cloud.¹ In 2023, revenue for managed cloud database services (dbPaaS) is estimated to grow +25.1% YoY to \$63B, which represents over 47% of all DBMS revenue, according to Gartner.² With cloud services, companies are able to free IT teams from time-consuming database tasks like server provisioning, patching, and backups. As an example, AWS [fully managed database services](#) provide continuous monitoring, self-healing storage, and automated scaling to help companies focus on application development.

As the shift to cloud services continue, companies often look to break free from a monolithic software architecture and leverage microservices to reduce application complexity and increase innovation and agility. However, some companies may still use a monolithic database to serve all microservices. Imagine many microservices, each with their own data requirements and growing at their own pace, some of which are relational and some not. At the same time, all microservices are forced to use the same monolithic database engine. This approach will most likely negatively impact developers' flexibility and agility.

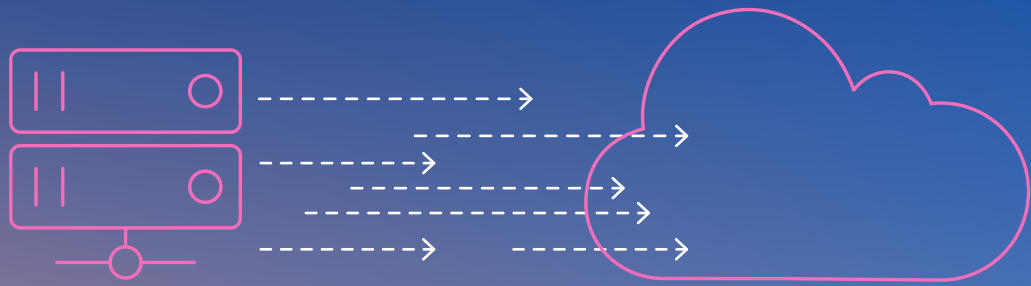
Oracle Exadata is an engineered system which consists of hardware and software components, designed to exclusively run high performance Oracle database workloads. However, running with a single database engine can introduce business agility challenges as many companies realize that the days of "one size fits all" are gone, and each workload may require a different database engine for its needs. Furthermore, Exadata introduces total cost of ownership (TCO) challenges for many companies due to dependency on Oracle for its hardware deployment, maintenance, and support. This creates lock-in challenges because it uses proprietary features that inhibit the ability to move Oracle workloads and applications onto non-Exadata systems or to other databases.

For these reasons, companies are considering a migration from Exadata to one or more AWS fully managed databases. AWS offers 7 relational and 8 purpose-built engines to support diverse data models, including relational, key-value, document, in-memory, graph, time series, wide column, and ledger databases.

AWS has helped customers, such as [California Healthcare Eligibility, Enrollment, and Retention System \(CalHEERS\)](#), [Australia Finance Group](#), and [EDF UK](#), shift their Exadata workloads to AWS. This whitepaper highlights the migration strategies, considerations, and best practices, as well as answers to common questions.

¹ <https://blogs.gartner.com/merv-adrian/2022/04/16/dbms-market-transformation-2021-the-big-picture/>

² Gartner's Forecast Public Cloud Services, Worldwide, 2020-2026, 4Q22 Update



INTRODUCTION

MIGRATION STRATEGIES

KEY EXADATA PRE-MIGRATION CONSIDERATIONS

MIGRATION TOOLS: CHOOSING THE RIGHT TOOL FOR THE RIGHT JOB

COMMON QUESTIONS ABOUT MIGRATING EXADATA WORKLOADS TO AWS

GET STARTED TODAY

Migration Strategies

It is important to understand the options for migrating Exadata workloads to AWS. This section describes three different migration strategies (rehost, replatform, refactor), as well as typical migration patterns.

Rehost

Rehost, also commonly referred to as lift-and-shift, allows companies to move an application to the cloud with minimal to no code changes. An example is migrating an on-premises Oracle database to a self-managed Oracle database on [Amazon Elastic Compute Cloud](#) (Amazon EC2). This option gives companies full control over the database and operating system. However, it requires companies to set up, configure, manage, and tune all of the components, including Amazon EC2 instances, storage volumes, and databases similar to on-premises. Rehosting is the quickest way to onboard workloads to AWS and benefit from improved operational efficiency and business agility compared to running Oracle Exadata on-premises. It is a good migration option for companies who need capabilities such as:

- Full access control over the database and its underlying AWS infrastructure (e.g., Amazon EC2, Amazon EBS).
- Specific Oracle versions not currently supported by Amazon Relational Database Service (Amazon RDS).
- Input/Output (I/O) operations per second (IOPS) or provisioned capacity greater than the [current Amazon RDS limits](#).
- Features and options that are currently [not supported by Amazon RDS](#).

Replatform

Replatform enables companies to move an application to the cloud and benefit from a fully managed database service. An example is migrating an on-premises Oracle database to [Amazon RDS for Oracle](#), which is a fully managed Oracle database that makes it easy to set up, operate, and scale Oracle deployments in the cloud. This is ideal for companies that want to spend less time managing undifferentiated database administrative tasks and more time innovating.

According to a study performed by [IDC](#), organizations realized the following benefits with Amazon RDS:

- 264% three-year ROI
- 37% more efficient DBA teams
- 42% more productive development teams
- 97% less unplanned downtime

INTRODUCTION

MIGRATION STRATEGIES

KEY EXADATA PRE-MIGRATION CONSIDERATIONS

MIGRATION TOOLS: CHOOSING THE RIGHT TOOL FOR THE RIGHT JOB

COMMON QUESTIONS ABOUT MIGRATING EXADATA WORKLOADS TO AWS

GET STARTED TODAY

Alternatively, [Amazon RDS Custom for Oracle](#) offers granular control over the database and operating system (OS) to run third party-applications that require privileged access, while at the same time providing the managed experience of Amazon RDS for Oracle, including automated backups, patching, scaling, and recovery from host failure.

Oracle Exadata customers who face challenges related to database maintenance overhead, which impacts their business agility and are required to keep using the Oracle database engine, can maximize the benefits of AWS by using a managed service like Amazon RDS for Oracle. It provides greater operational efficiency benefits compared to running Oracle on Amazon EC2 (rehost).

Refactor

Refactor involves reimagining how an application is designed and modifying its architecture to take full advantage of cloud-native features to improve agility, performance, and scalability. An example is migrating an Oracle database to [Amazon Aurora PostgreSQL-Compatible Edition](#). This option is the more complex option of the three listed, requiring additional time and planning to assess, plan, and convert source schema and code to be compatible with the target engine before the application can be moved to a different database engine. However, companies benefit from the long-term business value, including maximizing cost savings by breaking free from expensive, punitive commercial licenses and taking full advantage of the innovation capabilities (e.g., [serverless configuration](#), [Global Database](#), [Amazon DevOps Guru](#), [Amazon RDS Blue/Green Deployments](#), [machine learning integration](#)), that the cloud has to offer.

Migration patterns when choosing refactor

For Exadata workloads that will be refactored, for example to Amazon Aurora, there are two migration pattern choices:

- **One-step approach** - Companies can break free from Oracle database with full database modernization as a one-step approach by migrating from Oracle Exadata directly to Amazon Aurora. While this approach requires more planning and effort, it enables companies to maximize the cloud benefits as soon as Exadata workloads have been migrated to AWS.
- **Phased approach** - Companies can start with rehosting the Oracle workloads on Amazon EC2 or replatforming the Oracle database on Amazon RDS for Oracle, and then refactoring to Amazon Aurora in the second phase. This approach helps reduce time, effort, and risk during the initial migration to AWS and focuses on optimization and modernization in the second phase. However, it does create multiple projects and periods of change management to achieve modernization.



INTRODUCTION

MIGRATION STRATEGIES

KEY EXADATA PRE-MIGRATION CONSIDERATIONS

MIGRATION TOOLS: CHOOSING THE RIGHT TOOL FOR THE RIGHT JOB

COMMON QUESTIONS ABOUT MIGRATING EXADATA WORKLOADS TO AWS

GET STARTED TODAY

Key Exadata pre-migration considerations

Formulating a migration strategy requires an understanding of the current workload characteristics, application, and database dependencies, as well as business requirements.

Nature of the workload (transactional vs. analytical)

Understanding the nature of the workload can assist with identifying the migration target. For example, for online transaction processing (OLTP) workloads, when applications are focused on facilitating, processing, and maintaining transactional data, AWS offers Amazon RDS. For online analytical processing (OLAP) workloads, used for reporting and data analysis, Amazon RDS can also support most analytic workloads, but for more complex analytic workloads and data integrations, AWS offers [Amazon Redshift](#), a fully managed cloud data warehouse, as an alternative migration target.

The following table shows typical database characteristics that can help in determining whether the nature of the workload is transactional or analytical.

| Oracle Exadata Character | OLTP | OLAP |
|--|--|--|
| Block size | <=8KB | >8KB |
| Commit rate | high | low |
| Buffer cache hit ratio | =>99% | <99% |
| Common wait events | "db file sequential read" "log file sync" | "db file scattered read" "direct path read" |
| Average I/O request size (I/O throughput divided by IOPS) | <120KB | >120KB |
| Star schema/s usage | no | yes |
| star_transformation_enabled parameter | false | true |
| Hybrid Columnar Compression (HCC) usage | no | yes |

Database editions and versions

Oracle Database editions and versions are often driven by application support requirements, advanced feature usage, and other factors such as financial restrictions. For example, some customers prefer to keep older database versions to ensure stability of the current release before they decide to upgrade. In this section, we will discuss these important considerations.

- **Editions** - Depending on workloads and use cases, Amazon RDS can support both Oracle Enterprise Edition (EE) and Standard Edition 2 (SE2). For SE2, customers have two licensing options, Bring-Your-Own-License (BYOL) or License Included (LI). With LI, the Oracle licensing cost is included in the hourly rate of the instance. Oracle Database EE is supported only under the BYOL model.
- **Versions** - It is important to understand the current version as this could impact the decision on whether to migrate to Amazon RDS for Oracle, Amazon RDS Custom for Oracle, or Oracle on Amazon EC2. While some customers are open to upgrading as part of their cloud migration journey, it is not always possible due to a strict version requirement (e.g., a third-party application may have specific version requirements), and that could impact the decision on the target migration path. The full list of supported Oracle versions can be reviewed for: [Amazon RDS for Oracle](#) and [Amazon RDS Custom for Oracle](#).

Exadata feature usage

Exadata has proprietary software that enables features in the Oracle Database to improve query performance, lower redo log latency, compress data, and improve other database operations. As a result, alternative approaches should be considered to provide similar functionality when migrating from Exadata. Exadata unique features and alternative AWS solutions are listed below.

- **Smart scans and storage indexes** - The Smart Scan feature can offload SQL processing to the Exadata Storage Servers. Exadata Smart Scan, which only works when the database performs full table scans or index fast full scans, are mostly used for OLAP applications. On the other hand, OLTP applications rarely leverage Smart Scans because they usually retrieve a small number of rows. Storage indexes reduce the amount of storage I/O required to service requests for Exadata Smart Scan. Exadata Storage Indexes are not database objects and do not exist in data files. Therefore, they do not need to be migrated. If a SQL statement takes advantage of Exadata Storage Indexes and query performance degrades after migrating the database to AWS, indexes should be created on the table(s) in AWS.
- **Exadata Smart Flash Cache** - Exadata Smart Flash Cache intelligently determines the data that is most useful to cache based on data usage, access patterns, and hints from the database that indicate the type of data being accessed. It also avoids caching data that will never be reused or will not fit into the cache. The flash cache is stored on flash-memory cards on the storage servers. While Exadata Smart Flash Cache is not available in non-Exadata environments, Oracle has a database feature called Database Smart Flash Cache (DSFC) which extends Oracle's System Global Area (SGA) buffer cache with a secondary caching layer. While the two smart cache solutions are different, the DSFC available in AWS provides a secondary caching mechanism that can benefit read-intensive workloads that use mostly single-block random reads. As a best practice, size the cache to hold a large

portion of the active data set that does not fit in the buffer cache. This feature is supported in Amazon RDS for Oracle (for details and list of instance classes which support this feature click [here](#)) and can run on any EC2 instance with local non-volatile memory express (NVMe) storage (e.g. M5d, R5d, R6id, X2idn, and X2iedn). When using Amazon RDS for Oracle or Oracle on Amazon EC2 without the flash cache feature, queries read data from Amazon Elastic Block Store (Amazon EBS), which has a higher latency than local NVMe storage physically on the host computer.

- **Hybrid Columnar Compression (HCC)** - HCC enables high levels of data compression, which can result in cost savings and performance improvements. It is used primarily in data warehouses because it is not optimized for updates, making it less performant for most OLTP workloads. To overcome the challenge of managing larger data sets in target databases in AWS, inactive parts of data can be moved outside the primary database and stored in a cost-efficient storage solution, such as [Amazon Simple Storage Service \(Amazon S3\)](#) with Intelligent-Tiering.

For workloads that have Oracle database dependencies, HCC-enabled segments can be converted to use BASIC or ADVANCED compression features offered by the Oracle database. Oracle Advanced Compression Option (ACO) is separately licensed, yet is fully supported on RDS Oracle, RDS Custom for Oracle, and Oracle on EC2. For companies who choose to refactor their workloads to Amazon RedShift, column compression is used. By default, Amazon Redshift automatically manages compression encoding for all columns in the table. It is also possible to manually apply a compression type or encoding to the columns in a table (click [here](#) for more details).

When choosing a physical database migration to AWS, it is important to disable and stop using HCC. Oracle Support document "[Exadata: How To Disable Exadata Hybrid Columnar Compression? \(Doc ID 1080301.1\)](#)" provides the steps for disabling HCC. Disabling HCC can be done in the source Exadata environment or in the target AWS Oracle database. Typically, Exadata HCC is removed once the database is migrated to AWS as companies prefer not to disable Exadata HCC or make any modifications in their source production Exadata environment. When choosing a logical database migration to AWS with tools such as AWS Database Migration Service (AWS DMS) or Oracle GoldenGate, there is no need to disable HCC in the source Exadata environment.

Performance and resource requirements

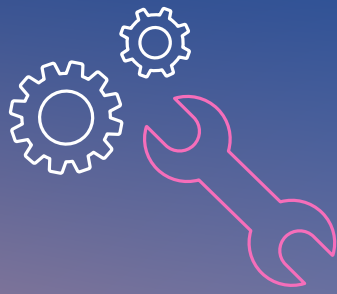
As part of migration planning, it is important to clearly understand current resource requirements (e.g., CPU, memory, IOPS, storage throughput) in order to size the target database properly to ensure performance needs are met in the most cost-efficient manner. This is particularly important when migrating from Oracle Exadata, which is often over-provisioned. There are various native Oracle tools for right-sizing, such as Automatic Workload Repository (AWR), Oracle Enterprise Manager, and Cell Control Command-Line Interface (CellCLI). In addition to native Oracle tools, AWS Database Specialists also have access to in-house tooling and best practices that help streamline the process of discovery and right-sizing for Exadata workloads.

Exadata I/O rates can sometimes be inflated because of frequent full table scans. If major Exadata Smart Scan savings were identified as part of the pre-migration, there is a SQL tuning opportunity for creating additional indexes in the target Oracle environment that could significantly reduce the storage IOPS and throughput requirements for better performance and cost-savings.

For storage IOPS and throughput, Amazon RDS offers Amazon EBS Provisioned IOPS volumes (io1), which are designed to meet the needs of I/O-intensive workloads that are sensitive to storage performance and consistency. When io1 EBS volumes are selected for Amazon RDS for Oracle, customers need to specify the allocated storage capacity and the number of provisioned IOPS. They can choose up to 64 TiB of storage, 256,000 IOPS, and a maximum storage throughput of 4 GB/s, which is more than sufficient to meet the needs of many Oracle database workloads. Workloads that require more than Amazon RDS for Oracle can support should consider other alternatives, such as Oracle on Amazon EC2 (rehost) or Amazon Aurora (refactor).

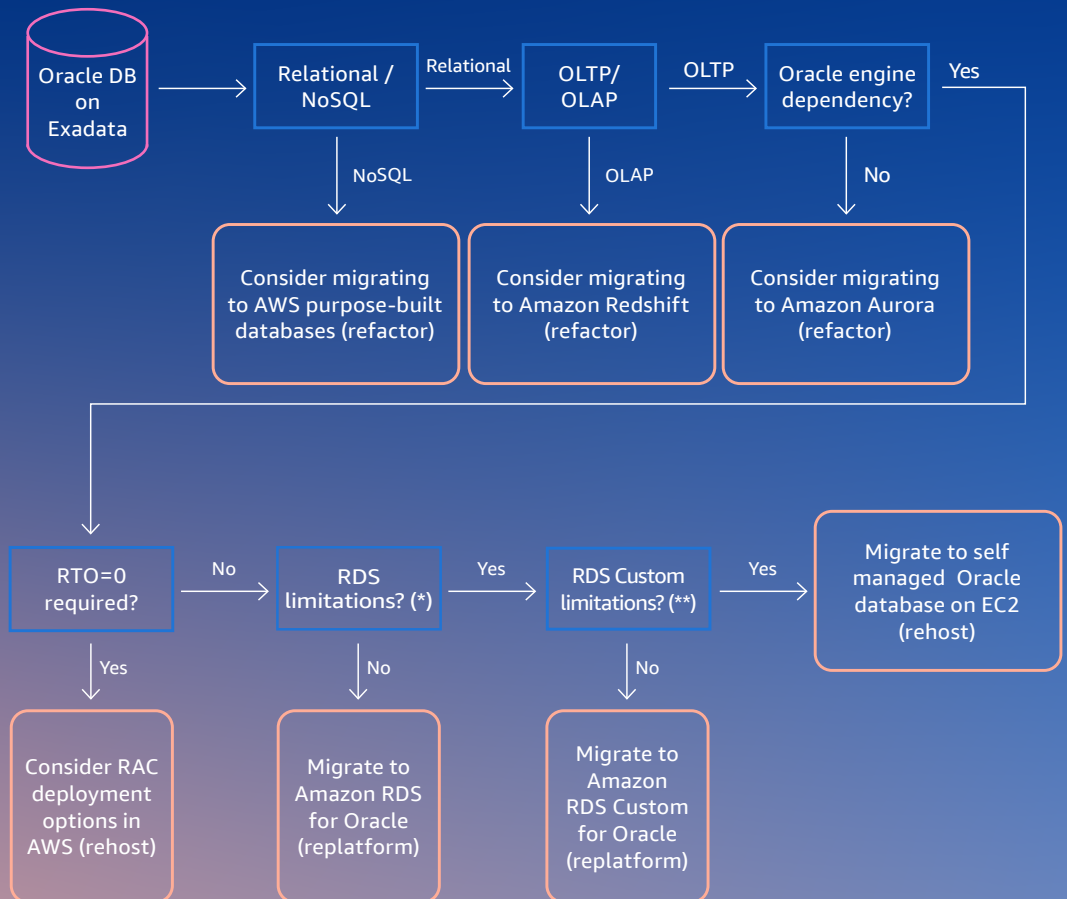
Additional migration considerations

- **Database engine dependency** - Companies should evaluate if the database supporting the application or part of the application can be ported to an open-source database, such as PostgreSQL or MySQL. When the database dependency is not deeply rooted, it is worth considering a full modernization with database services, like Amazon Aurora, to maximize benefits such as cost savings.
- **Availability requirements** - It is important to understand the maximum affordable downtime for the migration, as well as additional SLA objectives such as RTO and RPO. For example, migrations that have minimum migration downtime requirement should use tools which support change data capture (CDC) such as AWS DMS. RTO and RPO requirements can also impact the target database high availability (HA) and disaster recovery (DR) architecture, indicating solutions such as Multi-AZ deployments and read replicas.
- **Application and database dependencies** - Consolidation requirements and interdependencies between schemas and databases can also impact the migration process. Ensuring companies take this into consideration is essential to meet application integrity and reliability. For example, one option is to group dependent schemas into the same migration phase.
- **Migration structure and steps** - Companies should be assessing whether the migration and switchover to AWS will be done in one step or a sequence of steps, based on business and application requirements.
- **Static/stale data** - If part of the data is static and/or stale, it is worth considering moving it to other cost-efficient storage solutions, such as Amazon S3 with Intelligent-Tiering.
- **Network** - It is essential to understand whether the network bandwidth and latency between the source database server and AWS can meet migration needs, such as database size, transactional throughput rate, and expected migration duration. When high network performance with consistent low latency is required, [accelerated Site-to-Site VPN connection](#) or [AWS Direct Connect](#) are two potential solutions.



Migration Tools: Choosing the right tool for the right job

Using common business and technical requirements, the following diagram represents a simple decision flowchart for migrating an application running on Exadata to AWS. Given no two migration projects are the same, the diagram offers directional guidance on target databases options, and this should be complemented with a due diligence that comes from a discovery assessment or proof of concept (POC).



(*) For example: OS access, custom patching, flashback database, multiple pluggable databases, physical migration using Data Guard/RMAN

(**) For example: Oracle versions, sizing requirements that can only be supported in Oracle on Amazon EC2

- INTRODUCTION
- MIGRATION STRATEGIES
- KEY EXADATA PRE-MIGRATION CONSIDERATIONS
- MIGRATION TOOLS: CHOOSING THE RIGHT TOOL FOR THE RIGHT JOB
- COMMON QUESTIONS ABOUT MIGRATING EXADATA WORKLOADS TO AWS
- GET STARTED TODAY

Once the migration target decision has been made, companies can select the right tool to perform the migration. The chart below includes the most common Exadata to AWS migration tools and strategies:

| Migration Tools | Migration Strategy | Method | Change Data Capture |
|--|--------------------------------------|---|---------------------|
| AWS Schema Conversion Tool (SCT) | All | Migration assessment reports, procedural code conversion, schema conversion | No |
| AWS Database Migration Service (DMS) | All | Logical migration | Yes |
| Oracle GoldenGate | All | Logical migration | Yes |
| Oracle Data Pump | Rehost, Replatform | Logical migration | No |
| Oracle Recovery Manager (RMAN) | Rehost, Replatform (RDS Custom only) | Physical migration | No |
| Oracle Physical Data Guard | Rehost, Replatform (RDS Custom only) | Physical migration | Yes |

In addition, AWS offers specialized experts through [AWS Professional Services](#) and [AWS Partner Network](#) (APN) to assist in the migration process. AWS experts are tenured specialists with deep knowledge to help support companies through their data modernization journey. Here are examples of [Navisite](#) and [TCS](#) blog posts to show how AWS Partner can assist.



INTRODUCTION

MIGRATION STRATEGIES

KEY EXADATA PRE-MIGRATION CONSIDERATIONS

MIGRATION TOOLS: CHOOSING THE RIGHT TOOL FOR THE RIGHT JOB

COMMON QUESTIONS ABOUT MIGRATING EXADATA WORKLOADS TO AWS

GET STARTED TODAY

Common questions about migrating Exadata workloads to AWS

Migrating from Exadata to AWS is feasible by developing a thoughtful migration plan. In this section, we will address some common questions related to migrating Exadata workloads to AWS.

- **What are the Amazon RDS for Oracle high availability (HA) and disaster recovery (DR) options?**

Reliability is a core pillar of the AWS [Well-Architected Framework](#), and there are various HA and DR native capabilities provided in Amazon RDS to meet requirements of mission critical applications. For example, when using Amazon RDS for Oracle, companies can use Multi-AZ deployments for synchronous database replication to another AWS availability zone, providing zero data loss replication (RPO = 0). This could result in a cost advantage compared to running Oracle with Data Guard for DR as Multi-AZ does not require Oracle Enterprise Edition or Oracle Data Guard. For companies that require a cross-Region disaster recovery solution, Amazon RDS for Oracle supports managed disaster recovery and data proximity with [cross-Region read replicas](#) and database restoration to a specific point in time with [cross-Region automated backups](#). Meanwhile, for companies that require increased flexibility to create custom HA and DR solutions, RDS Custom for Oracle allows companies the ability to fine tune their HA and DR needs using 3rd party tools.

- **Is it possible to evaluate performance differences between Exadata database workloads and Oracle database workloads running on AWS?**

Yes, Oracle Real Application Testing (RAT) can be used to evaluate differences between running workloads on Exadata and AWS. AWS has published two blog posts with detailed information on the steps needed to perform this evaluation. This is also available for other Oracle database workloads (non-Exadata), which companies may consider for migration into AWS.

- [Use Oracle Real Application Testing features with Amazon RDS for Oracle](#)
- [Use Oracle Real Application Testing features with Amazon EC2](#)

- **Have companies successfully migrated Exadata workloads to AWS?**

Yes, [California Healthcare Eligibility, Enrollment, and Retention System \(CalHEERS\)](#), [Australia Finance Group](#), [EDF UK](#), and more have all successfully migrated workloads from Exadata to AWS.



Get Started Today

To maximize business agility and cost savings, companies are increasingly moving their databases to the cloud, including Exadata workloads. AWS is here to help. To start your modernization journey, it begins by contacting the appropriate [AWS account team](#) to set up a complimentary discovery session.

INTRODUCTION

MIGRATION STRATEGIES

KEY EXADATA PRE-MIGRATION CONSIDERATIONS

MIGRATION TOOLS: CHOOSING THE RIGHT TOOL FOR THE RIGHT JOB

COMMON QUESTIONS ABOUT MIGRATING EXADATA WORKLOADS TO AWS

GET STARTED TODAY



Author: Pini Dibask is a senior database specialist solutions architect at AWS, based in Israel. With 17 years in various data roles and former Oracle ACE, he is a trusted technical advisor guiding customers throughout their cloud data journey.



