Blueprint for successful migrations from Oracle Exadata to AWS

AWS Prescriptive Guidance

AWS Prescriptive Guidance: Blueprint for successful migrations from Oracle Exadata to AWS

Table of Contents

Blueprint for successful migrations from Oracle Exadata to AWS

Amazon Web Services (contributors)

July 2024 (document history)

Databases are undergoing a major transformation as a result of the explosion of data and shift to cloud services. The database management system (DBMS) market has added 40 billion USD to its 2017 revenue of 38.6 billion USD—doubling in five years—and the biggest DBMS market story continues to be the impact of revenue shifting to the cloud. According to Gartner Research, "The DBMS market grew by 14.4% in 2022, reaching $91B. Cloud dbPaaS captured nearly all the gain, with cloud spend (55.2%) exceeding on-premises (44.8%)."* Companies can use cloud services to free their IT teams from time-consuming database tasks such as 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 companies seek to maximize the benefits of moving to the cloud as part of their digital transformation, they focus on modernizing their data infrastructure. In order to meet data modernization goals, companies look to achieve the following capabilities:

- **Total cost of ownership (TCO) reduction** – A slowdown in global markets, increasing inflation, fear of a global recession, and other market conditions force companies to prioritize cost efficiency.
- **Speed and agility** – In a cloud computing environment, new IT resources are easy to deploy, which means that companies reduce the time to make those resources available to developers from weeks to just minutes. This results in a dramatic increase in agility for the organization, because the cost and time for experimentation and development are significantly lower.
- **Global scale, security, and high availability** – Companies serve customers around the globe, and therefore they often seek better ways to support their customers in different geographic regions and provide full data oversight with multiple levels of security, including network isolation and end-to-end encryption. High availability, reliability, and security are key for business-critical, enterprise workloads.
- **Performance at scale** – Companies are looking for elasticity: to start small and scale their relational or non-relational databases as their applications grow. They want to meet their storage and compute needs more easily, and preferably with no downtime.

As part of the shift to cloud services, companies often look to break free from a monolithic software architecture and use microservices to reduce application complexity and increase innovation and agility. However, some companies still use a monolithic database to serve multiple microservices. For example, microservices that have different data requirements, pace of growth, and databases (relational or non-relational) might be forced to use the same monolithic database engine. This means that developers are often required to normalize the data model to fit into a relational model instead of using a data model that supports their requirements. Therefore, using the same database engine might negatively impact developers' flexibility and agility.

An example of a monolithic approach is an architecture that uses Oracle Database on Oracle Exadata and that serves multiple workloads, multiple applications, and potentially multiple microservices. Oracle Exadata is an engineered system that consists of hardware and software components. It is designed to exclusively run Oracle Database workloads with high performance.

However, running your workloads with a single database engine can introduce business agility challenges. Many companies realize that each workload might require a different database engine for its needs. Furthermore, monolithic databases might introduce total cost of ownership (TCO) challenges for many companies because of their dependency on Oracle for hardware deployment and maintenance, in the case of Oracle databases that run on on-premises Exadata. Monolithic databases also create lock-in challenges because they use proprietary features that inhibit their ability to move Oracle workloads and applications to non-Exadata platforms or to other databases.

For these reasons, some companies consider migrating from Exadata to AWS fully managed, purpose-built databases. AWS offers many relational and purpose-built database types to support diverse data models, including relational, key-value, document, in-memory, graph, time series, and wide-column databases. AWS consultants have helped customers such as California Healthcare Eligibility, Enrollment, and Retention System (CalHEERS), Australia Finance Group (AFG), and EDF UK to migrate their Exadata workloads to AWS.

As companies consider migrating workloads from Oracle Exadata to AWS, they need to have an effective migration strategy that is aligned with their applications and business needs and clear guidance to ensure a smooth migration. The blueprint for a successful Oracle Exadata to AWS migration is a multi-step, systematic approach that includes pre-migration discovery and performance assessments, data migration, and post-migration routines for optimal performance and costs.

The purpose of this guide is to share insights, best practices, and tips on how to plan, perform, and maintain a successful migration from Oracle Exadata to AWS. It is intended to assist technical

audiences, including DBAs, IT architects, DevOps engineers, CTOs, and others in their migration journey from Oracle Exadata to AWS.

In this guide:

* Market Share: Database Management Systems, Worldwide, 2022 (Gartner Research, May 17, 2023)

Key database trends

This section discusses key database trends at the time of publication. This information helps clarify the motivations that drive database workloads into the cloud. The section covers the following topics:

- Database trends in the enterprise market
- Differences between purpose-built databases and converged databases

Database trends in the enterprise market

The database market is currently undergoing significant changes. Data volumes are growing exponentially. The total amount of data captured, copied, and consumed globally per year is increasing. Customers must derive more value from their data. Cloud companies such as AWS offer a variety of database technologies that are purpose-built for database needs. These services offer agility, innovation, less maintenance overhead, and more control, and are more cost-effective. Modern data strategies can support present and future use cases, including the steps to build an end-to-end data solution to store, access, analyze, visualize, and predict future outcomes. For more information about data services and solutions from AWS, see the AWS for Data website.

Commercial relational databases became mainstream over 40 years ago. Back then, hardware capacity was limited and costly. Storage costs were very high, and data was normalized to avoid storing duplicates. Now, most storage is cheaper than compute and memory. The requirements have changed too, and you might need microsecond performance on different datasets that include both structured and unstructured data. For years, customers have been limited to using a small set of database platforms. Commercial off-the-shelf (COTS) applications such as Oracle E-Business Suite, Siebel CRM, and Peoplesoft were able to run only on Oracle. Companies developed in-house applications by using proprietary features such as PL/SQL or Pro*C, and these custom applications satisfied business demands. However, over time, the proprietary features have become complex and harder to maintain. IT budget constraints forced many companies to rethink their approach to satisfy business demands and focus on optimizing their cost structures by migrating to less expensive options, where their migration costs were determined by the level of customization required.

As an alternative to commercial database products, AWS has introduced a portfolio of fully managed, relational, open source databases as well as purpose-built, non-relational database

engines for workload optimization of specific use cases. The main advantage of open source databases is their lower cost. IT budgets are unencumbered by contractual payments, because they no longer have to pay the license fees that are associated with commercial software. With these savings, IT departments have enormous flexibility, so they can experiment and be agile. For example, many customers modernize their Oracle workloads by moving to PostgreSQL. PostgreSQL functionality has improved significantly over the last 10 years and now includes many enterprise database features to support large, critical workloads.

The way databases have been operating is also undergoing change. For the last 30 years, customers have operated their own data centers on premises: they bought and managed infrastructure, maintained hardware, licensed networking and commercial databases, and employed IT professionals to run the data centers. The database administrators (DBAs) configured and operated primarily relational databases. Their operational tasks included hardware and software installation, sorting out licensing issues, configuration, patching, and database backup. DBAs also managed performance tuning, configuration for high availability, security, and compliance issues. Managing databases included tedious repetitive tasks and was time-consuming and expensive. Customers spent time managing infrastructure instead of focusing on core business competencies. For this reason, companies invested in automation of DBA and operational tasks where possible to better utilize DBA resources, so they could spend more time on innovation. For more information, see the IDC report Amazon Relational Database Service Delivers Enhanced Database Performance at Lower Total Cost.

Purpose-built versus converged databases

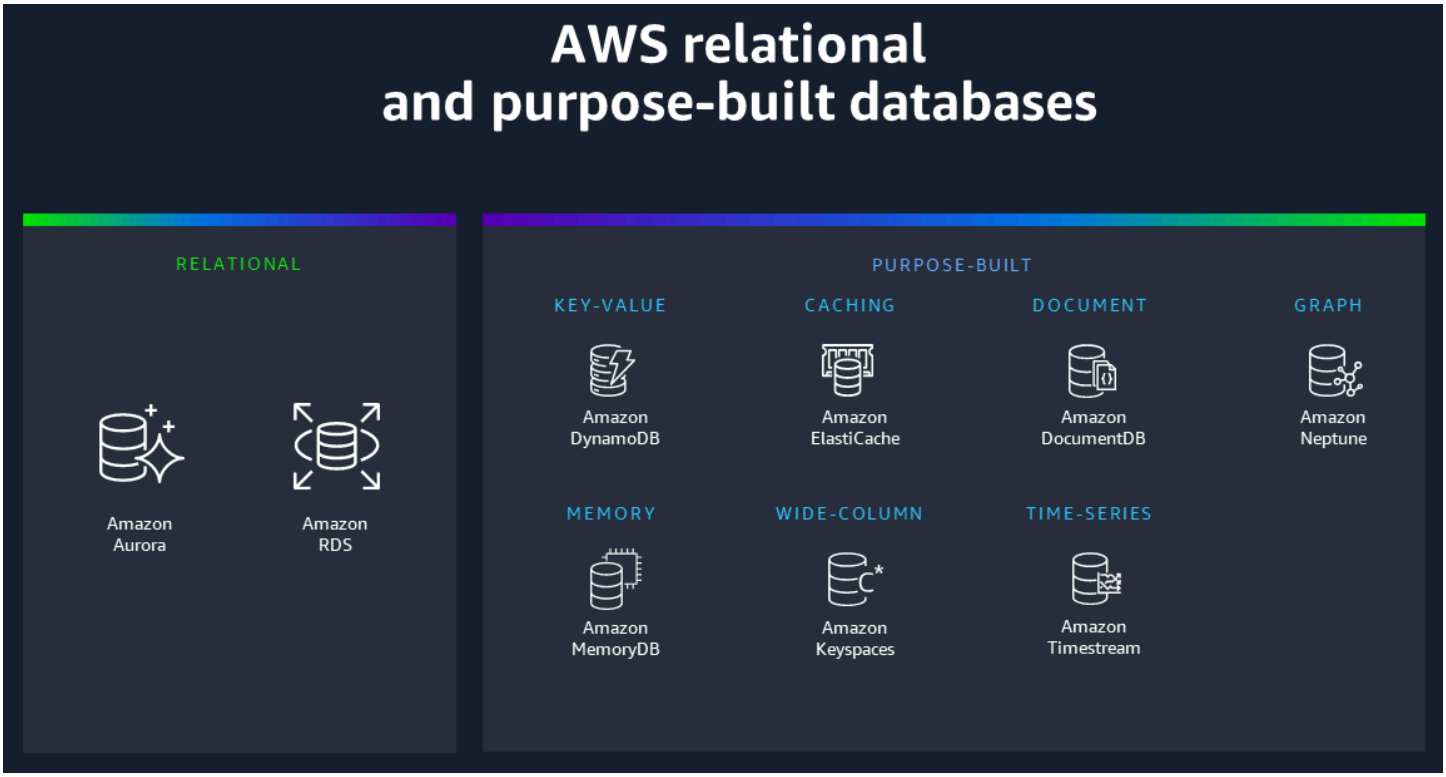
Oracle Exadata was initially released in 2008. It was designed to address a common bottleneck with large databases: moving large volumes of data from disk storage to database servers. Addressing this issue could be particularly beneficial for data warehouse applications where scanning large datasets is common. Exadata increased the pipe between the storage and database tier by using InfiniBand, and reduced the amount of data that would be transferred from disk to the database tier by using software features such as Exadata Smart Scan. In some cases, Exadata introduced performance improvements, but this came at the cost of increased total cost of ownership (TCO) and reduced agility, for the reasons mentioned in the previous section.

There are two approaches for hosting database applications:

- Using specific, purpose-built databases for specific workloads or use cases
- Using a converged database that supports different database workloads in the same database

After customers migrate to the cloud, they often want to modernize their application architectures by using microservices, containers, and serverless architectures. These modern applications have unique functionality, performance, and scalability demands, which require specific database types to support each workload.

AWS offers high-performance relational databases at a much lower cost compared with enterprise-grade, commercial databases, and eight purpose-built databases. Each purpose-built database is uniquely designed to provide optimal performance for a specific use case, so companies don't have to compromise, as often happens when using the converged database approach. The following diagram illustrates AWS database offerings.



Database type	Use cases	AWS service
Relational	Traditional applications, enterprise resource planning, customer relationship management	Amazon Aurora, Amazon RDS, Amazon Redshift

Database type	Use cases	AWS service
Key-value	High-traffic web applications, ecommerce systems, gaming applications	Amazon DynamoDB
In-memory	Caching, session managemen t, gaming leader boards, geospatial applications	Amazon ElastiCache, Amazon MemoryDB
Document	Content management, catalogs, user profiles	Amazon DocumentDB (with MongoDB compatibility)
Wide-column	High-scale industrial applicati ons for equipment maintenan ce, fleet management, and route optimization	Amazon Keyspaces (for Apache Cassandra)
Graph	Fraud detection, social networking, recommendation engines	Amazon Neptune
Time series	Internet of Things (IoT) applications, DevOps, industrial telemetry	Amazon Timestream

Database migration strategies

This section discusses the strategies for migrating Exadata workloads to the AWS Cloud. Planning a comprehensive database migration strategy is key to a successful Exadata migration. The section covers the following topics:

- Database migration dependencies before migration
- Database migration paths

Database migration dependencies before migration

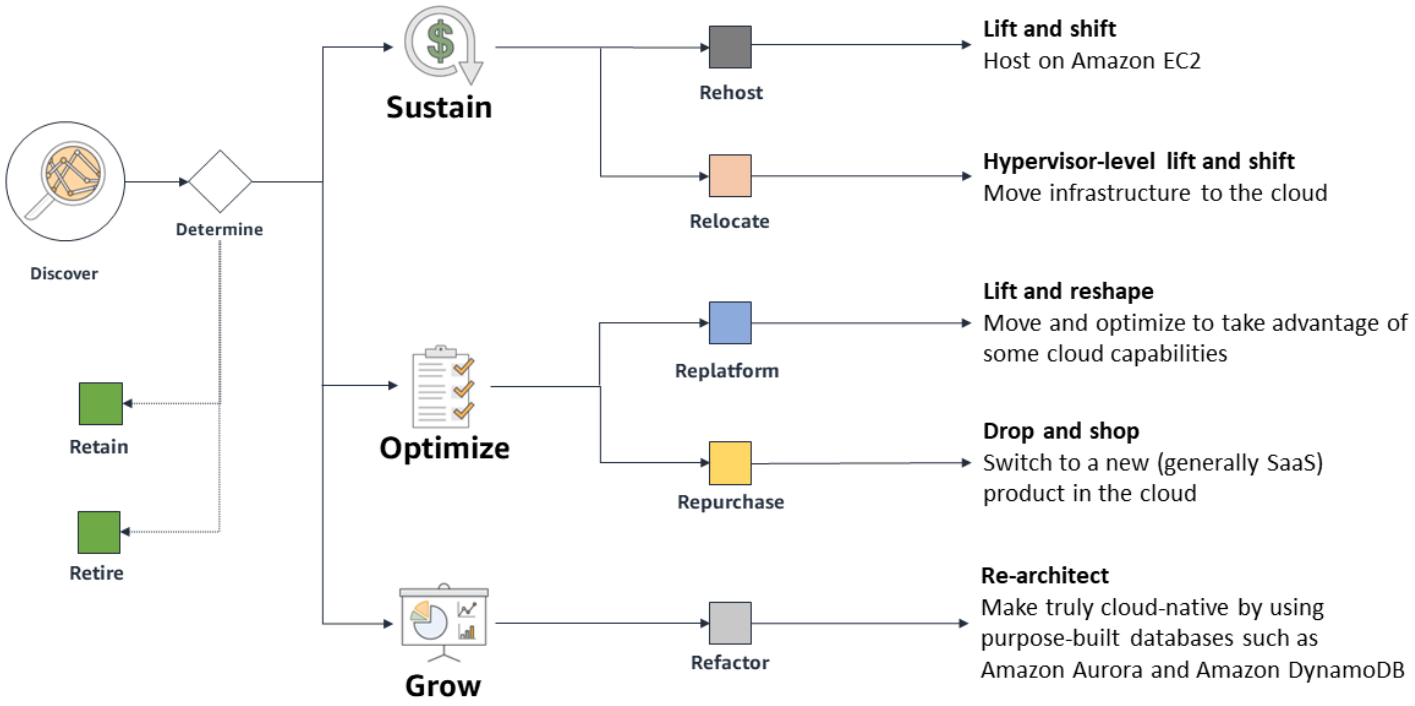
Formulating a migration strategy requires an understanding of key dependencies and the future operation of the workload on AWS. Before you choose a migration approach, we recommend that you collect and analyze the following information:

- Understand the source Exadata system.
  - The version, edition, and size of the Exadata hardware appliance
  - The database options and versions, tools, and utilities that are available
  - The size and number of the databases to be migrated
  - The Oracle licensing position
- Understand application and database dependencies.
  - Which applications use the database? Is the database part of an integrated application where multiple databases are connected?
  - Are there on-premises dependencies for moving the database?
- Understand the business requirements around the migration window.
  - How much time is available for migration?
  - What is the network connectivity between the source server and AWS?
  - What is the long-term business outlook for the database and application?
  - Will the migration and switchover to AWS be completed in one step or a sequence of steps over time?
- Understand the level of database modernization possible, given application requirements.
  - Does the workload have to stay on Oracle?
  - Can the source database be modernized? If so, to what level?

  - Which AWS database services can host the Oracle workload?
- Understand the business and performance requirements after the Exadata workload is migrated to AWS.

Database migration paths

Migration paths and choices are known as the *7 Rs* and illustrated in the following diagram.



These paths are:

- **Rehost** (lift and shift) – Move an application to the cloud without making any changes. For example, migrate your on-premises Oracle database to Oracle on an Amazon Elastic Compute Cloud (Amazon EC2) instance in the AWS Cloud.
- **Relocate** (hypervisor-level lift and shift) – Move infrastructure to the cloud without purchasing new hardware, rewriting applications, or modifying existing operations. You migrate servers from an on-premises platform to a cloud service for the same platform. For example, migrate a Microsoft Hyper-V application to AWS.

- **Replatform** (lift and reshape) – Move an application to the cloud and introduce some level of optimization to take advantage of cloud capabilities. For example, migrate on-premises Oracle databases to Amazon RDS for Oracle in the AWS Cloud.
- **Repurchase** (drop and shop) – Change to a different product, typically by moving from a traditional application to a software as a service (SaaS) product, and migrate data from your on-premises application to the new product. For example, migrate customer data from an on-premises customer relationship management (CRM) system to Salesforce.com.
- **Refactor** (re-architect) – Move an application and modify its architecture by taking full advantage of cloud-native features to improve agility, performance, and scalability. For example, migrate using one of the AWS Prescriptive Guidance migration strategies for relational databases. A refactoring strategy can also include rewriting the application to use the purpose-built databases that AWS offers for different workloads. Or, choose to modernize monolithic applications by breaking them down into smaller microservices.
- **Retain** (revisit) – Keep applications in the source environment. These might include applications that require major refactoring, where you might want to postpone the work until a later time. Or you might have a legacy application that you want to retain because there's no business justification for migrating it.
- **Retire** – Decommission or remove applications that are no longer needed in the source environment.

Typically, with Exadata stacks, rehost and replatform are the primary migration paths. The rehosting approach is used when the Exadata workload is complex or uses a commercial off-the-shelf (COTS) application. Refactoring is too time-consuming and resource-intensive to implement in a single step if the goal is database modernization (for example, replacing the Oracle Exadata database with Amazon Aurora PostgreSQL-Compatible Edition). You might consider a two-step approach instead: First, rehost the Oracle database on Amazon EC2 or replatform the database on Amazon RDS for Oracle. You can then refactor the database to Aurora PostgreSQL-Compatible. This approach helps reduce costs, resources, and risks during the first phase and focuses on optimization and modernization in the second phase.

There are four AWS database offerings that support rehost or replatform migrations:

- **Amazon Relational Database Service (Amazon RDS)** and **Amazon Aurora** are fully managed services that make it simple to set up, operate, and scale databases in the cloud. Currently, they support eight database engines: Amazon Aurora with MySQL compatibility, Amazon Aurora with

PostgreSQL compatibility, and Amazon RDS for Db2, MySQL, MariaDB, PostgreSQL, Oracle, and SQL Server.

- **Amazon EC2** supports a self-managed Oracle database. It provides full control over the infrastructure and the setup of the database environment. Running your database on Amazon EC2 is very similar to running your database on a dedicated server. You have full control of the database and operating system-level access with a choice of tools to manage the operating system, database software, patches, data replication, backup, and restoration. This migration option requires setting up, configuring, managing, and tuning all the components as you would on premises. It includes configuration of EC2 instances, storage volumes, scalability, networking, and security.
- **Amazon RDS Custom for Oracle** supports the customization of the underlying operating system and database environment. It gives you more control than Amazon RDS, but also more responsibility for tasks such as operating system patching. You also need to ensure that your customizations don't interfere with AWS automation, which is a core part of our shared responsibility model with Amazon RDS Custom.

Customers often migrate their workloads to Amazon RDS or Amazon EC2 (for a self-managed Oracle database). For Amazon RDS, AWS manages the operating system and provides limited permissions on the database layer. When you create an Amazon RDS database, AWS provides a database endpoint through which you can connect to the database instance. Amazon RDS Custom gives you full access to the underlying database, the operating system, and all resources. Some database activities are shared between you and AWS automation. If you rehost your Oracle database on an EC2 instance, you manage your database, operating system, and resources as you would when you run your Oracle database on premises. Therefore, if you have a workload that can't move to Amazon RDS, consider migrating your Oracle database to Amazon RDS Custom or Amazon EC2. For additional guidance, see Choosing an AWS database service in the *AWS Getting Started Resource Center*. The later sections of this guide discuss these options in more detail.

Migration considerations

There are many tools and techniques to migrate your Exadata workload to AWS. These fall into two main categories: *physical migration* and *logical migration*. Physical migration refers to lifting the database block by block from one server to another. Logical migration involves extracting the data from one database and loading it into another.

You can also choose an *online* or *offline* migration method based whether your workload can tolerate minimal (zero or near zero) or longer downtime.

Online migration

This method is used when the application requires near zero to minimal downtime. Typically, large and critical databases use this method. In an online migration, the source database is migrated in multiple steps to AWS. In the initial steps, the data in the source database is copied to the target database while the source database is still running. In subsequent steps, all changes from the source database are propagated to the target database. When the source and target databases are in sync, they are ready for cutover. During cutover, the application switches its connections over to the target database on AWS.

An online migration from your Oracle database to Amazon RDS for Oracle usually involves Oracle Data Pump for the initial steps (full load). Inflight transactions are then handled by using a logical replication tool such as AWS Database Migration Service (AWS DMS) or Oracle GoldenGate. If you're using this method to migrate to Amazon EC2, you can handle both full load and inflight transactions by using Oracle Data Guard or Oracle Recovery Manager (RMAN). You can also use logical tools such as AWS DMS and Oracle GoldenGate. The Performing the migration section describes these tools in more detail.

Offline migration

You can use the offline migration method if your application can afford planned downtime. Typically, small, less critical databases use this method. With this type of migration, a logical replication tool isn't typically necessary. For offline migration to Amazon RDS for Oracle, you can use Oracle Data Pump. For offline migration to Amazon EC2, you can use Oracle RMAN or Data Pump. The Performing the migration section discusses these tools in more detail.

Additional considerations

Another consideration is whether to move all data to the new environment or archive data before the migration takes place. Also, consolidation of schemas might be required. If the migration involves multiple terabytes, using a physical device to copy the data and then transporting it is quicker than copying the data across the network. Later sections of this guide expand on these techniques.

Discovery phase

Exadata is an engineered system that's optimized for running different types of Oracle database workloads and is widely used as a consolidation platform for Oracle databases. These workloads include online transaction processing (OLTP) and online analytical processing (OLAP) workloads, high transaction-intensive, business-critical applications, and non-critical workloads that don't need the capabilities of an engineered system such as Exadata. One of the key phases in a successful Exadata workload migration is the discovery phase. In this phase, you analyze the source Exadata platform to assess critical details such as how application and business units use their Exadata systems to meet their performance and availability requirements and the benefits from Exadata-specific features. The information that you gather during the discovery phase is critical to understanding your workload requirements and choosing the right platform on AWS to meet the performance and availability requirements of your applications.

This section discusses how you can assess your source Exadata platform to gather key information such as workload characteristics, Exadata feature dependencies, and other considerations. The section also covers how to choose the right platform to host the workload on AWS and how to right-size the target instance by using the information you gathered.

For a questionnaire that you can use as a starting point to gather information for the discovery phase of your migration project, see the appendix in the AWS Prescriptive Guidance guide *Migrating Oracle databases to the AWS Cloud*.

In this section:

- Decision flowchart

Workload characteristics

Historically, specialized database computing platforms were designed for a particular workload, such as online transaction processing (OLTP) or online analytical processing (OLAP), and those specific design patterns made it a poor choice for other workloads. For example, Oracle databases that host decision support systems typically use a larger block size to support reading more data from the cache with fewer I/O operations. On the other hand, OLTP workloads benefit from a smaller block size to favor random access to small rows and to reduce block contention. Exadata is effective at running any type of Oracle database workload or any combination of workloads because of features such as persistent memory (PMEM) and Exadata Smart Flash Cache to boost the performance of OLTP transactions, and Hybrid Columnar Compression (HCC) and Smart Scan to favor analytical queries. However, migrating an Exadata workload gives you a good opportunity to consider using a purpose-built database engine for the workload instead of using your existing database type or instance. AWS purpose-built databases make it easy to select a specific type of service for a specific workload on a consumption-based model instead of trying to force multiple workloads onto the same platform. As discussed earlier, AWS offers over 15 purpose-built engines to support diverse data models, including relational, key-value, document, in-memory, graph, time series, and wide-column databases.

Traditionally, databases that are optimized for decision support systems follow specific design patterns and workload characteristics such as the following:

- Larger database block size (16K or 32K)
- Star schemas with fact and dimension tables and the `star_transformation_enabled` parameter set to `TRUE`
- Compression features such as HCC, Advanced Compression, or Basic Compression
- OLAP feature
- Presence of materialized views in the database with `query_rewrite_enabled` set to `TRUE`
- Massive parallel processing
- Heavy I/O footprint

On the other hand, databases that are optimized for OLTP have smaller database block sizes (8K or smaller), single block reads, heavy concurrency, high buffer cache hit ratio, and serial execution of

transactions. In Exadata, it is typical to see anti-patterns where a database designed for an OLTP workload is heavily used for analytical queries, or the other way around. It is highly unlikely for an Oracle database to be used for pure OLTP workloads, because it is a common practice to run reporting queries on the transactional database for convenience.

Various system statistics available in Oracle dynamic performance views, the Automatic Workload Repository (AWR) report, and the Statspack report can reveal how similar a database workload is to an OLTP or OLAP system. The statistic `Physical read total multi block requests` indicates the total number of read requests that were read in two or more database blocks per request. The difference between `Physical read total IO requests` and `Physical read total multi block requests` indicates the total number of single block read requests that were issued by the database. A high number of multi-block requests typically indicate an OLAP system, and a high number of single-block read requests indicate an OLTP system. Furthermore, the following statistics in the AWR report can also reveal whether a workload that's running on an Oracle database is primarily an OLTP or OLAP workload:

- `user commits` – Reflects the number of commits issued at the boundary of a transaction. Typically, OLTP systems have a high number of small transactions, which result in a high number of user commits. On the other hand, OLAP systems run a smaller number of heavy transactions.
- `Buffer hit` – Indicates how often a requested block is found in the buffer cache without requiring disk access. OLTP systems typically have a buffer hit ratio above 99 percent, whereas the buffer hit ratio for OLAP systems is typically low.

The following table summarizes the common differences in workload characteristics between OLTP and OLAP systems.

Characteristic	OLTP	OLAP
Block size	<= 8K	> 8K
Commit rate	High	Low
Buffer cache hit ratio	> 99%	< 99%
Prominent I/O wait events	DB file sequential read, log file sync	DB file scattered read, direct path read

Characteristic	OLTP	OLAP
Average I/O request size (I/O throughput / IOPS)	< 120K	> 400K
Star schema	Does not exist	Might exist
`star_transformation_enabled` **parameter**	FALSE	TRUE
Parallelism	Low degree or disabled	Enabled with high degree

If your database primarily supports an OLAP workload, a purpose-built data warehouse solution such as Amazon Redshift might be a better fit when you migrate your workload to AWS. You can then build an analytical solution on AWS by using services such as Amazon S3, Amazon Athena, and Amazon QuickSight. For OLTP workloads, Amazon RDS comes with a choice of six relational engines, including Amazon RDS for Oracle, if you have a dependency on an Oracle database. If you don't, you can choose an open source engine such as Amazon RDS for PostgreSQL or Aurora PostgreSQL-Compatible. Amazon DynamoDB can also host highly scalable transactional systems that do not require a relational model and could be served by a key-value store.

Read/write ratio

Another important factor is the read/write ratio of the workload hosted on the database that you want to migrate. Most OLTP systems are also used for reporting purposes, and ad-hoc, resource-intensive queries are run against critical transactional databases. This often causes performance issues in critical application components. Those less critical, resource-intensive reporting queries can be redirected to a copy of the production instance to avoid any performance impact to the critical production application. The AWR `physical writes` statistic reflects the total number of data blocks written to disk, and the `physical reads` statistic specifies the total number of data blocks read from disk. Using these statistics, you can determine the read percentage of the workload as follows:

```
Read percentage = physical reads/(physical reads + physical writes)*100
```

Depending on how a transaction issues read operations on the database, you can deploy a read replica solution or a caching solution that's external to the database—for example, Amazon ElastiCache—in the target architecture. This helps reduce the resources that the primary database

instance requires to serve the read workload. Amazon Aurora, which is a cloud-native relational database engine that's part of the Amazon RDS family, provides an automatic scaling option that supports a highly scalable, read-only workload with up to 15 read instances. You can also use Aurora global databases to span multiple AWS Regions with fast local read operations and low latency in each Region.

Non-relational workloads

Oracle Database version 12.c supports the storage of JSON data natively with relational database features. In 21c, Oracle Database introduced the JSON data type. Additionally, the Simple Oracle Document Access (SODA) feature lets you create, store, and retrieve collections of documents by using NoSQL APIs. You can also use Oracle Graph Server for graph workloads. However, you can run those non-relational workloads most efficiently when you use AWS purpose-built databases such as Amazon DynamoDB, Amazon DocumentDB, or Amazon Neptune. These services are specifically optimized for NoSQL access patterns and specialized use cases.

Database engine dependencies

Many customers consider migrating their workloads from Oracle Database to Amazon Aurora PostgreSQL-Compatible. This AWS service provides a cloud-native, relational, cost-effective database with enterprise-class features, enhanced performance, and security with no licensing cost. Heterogeneous migration from Oracle Database to PostgreSQL has become much easier with AWS Database Migration Service (AWS DMS) and AWS Schema Conversion Tool (AWS SCT). AWS SCT makes heterogeneous database migrations predictable. It automatically converts the majority of schema and code objects to the target platform, and also predicts the effort required to convert objects manually when automatic conversion is not an option.

Heterogeneous migration might not be feasible in all migration scenarios. For example, workloads that involve Oracle packaged applications such as Oracle E-Business Suite (Oracle EBS) cannot be easily migrated to PostgreSQL or other database engines. Similarly, modernizing applications that depend on specific features of Oracle database, such as Java Virtual Machine (JVM) or Advanced Compression, might require more time, effort, and resources. During the discovery phase, you should analyze any dependencies that your application might have on Oracle Database and its features. Consider the feasibility of migrating your workload to an open source engine based on factors such as migration complexity, effort required, cost benefits, and skill sets.

Database editions and versions

If your Exadata workload can be hosted on an Oracle Database on AWS, there are multiple options to choose from, including Amazon RDS for Oracle, Amazon RDS Custom for Oracle, self-managed instances on Amazon EC2, and Oracle Real Application Cluster (RAC) deployment options on AWS. You should evaluate any dependencies your application might have on a specific edition or version of Oracle Database. If your application depends on a legacy version of Oracle Database, you might encounter challenges when you try to deploy that version in Amazon RDS for Oracle, which enforces the Oracle support lifecycle. On the other hand, Amazon RDS Custom for Oracle uses Bring Your Own Media (BYOM) and Bring Your Own License (BYOL) policies that currently enable you to deploy legacy versions of Oracle Database such as 12.1, 12.2, and 18c.

You might consider migrating from Oracle Database Enterprise Edition (EE) to Standard Edition 2 (SE2) to reduce the license cost. Understanding feature dependencies and advanced planning of mitigation strategies are key to successful migrations from Oracle Database EE to SE2. Amazon RDS for Oracle provides two licensing options: License Included (LI) and BYOL. If you use the LI option for Oracle Database SE2, you don't have to purchase your Oracle Database licenses separately. You can run Oracle Database SE2 with an LI license on AWS without a support contract with Oracle and no annual support fees. LI pricing is inclusive of software, underlying hardware resources, and Amazon RDS management capabilities. By using On-Demand Instances for the LI model, you can pay for the DB instance by the hour with no long-term commitments.

AWS SCT can analyze your workload's current use of Oracle Database EE-specific features. **The License Evaluation and Cloud Support** section of the AWS SCT report provides detailed information about Oracle features in use to ensure that you can make informed decisions when you migrate to Amazon RDS for Oracle.

If your workload uses Oracle Database EE features and options such as Oracle Data Guard for high availability or Automatic Workload Repository (AWR), which are licensed under the Oracle Diagnostics Pack, to diagnose performance issues, it might still be possible to move to Oracle Database SE2 on AWS. The Amazon RDS Multi-AZ option provides high availability and helps prevent data loss. This feature uses storage replication without depending on Oracle Data Guard and is available for both Oracle Database EE and SE2. Similarly, you can meet your performance monitoring requirements without Oracle Diagnostics Pack, by using Oracle Statspack with AWS monitoring capabilities such as Performance Insights, Amazon CloudWatch metrics, and Enhanced Monitoring.

The blog post Rethink Oracle Standard Edition Two on Amazon RDS for Oracle discusses various tactics for mitigating feature gaps in Amazon RDS for Oracle when you use Oracle Database SE2. We also recommend that you review the AWS Prescriptive Guidance publications Evaluate downgrading Oracle databases to Standard Edition 2 on AWS and Replatform Oracle Database Enterprise Edition to Standard Edition 2 on Amazon RDS for Oracle.

Consolidation of databases

Exadata is considered a convenient platform for consolidating Oracle databases when the primary objective is to reduce the cost of the database environment with higher utilization of infrastructure resources. The consolidation of databases in Exadata helps justify the high cost of Exadata when a single database workload cannot fully utilize all the resources and capabilities of an Exadata system. Consolidation also helps to improve operational efficiency and standardization.

Common strategies on Exadata platforms include the consolidation of:

- Multiple databases that are a part of a single Real Application Cluster (RAC) in an Exadata system
- Multiple databases that are deployed under different RACs or a combination of RAC and single-instance databases
- Multiple pluggable databases (PDBs) in a container database
- Multiple schemas in a single database

These consolidation strategies often make it difficult to meet the different levels of security, scalability, performance, and SLA requirements associated with the workloads that are consolidated in Exadata.

On AWS, you can scale resources easily and adopt a cost-effective deployment model without consolidating your databases. However, you might still want to consolidate your databases and schemas in the target AWS environment for various reasons, including complex interdependencies between schemas or low-latency database links between multiple databases.

Considerations for consolidating your Oracle databases on AWS:

- You can implement a schema consolidation strategy with any Oracle deployment model on AWS.
- Amazon RDS for Oracle and Amazon RDS Custom for Oracle support multitenant architectures with multiple pluggable databases inside a container database.

Exadata feature usage

This section discusses key Exadata features that you should take into consideration when you migrate your Exadata workloads. These features include Smart Scan, Hybrid Columnar Compression (HCC), storage indexes, and Persistent Memory (PMEM). This section also discusses ways to assess the Exadata feature dependency of your workloads, how to measure the degree of Exadata feature usage, and strategies for meeting your application requirements in the target platform without using Exadata-specific features.

> **Note**
>
> Oracle enhances Exadata by introducing new hardware and software features on a regular basis. It is beyond the scope of this guide to cover all those features.

In this section:

Smart Scan

Exadata uses its database-aware storage subsystem to offload processing from database servers by moving some of the SQL processing to the storage cell servers. Exadata Smart Scan can reduce the volume of data that's returned to the database servers through offloaded filtration and column projection. This feature solves two of the main challenges in dealing with large datasets: the transference of huge and unnecessary data from the storage layer to database servers, and the time and resources spent filtering required data. Smart Scan is an important capability of Cell Offload Processing, which also includes datafile initialization, HCC decompression, and other functionality.

The flow of data from Smart Scan can't be buffered in the system global area (SGA) buffer pool. Smart Scan requires a direct path read, which is buffered in the program global area (PGA). A SQL statement must meet a few requirements to work with Smart Scan:

- The segment queried by the SQL statement must be stored in an Exadata system where the ASM disk group setting `cell.smart_scan_capable` attribute is set to `TRUE`.
- A full table scan or an index fast full scan operation must occur.
- The segment involved in the SQL statement must be big enough to undergo a direct path read operation.

To assess the efficiency of Smart Scan in an Exadata system, you should consider the following key database statistics:

- `physical read total bytes` – The total amount of I/O bytes for read operations issued by the database, regardless of whether the operation was offloaded to the storage servers. This indicates total read operations, in bytes, issued by database servers to Exadata storage cells. This value reflects the read I/O capacity that the target platform on AWS has to meet when you migrate the workload to AWS without tuning it.
- `cell physical IO bytes eligible for predicate offload` – The amount of read operations, in bytes, that are input to Smart Scan and are eligible for predicate offload.
- `cell physical IO interconnect bytes` – The number of I/O bytes that are exchanged over the interconnect between the database server and the storage cells. This covers all types of I/O traffic between database and storage nodes, including bytes returned by Smart Scan, bytes returned by queries that aren't eligible for Smart Scan, and write operations.
- `cell physical IO interconnect bytes returned by smart scan` – I/O bytes returned by the cell for Smart Scan operations. This is the output of Smart Scan.
- `cell physical IO bytes eligible for predicate offload` – You can compare this value with physical read total bytes to understand how many total read operations are subject to Smart Scan. The ratio of `cell physical IO bytes eligible for predicate offload` (input for Smart Scan) to `cell physical IO interconnect bytes returned by smart scan` (output of Smart Scan) indicates the efficiency of Smart Scan. For an Exadata system that includes mostly read operations, the ratio of `cell physical IO interconnect bytes returned by smart scan` to `cell physical IO interconnect bytes` can indicate the dependency on Smart Scan. However, this might not always be the case, because `cell`

`physical IO interconnect bytes` also includes double the number of write operations (with ASM mirroring) between the compute and storage servers.

You can get these database I/O statistics and Exadata-specific metrics from the AWR report or by directly querying the underlying V$ views such as `V$SYSSTAT`, `V$ACTIVE_SESSION_HISTORY`, and `V$SQL`.

In the following example from an AWR report collected from an Exadata system, the database requested 5.7 Gbps of read throughput, 5.4 Gbps of which was eligible for Smart Scan. Smart Scan output contributed to 55 MBps out of 395 MBps of total interconnect traffic between database and compute nodes. These statistics point to an Exadata system that has a high dependency on Smart Scan.

Statistic	Total	per Second
physical read total bytes	41,486,341,567,488	5,758,375,137.90
cell physical IO bytes eligible for predicate offload	39,217,360,822,272	5,443,436,754.68
cell physical IO interconnect bytes	2,846,913,082,080	395,156,370.37
cell physical IO interconnect bytes returned by smart scan	400,725,918,720	55,621,456.14

You can assess Smart Scan efficiency and dependencies at the SQL level by using the following columns of the `V$SQL` view.

- `IO_CELL_OFFLOAD_ELIGIBLE_BYTES` – Number of I/O bytes that can be filtered by the Exadata storage system.
- `IO_INTERCONNECT_BYTES` – Number of I/O bytes exchanged between the Oracle database and the storage system.
- `PHYSICAL_READ_BYTES` – Number of bytes read from disks by the monitored SQL.

The following query output shows Smart Scan benefits for a SQL query that has the SQL ID `xn2fg7abff2d`.

```
select
  ROUND(physical_read_bytes/1048576) phyrd_mb
  , ROUND(io_cell_offload_eligible_bytes/1048576) elig_mb
  , ROUND(io_interconnect_bytes/1048576) ret_mb
  , (1-(io_interconnect_bytes/NULLIF(physical_read_bytes,0)))*100 "SAVING%"
  from v$sql
  where sql_id = 'xn2fg7abff2d' and child_number = 1;
```

```
 PHYRD_MB      ELIG_MB     RET_MB  SAVING%
----------     ---------- ---------- ----------
 10815             10815   3328     69.2%
```

To test the influence of Smart Scan on the workload, you can disable the feature by setting the `cell_offload_processing` parameter to `FALSE` at the system, session, or query level. For example, to disable Exadata Storage Server cell offload processing for a SQL statement, you can use:

```
select /*+ OPT_PARAM('cell_offload_processing' 'false') */ max(ORDER_DATE) from SALES;
```

To disable Exadata Storage Server cell offload processing for a database session, you can set the following Oracle database initialization parameter:

```
alter session set CELL_OFFLOAD_PROCESSING=FALSE;
```

To disable Exadata Storage Server cell offload processing for the entire Exadata database, you can set:

```
alter system set CELL_OFFLOAD_PROCESSING=FALSE;
```

Migrating to AWS

When you initially migrate workloads to Exadata, several design changes are implemented as a common practice to favor Smart Scan, including dropping schema indexes to favor full table scans . When you migrate such workloads to non-Exadata platforms, you need to reverse those design changes.

When you migrate your Exadata workloads to AWS, consider these tuning actions to optimize the performance of queries that use Smart Scan:

- Use memory optimized instances and configure a larger SGA to increase the buffer hit ratio.
- Identify queries that run with suboptimal execution plans and tune them to reduce their I/O footprint.
- Adjust optimizer parameters such as `db_file_multiblock_read_count` and `optimizer_index_cost_adj` to avoid full table scans.
- Choose an appropriate compression option.
- Create additional schema indexes as required.

Storage indexes

A storage index is a memory-based structure that reduces the amount of physical I/O performed in an Exadata storage cell. The storage index keeps track of minimum and maximum column values, and this information is used to avoid unnecessary I/O operations. The storage index enables Exadata to speed up I/O operations by eliminating access to storage regions that don't contain the data the queries are looking for.

The following database statistics help assess the benefits of storage indexes in the system:

- `cell physical IO bytes saved by storage index` – Shows how many bytes of I/O were eliminated by the application of storage indexes at the storage cell level.
- `cell IO uncompressed bytes` – Reflects the data volume for predicate offloading after storage index filtering and any decompression.

For more information about these, see the Oracle documentation. In the following example from an AWR report collected from an Exadata system, 5.4 Gbps of read operations were Smart Scan eligible. 4.6 Gbps of those I/O operations were processed by cells before predicate offloading, and 55 MBps were returned to the compute nodes with a savings of 820 MBps I/O by storage index. In this example, the dependency on the storage index isn't very high.

Statistic	Total	per Second
cell physical IO bytes eligible for predicate offload	39,217,360,822,272	5,443,436,754.68
cell physical IO interconnect bytes returned by smart scan	400,725,918,720	55,621,456.14
cell physical IO bytes saved by storage index	5,913,287,524,352	820,775,330.00
cell IO uncompressed bytes	33,217,076,600,832	4,610,586,117.33

Migrating to AWS

If you migrate to a platform that doesn't provide a storage index, in most cases, you can create schema indexes to avoid full table scans and reduce the number of blocks that are accessed by queries. To test the influence of storage indexes on your workload performance, set the `kcfis_storageidx_disabled` parameter to TRUE at the system, session, or query level.

For example, use the following SQL statement to disable the storage index at the session level:

```
alter session set "_KCFIS_STORAGEIDX_DISABLED"=TRUE;
```

Smart Flash Cache

The Exadata Smart Flash Cache feature caches database objects in flash memory to boost the speed of accessing database objects. Smart Flash Cache can determine which types of data segments and operations need to be cached. It recognizes different types of I/O requests so that non-repeatable data access (such as RMAN backup I/O) doesn't flush database blocks from the cache. You can move hot tables and indexes to Smart Flash Cache with `ALTER` commands. When you use the Write Back Flash Cache feature, Smart Flash can also cache database block write operations.

The Exadata storage server software also provides Smart Flash Logging to speed up redo log write operations and reduce the service time for the log file sync event. This feature performs redo write operations simultaneously to both flash memory and the disk controller cache, and completes the write operation when the first of the two completes.

The following two statistics provide quick insights into Exadata Smart Flash Cache performance. These are available in dynamic performance views such as `V$SYSSTAT` and in the *Global Activity Statistics* or *Instance Activity Statistics* section of the AWR report.

- `Cell Flash Cache read hits` – Records the number of read requests that found a match in the Smart Flash Cache.
- `Physical read requests optimized` – records the number of read requests that were optimized either by Smart Flash Cache or through storage indexes.

Exadata metrics collected from storage cells are also useful for understanding how a workload uses Smart Flash Cache. The following CellCLI command lists different metrics available for monitoring Smart Flash Cache usage.

```
CellCLI> LIST METRICDEFINITION ATTRIBUTES NAME,DESCRIPTION WHERE OBJECTTYPE =
 FLASHCACHE
FC_BYKEEP_DIRTY                         "Number of megabytes unflushed for keep objects
 on FlashCache"
FC_BYKEEP_OLTP                          "Number of megabytes for OLTP keep objects in
 flash cache"
FC_BYKEEP_OVERWR                      "Number of megabytes pushed out of the FlashCache
 because of space limit
for keep objects"
FC_BYKEEP_OVERWR_SEC                "Number of megabytes per second pushed out of the
 FlashCache because of
```

```
space limit for keep objects"
...
```

Migrating to AWS

Smart Flash Cache doesn't exist on AWS. There are few options to mitigate this challenge and avoid performance degradation when migrating Exadata workloads to AWS, including these, which are discussed in the following sections:

- Using extended memory instances
- Using instances with NVMe-based instance stores
- Using AWS storage options for low latency and high throughput

However, these options can't reproduce Smart Flash Cache behavior, so you need to assess the performance of your workload to make sure that it continues to meet your performance SLAs.

Extended memory instances

Amazon EC2 offers many high memory instances, including instances with 12 TiB and 24 TiB of memory. These instances support extremely large Oracle SGAs that can reduce the impact of the missing Smart Flash Cache by increasing the buffer hit ratio.

Instances with NVMe-based instance stores

An instance store provides temporary block-level storage for the instance. This storage is located on disks that are physically attached to the host computer. Instance stores allow workloads to achieve low latency and higher throughput by storing data on NVMe-based disks. The data in an instance store persists only during the lifetime of an instance, so instance stores are ideal for temporary tablespaces and caches. Instance stores can support millions of IOPS and more than 10 Gbps throughput at microseconds latency depending on the type of instances and I/O size. For more information about instance store read/write IOPS and throughput support for different instance classes, see general purpose, compute optimized, memory optimized, and storage optimized instances in the Amazon EC2 documentation.

In Exadata, the Database Flash Cache allows users to define a second buffer cache tier on instance store volumes with an average I/O latency of 100 microseconds to improve the performance of read workloads. You can activate this cache by setting two database initialization parameters:

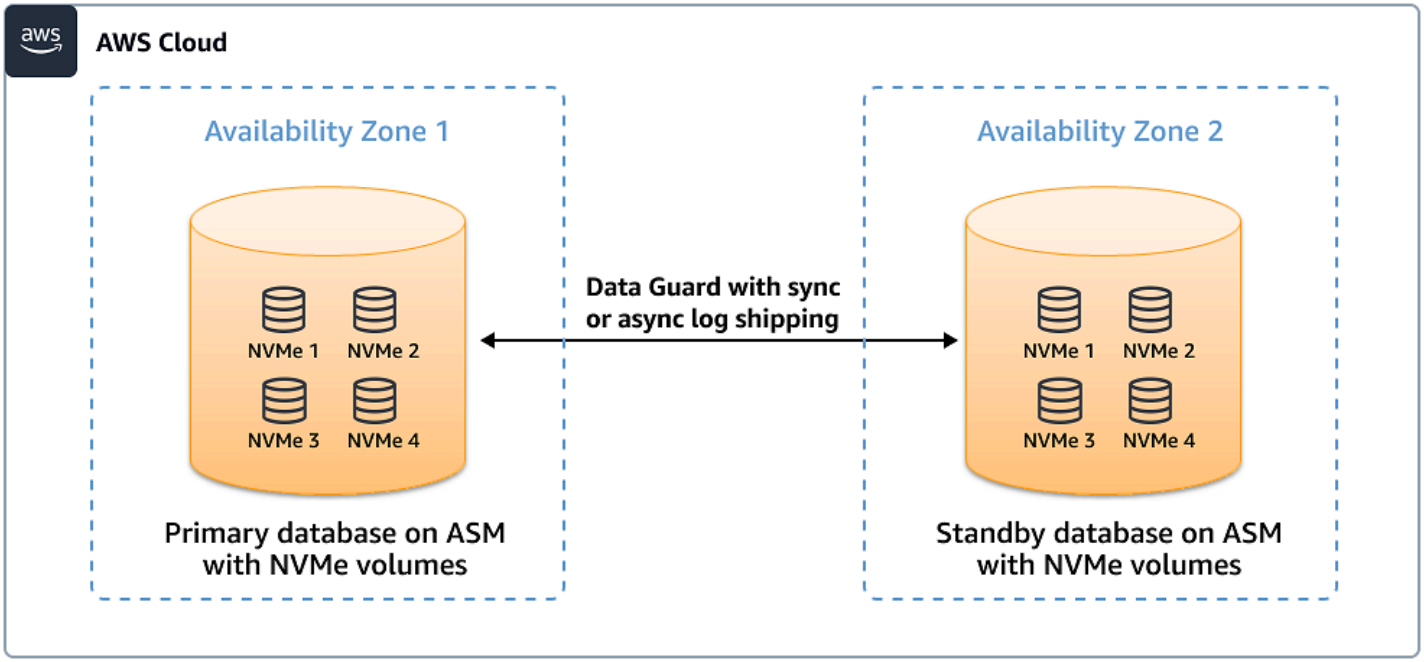
- `db_flash_cache_file = /<device_name>`

- `db_flash_cache_size = <size>G`

You can also design high-performance architectures for Oracle databases that are hosted on Amazon EC2 by placing database files on instance stores, and using the redundancy provided by Oracle Automatic Storage Management (ASM) and Data Guard for data protection and recovery in case the data is lost on the instance stores. These architecture patterns are ideal for applications that require extreme I/O throughput at low latency and can afford a higher RTO to recover the system in certain failure scenarios. The following sections briefly discuss two architectures that include database files hosted on NVMe-based instance stores.

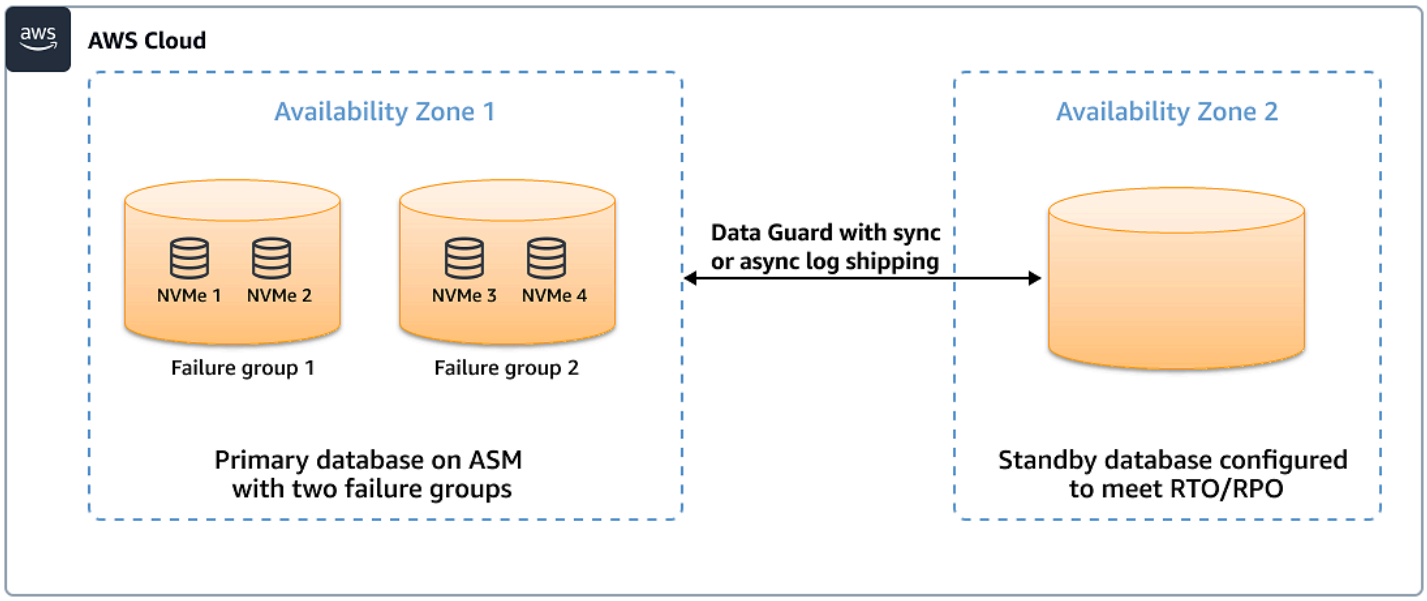
Architecture 1. Database is hosted on instance stores on both primary and standby instances with Data Guard for data protection

In this architecture, the database is hosted on an Oracle ASM disk group to distribute the I/O across multiple instance store volumes for high throughput, low latency I/O. A Data Guard standby is placed in the same or in another Availability Zone for protection from data loss in instance stores. The disk group configuration depends on RPO and commit latency. If the instance store is lost on the primary instance for any reason, the database can fail over to the standby with zero or minimum data loss. You can configure the Data Guard observer process to automate the failover. Both read and write operations benefit from high throughput and low latency offered by instance stores.



Architecture 2. Database is hosted on an ASM disk group with two failure groups that combine both EBS volumes and instance stores

In this architecture, all read operations are performed from local instance stores by using the `ASM_PREFERRED_READ_FAILURE_GROUP` parameter. Write operations apply to both instance store volumes and Amazon Elastic Block Store (Amazon EBS) volumes. However, Amazon EBS bandwidth is dedicated to write operations as read operations are offloaded to instance store volumes. In case of data loss in the instance stores, you can recover data from the ASM failure group based on EBS volumes or from the standby database. For more information, see the Oracle white paper Mirroring and Failure Groups with ASM. You can deploy the Data Guard standby in a different Availability Zone for additional protection.



Amazon RDS for Oracle supports Database Smart Flash Cache and temporary tablespaces on instance stores. Oracle database workloads can use this feature to achieve lower latency for read operations, higher throughput, and efficient utilization of Amazon EBS bandwidth for other database I/O operations. This feature is currently supported on db.m5d, db.r5d, db.x2idn, and db.x2iedn instance classes. For the latest information, see Supported instance classes for the RDS for Oracle instance store in the Amazon RDS documentation.

AWS storage options for workloads that demand low latency and high throughput

The EBS volume types that Amazon RDS for Oracle currently supports, gp2, gp3, and io1, are based on solid-state drives (SSDs). When you deploy these volume types with the appropriate Amazon EBS-optimized instance classes, they can usually meet your service time, IOPs, and throughput requirements.

For self-managed Oracle database deployments on Amazon EC2, Amazon EBS io2 and io2 Block Express EBS volumes provide additional choices for workloads that need lower latency and higher throughput.

Workloads that need higher throughput or microsecond latencies can use storage volumes that aren't based on Amazon EBS when deploying as self-managed Oracle databases on Amazon EC2. For example, Amazon FSx for OpenZFS can deliver more than 1 million IOPS with 20 Gbsp or higher throughput with a latency of a few hundred microseconds. Amazon FSx for NetApp ONTAP can deliver hundreds of thousands of IOPS with a latency of less than one millisecond.

Hybrid Columnar Compression

Oracle Hybrid Columnar Compression (HCC) in Exadata allows the highest compression ratio among available compression options for Oracle databases. It uses both database and Exadata storage capabilities to achieve a high compression ratio that leads to reduced storage cost and better performance for certain workloads due to reduced I/O. There are two HCC options: Warehouse Compression and Archive Compression. Warehouse Compression reduces storage costs and provides better performance when you use Smart Scan queries to decompress HCC compression units in storage cells. Archive Compression is an information lifecycle management (ILM) solution that provides a higher compression ratio at the cost of performance overhead and is meant for rarely accessed data.

You can use the following query to identify tables that have compression enabled:

```
select table_name, compression, compress_for from dba_tables where compression =
 'ENABLED';
```

For HCC-enabled tables, the `compress_for` column shows one of the following values depending on the configuration:

```
QUERY LOW,QUERY HIGH,ARCHIVE LOW,ARCHIVE HIGH
```

Additionally, you can use the `DBMS_COMPRESSION.GET_COMPRESSION_TYPE` function to understand the HCC configuration of a segment, and the `dbms_compression.get_compression_ratio` procedure to analyze the compression ratio of a segment that's enabled to use HCC.

In the following example, `TEST_HCC` is a table that's approximately 30 MB in size. It is HCC-enabled through the use of the `ARCHIVE HIGH` option. The output of

dbms_compression.get_compression_ratio shows that the table gets a compression ratio of 19.4.

Without HCC, this table will grow to approximately 580 MB in size.

```
SET SERVEROUTPUT ON

DECLARE
l_blkcnt_cmp PLS_INTEGER;
l_blkcnt_uncmp PLS_INTEGER;
l_row_cmp PLS_INTEGER;
l_row_uncmp PLS_INTEGER;
l_cmp_ratio NUMBER;
l_comptype_str VARCHAR2(32767);

BEGIN
DBMS_COMPRESSION.get_compression_ratio (
        scratchtbsname => 'USERS',
        ownname => upper('TEST_USER'),
        objname => upper('TEST_HCC'),
        subobjname => NULL,
        comptype => DBMS_COMPRESSION.COMP_ARCHIVE_HIGH,
        blkcnt_cmp => l_blkcnt_cmp,
        blkcnt_uncmp => l_blkcnt_uncmp,
        row_cmp => l_row_cmp,
        row_uncmp => l_row_uncmp,
        cmp_ratio => l_cmp_ratio,
        comptype_str => l_comptype_str,
        subset_numrows => DBMS_COMPRESSION.comp_ratio_allrows,
        objtype SQL> => DBMS_COMPRESSION.objtype_table
        );

  DBMS_OUTPUT.put_line('Number of blocks used (compressed) : ' || l_blkcnt_cmp);
  DBMS_OUTPUT.put_line('Number of blocks used (uncompressed) : ' || l_blkcnt_uncmp);
  DBMS_OUTPUT.put_line('Number of rows in a block (compressed) : ' || l_row_cmp);
  DBMS_OUTPUT.put_line('Number of rows in a block (uncompressed) : ' || l_row_uncmp);
  DBMS_OUTPUT.put_line('Compression ratio : ' || l_cmp_ratio);
  DBMS_OUTPUT.put_line('Compression type : ' || l_comptype_str);
  END;
  /
Compression Advisor self-check validation successful. select count(*) on both
Uncompressed and EHCC Compressed format = 3851900 rows
Number of blocks used (compressed) : 3816
Number of blocks used (uncompressed) : 74263
```

```
Number of rows in a block (compressed) : 1009
Number of rows in a block (uncompressed) : 51
Compression ratio : 19.4
Compression type : "Compress Archive High"
PL/SQL procedure successfully completed.
```

Migrating to AWS

Because HCC is a proprietary, hardware-dependent compression technology, segments that are enabled for HCC must be uncompressed during migration to the target platform on AWS. It is a common practice to store archived data along with less frequently accessed data in Exadata because of the high compression ratio offered by the Exadata HCC feature. To address the challenge of managing larger datasets on AWS without HCC, consider moving inactive parts of your dataset out of your primary database and storing them in other inexpensive and efficient storage solutions such as Amazon S3 Intelligent-Tiering. This might require changes in the application logic or workflow depending on how your application accesses the inactive data. For additional information, see the Data lifecycle management section of this guide.

For workloads that have dependencies on Oracle Database, HCC-enabled segments can also be converted to use the basic or advanced compression features offered by Oracle Database. Basic and advanced compression are supported in Oracle Database EE only. Advanced compression requires additional licensing. Amazon EC2 and Amazon RDS support both of these compression options.

I/O Resource Management

I/O Resource Management (IORM) is an Exadata feature that manages how multiple workloads and databases share the I/O resources of an Exadata system. IORM complements the Oracle Database Resource Manager (DBRM) to provide necessary isolation for different workloads in a consolidated environment. Whenever I/O requests start to saturate the I/O capacity of storage cell servers, IORM schedules and prioritizes incoming I/O requests based on the resource plans you configured.

You can collect IORM metrics from Exadata storage cells by using the script `metric_iorm.pl` as discussed in My Oracle Support (MOS) Note 337265.1, Tool for Gathering I/O Resource Manager Metrics: metric_iorm.pl (requires an Oracle account). These metrics can be useful for organizing workloads that run in a consolidated environment in Exadata when you migrate the workloads to the target platform on AWS.

Migrating to AWS

In the AWS Cloud, we recommend that you host different workloads on separate instances. This approach provides more flexibility in maintaining the databases according to the resource, performance, and SLA requirements of individual applications instead of consolidating them into a single instance. The following practices can be useful when you migrate such workloads to AWS:

- Identify interdependencies among databases and classify the workloads that must be migrated to the same instance on the target platform. These databases might have unresolvable cross-schema references or low-latency database link connectivity.
- Based on the statistics you collected by using the `metric_iorm.pl` script, identify databases and workloads that initiate and benefit from IORM. Use this information to determine the databases that can be consolidated or migrated to independent instances. Choose appropriate storage types and instance classes to avoid I/O saturation.
- If the target platform is Oracle Database, consider using Oracle Database Resource Manager (DBRM) to prioritize or throttle resources such as CPU, PGA, and parallelism for multiple workloads that are consolidated in the same instance as multiple pluggable databases or schemas.
- Consider implementing caching solutions such as Amazon ElastiCache and Amazon RDS for Oracle read replicas to serve read-only workloads. These solutions reduce the I/O footprint on the primary instance.
- For workloads that don't have a dependency on Oracle Database, Amazon Aurora provides a distributed and decoupled architecture that provides high I/O throughput. You can meet the demands of a heavy, I/O-intensive workload by designing an Aurora cluster with an appropriate number of reader instances and by using features such as Amazon Aurora global databases.

Persistent Memory (PMEM)

Oracle Exadata X8M and later releases use Persistent Memory (PMEM) to achieve higher I/O rates as well as low-latency storage access. Exadata is able to achieve less than 19 microseconds storage latency with PMEM combined with Remote Direct Memory Access over Converged Ethernet (RoCE) to bypass layers of code. The PMEM cache works in conjunction with Exadata Smart Flash Cache to provide three tiers of storage layers: PMEM acts as the hot storage tier, Smart Flash Cache as the warm storage tier, and disks in storage cells as the cold storage tier to deliver higher IOPS and improved performance for commit operations.

The performance benefits of PMEM can be seen from AWR statistics as low service time, in microseconds, for read wait events such as cell single block physical read, and redo log write wait events such as log file sync and log file parallel write. You can also monitor PMEM cache hits by using additional statistics such as cell pmem cache read hits and cell pmem cache writes, which are available in dynamic performance views such as V$SYSSTAT and in the AWR report.

Migrating to AWS

EC2 instances on AWS don't currently offer PMEM features. However, EC2 instances with large memory capabilites can support extremely large Oracle SGAs that can cache Oracle Database objects. For workloads that need read and write service time in microseconds, Amazon FSx for OpenZFS can deliver more than 1 million IOPS with 20 Gbps or better throughput with a latency of a few hundred microseconds.

Summary of Exadata features and AWS alternatives

The following table summarizes common tactics and approaches for addressing missing Exadata features when you migrate your Exadata workloads to AWS. For a detailed discussion of each Exadata feature and AWS alternative, see the previous sections.

Exadata feature	Tactics to address the feature gap	Applicable migration strategy
Smart Scan	Use memory optimized instances.	Rehost, replatform, refactor
	Optimize SGA/PGA configuration.	Rehost, replatform
	Adjust optimizer parameters such as `optimizer_index_cost_adj` .	Rehost, replatform
	Create additional schema indexes.	Rehost, replatform, refactor
	Optimize SQL to reduce I/O footprint.	Rehost, replatform, refactor

Exadata feature	Tactics to address the feature gap	Applicable migration strategy
Storage indexes	Create appropriate schema indexes.	Rehost, replatform, refactor
Smart Flash Cache	Use memory optimized instances.	Rehost, replatform, refactor
	Optimize SGA.	Rehost, replatform
	Configure the Database Flash Cache feature on Amazon EC2 or Amazon RDS for Oracle with local SSD storage.	Rehost, replatform
	Use external caching solutions such as Amazon ElastiCache.	Rehost, replatform, refactor
	Consider building a high-performance architecture for Oracle on Amazon EC2 by using instances with NVMe disks.	Rehost
	Consider io2 Block Express EBS volumes and Amazon FSx services as the storage layer.	Rehost
Hybrid Columnar Compression (HCC)	Migrate archive and infrequently accessed data to other storage solutions.	Rehost, replatform, refactor
	Use advanced compression or basic compression.	Rehost, replatform
I/O Resource Management (IORM)	Use appropriate instances and storage types to avoid I/O saturation	Rehost, replatform, refactor

Exadata feature	Tactics to address the feature gap	Applicable migration strategy
	Use Oracle Database Resource Manager.	Rehost, replatform
	Use external caching solutions such as Amazon ElastiCache.	Rehost, replatform, refactor
	Use Amazon Aurora, which offers high I/O scalability.	Refactor
Persistent Memory (PMEM)	Use EC2 instances with high memory.	Rehost, replatform, refactor
	Consider io2 Block Express EBS volumes and Amazon FSx services as a storage layer for low latency.	Rehost

Tools for the discovery phase

This section discusses the AWS and Oracle tools that are available for the discovery phase and the purpose of each. You can use one or more tools from this list based on your requirements, skills, and the licenses required for tools such as Oracle Automatic Workload Repository (AWR).

Purpose	Tool
Determine which Exadata features you're currently using	Oracle Automatic Workload Repository (AWR), Oracle Enterprise Manager (OEM), dictionary views, Cell Control Command-Line Interface (CellCLI)
Determine which Enterprise Edition features you're currently using	Dictionary views, AWS Schema Conversion Tool (AWS SCT)
Analyze database statistics and wait events	AWR, OEM, dictionary views

Purpose	Tool
Estimate resources and right-size	AWR, OEM, dictionary views, CellCLI

AWR

Oracle Automatic Workload Repository (AWR) is included in Oracle Database Enterprise Edition (EE). It automatically collects, processes, and maintains performance statistics for the database. You can access these statistics through AWR reports, database views, or Oracle Enterprise Manager (OEM). When you consolidate multiple workloads into a single database by using different Oracle services, AWR collects service-level statistics that are useful for right-sizing those consolidated workloads into stand-alone instances on AWS.

AWR is licensed under the Oracle Diagnostics Pack (see licensing information). Statspack, an alternative to AWR, is a free tool for analyzing performance statistics and metrics. However, Statspack doesn't provide the same level of metrics and statistics related to Exadata components as AWR.

You can generate AWR reports at the instance level or globally for all instances of a Real Application Cluster (RAC) database or for a specific SQL ID. For more information, see the Oracle Database performance tuning guide.

You can use AWR to analyze your Exadata workload, the specific Exadata features used by your workload, the benefits from Exadata-specific features, different database statistics and wait events, and the resources required for hosting the workload on AWS. These rich statistics and metrics collected by AWR span multiple layers of the Exadata system, including database servers, storage cells, interconnect network, RAC, and ASM disk groups. The following table summarizes the key AWR metrics and statistics to focus on during an Exadata migration. Covering all relevant statistics and metrics for the discovery phase is beyond the scope of this guide.

Metric	Indicates	Relevance
User commits	Commits issued at the boundary of a transaction	Nature of the workload
Buffer cache hit ratio	How often a requested block has been found in the buffer	Nature of the workload

Metric	Indicates	Relevance
	cache without requiring disk access	
Physical read multi-block requests	The total number of read requests that were read in two or more database blocks per request	Nature of the workload, I/O characteristics
Physical read total I/O requests	The total number of read requests	Nature of the workload, I/O characteristics
Cell physical I/O bytes eligible for predicate offload	The number of bytes on disk eligible for predicate offloading	Exadata Smart Scan feature dependency
Cell physical I/O interconnect bytes	The number of I/O bytes that were exchanged over the interconnect between the database host and the cells	Exadata Smart Scan feature dependency
Cell physical I/O interconnect bytes returned by Smart Scan	The number of I/O bytes that are returned by the cell for Smart Scan operations	Exadata Smart Scan feature dependency
Cell physical I/O bytes saved by storage index	How many bytes of I/O were eliminated by the applicati on of storage indexes at the storage cell level.	Exadata Storage Index feature dependency
Physical optimized read requests	The number of read requests that were optimized either by the Exadata Smart Flash Cache or through storage indexes	Exadata storage index and Smart Flash Cache feature dependency

Metric	Indicates	Relevance
Cell Flash Cache read hits	The number of read requests that found a match in the Exadata Smart Flash Cache	Exadata Smart Flash Cache feature dependency

CellCLI

The Cell Control Command-Line Interface (CellCLI) is the command-line administration and monitoring tool for Exadata storage cells that is preconfigured in Exadata storage cell servers. This utility extracts information directly from the hardware or storage server software.

For the full list of metrics available for CellCLI, see the Oracle Exadata documentation. To see a list of all available metrics and their definitions, run the following command while connected to CellCLI from one of the storage servers.

```
CellCLI>LIST metricDefinition WHERE objectType=cell;
```

To analyze different metrics, connect directly to the storage server and use the CellCLI `list metriccurrent` or `list metrichistory` command to read it.

```
CellCLI> list metriccurrent

         CD_BY_FC_DIRTY                          CD_00_celladm-01
       0.000 MB
…
…
         SIO_IO_WR_RQ_FC_SEC                     SMARTIO
          0.000 IO/sec
         SIO_IO_WR_RQ_HD                         SMARTIO
          3,660,097 IO requests
         SIO_IO_WR_RQ_HD_SEC                     SMARTIO
          0.000 IO/sec
```

You must run CellCLI on individual cell nodes to gather metrics for that node. You can also run CellCLI commands from `dcli` to collect metrics for a group of cell nodes.

```
./dcli -g mycells "cellcli -e list metriccurrent GD_IO_BY_R_LG \
```

```
  attributes alertstate, metricvalue";
```

Exadata offloads many resource-intensive tasks to storage cell servers. Therefore, it's important to understand how various resources are used on the storage cells to right-size the compute instances in the target environment. The following table shows a few key Exadata metrics from storage cell servers that can help you understand how resources are used in the storage cells.

Metric	Description
CL_CPUT	The cell CPU utilization
CL_MEMUT	The percentage of total physical memory used
N_HCA_MB_RCV_SEC	The number of megabytes received by the InfiniBand interfaces per second
N_HCA_MB_TRANS_SEC	The number of megabytes transmitted by the InfiniBand interfaces per second
N_MB_RECEIVED_SEC	The rate (number of megabytes) received per second from a particular host
N_MB_SENT_SEC	The rate (number of megabytes) sent per second from a particular host
FL_RQ_TM_W_RQ	Average redo log write request latency
FL_IO_TM_W_RQ	Average redo log write latency, which includes write I/O latency only
FC_IO_RQ_W_SKIP_SEC	The number of write I/O requests per second that bypass the Flash Cache
FC_IO_RQ_R_SKIP_SEC	The number of read I/O requests per second that bypass the Flash Cache
SIO_IO_EL_OF_SEC	The number of megabytes per second eligible for offload by smart I/O

Metric	Description
SIO_IO_OF_RE_SEC	The number of interconnect megabytes per second returned by smart I/O
SIO_IO_RD_FC_SEC	The number of megabytes per second read from the Flash Cache by smart I/O
SIO_IO_RD_HD_SEC	The number of megabytes per second read from the hard disk by smart I/O
SIO_IO_WR_FC_SEC	The number of megabytes per second of Flash Cache population write operations by smart I/O
SIO_IO_SI_SV_SEC	The number of megabytes per second saved by the storage index

The following CellCLI command runs against an Exadata cell node to show the statistics related to key Exadata features.

```
CellCLI> list metrichistory where collectionTime > '2022-06-13T15:42:00+01:00' and
 collectionTime < '2022-06-13T15:43:00+01:00' and name like 'SIO_.*SEC.*'

         SIO_IO_EL_OF_SEC        SMARTIO         1,223 MB/sec
 2022-06-13T15:42:03+01:00

         SIO_IO_OF_RE_SEC        SMARTIO         34.688 MB/sec
 2022-06-13T15:42:03+01:00

         SIO_IO_PA_TH_SEC        SMARTIO         0.000 MB/sec
 2022-06-13T15:42:03+01:00

         SIO_IO_RD_FC_HD_SEC     SMARTIO         0.174 MB/sec
 2022-06-13T15:42:03+01:00

         SIO_IO_RD_FC_SEC        SMARTIO         843 MB/sec
 2022-06-13T15:42:03+01:00

         SIO_IO_RD_HD_SEC        SMARTIO         0.101 MB/sec
 2022-06-13T15:42:03+01:00
```

```
         SIO_IO_RD_RQ_FC_HD_SEC  SMARTIO         0.183 IO/sec
 2022-06-13T15:42:03+01:00

         SIO_IO_RD_RQ_FC_SEC     SMARTIO         850 IO/sec
 2022-06-13T15:42:03+01:00

         SIO_IO_RD_RQ_HD_SEC     SMARTIO         0.000 IO/sec
 2022-06-13T15:42:03+01:00

         SIO_IO_RV_OF_SEC        SMARTIO         3.392 MB/sec
 2022-06-13T15:42:03+01:00

         SIO_IO_SI_SV_SEC        SMARTIO         362 MB/sec
 2022-06-13T15:42:03+01:00

         SIO_IO_WR_FC_SEC        SMARTIO         0.008 MB/sec
 2022-06-13T15:42:03+01:00

         SIO_IO_WR_HD_SEC        SMARTIO         0.000 MB/sec
 2022-06-13T15:42:03+01:00

         SIO_IO_WR_RQ_FC_SEC     SMARTIO         0.017 IO/sec
 2022-06-13T15:42:03+01:00

         SIO_IO_WR_RQ_HD_SEC     SMARTIO         0.000 IO/sec
 2022-06-13T15:42:03+01:00
```

In these example statistics, `SIO_IO_SI_SV_SEC` indicates that 362 MBps of I/O is saved by the storage index, `SIO_IO_RD_RQ_FC_SEC` indicates that 850 I/O per second is served by the Flash Cache, and `SIO_IO_OF_RE_SEC` indicates that 34 MBps of I/O is returned by Smart Scan.

In another example, the following `dcli` command output shows very low CPU utilization across all cell nodes in an Exadata system. This potentially indicates a workload that doesn't benefit significantly from Exadata storage layer features.

```
dcli -g
../cell_group cellcli -e \
list metriccurrent where name='CL_CPUT';
cm01cel01: CL_CPUT cm01cel01 0.2 %
cm01cel02: CL_CPUT cm01cel02 0.2 %
cm01cel03: CL_CPUT cm01cel03 0.7 %
```

OEM Cloud Control

Oracle Enterprise Manager (OEM) Cloud Control provides centralized, comprehensive, end-to-end monitoring, management, administration, and support capabilities for all key Oracle systems. The best way to monitor and manage Exadata is by using OEM, because it is tightly integrated with all Exadata software and hardware components.

You can access many of the metrics that have been discussed so far by using OEM dashboards. Some of the key dashboards that are helpful in the discovery phase of Exadata migration are:

- Resource utilization on database servers
- Storage and I/O statistics from the storage cells
- InfiniBand switch statistics
- ASM disk group statistics
- Database performance using AWR, Automatic Database Diagnostic Monitor (ADDM), and Active Session History (ASH)
- Advisory tools such as SGA Advisory and SQL Tuning Advisor

However, some of the dashboards are licensed under different packs such as the Oracle Diagnostics Pack or Oracle Tuning Pack. For details, see the Oracle licensing information.

Database views

You can query the database views (dictionary views and dynamic performance views) in an Oracle database to retrieve useful statistics related to Exadata features for your database or instance. The following table shows some of the key views that display critical statistics that are useful for the discovery phase.

View	Description
`DBA_TABLES`	Identifies tables that use the HCC feature
`DBA_HIST_SYSSTAT`	Shows historical Exadata-related statistics
`DBA_FEATURE_USAGE_STATISTICS`	Displays information about database feature usage

View	Description
DBA_HIST_SQLSTAT	Displays historical information about SQL statistics
DBA_HIST_ASM_DISKGROUP_STAT	Displays performance statistics for ASM disk groups
DBA_HIST_CELL_DISK_SUMMARY	Displays historical information about the performance of disks on cells
DBA_HIST_ACTIVE_SESS_HISTORY	Displays active session history
DBA_HIST_DB_CACHE_ADVICE	Provides predictions of the number of physical read operations for the cache size
DBA_ADVISOR_FINDINGS	Displays findings of various advisory tasks such as SQL Tuning Advisor

The following examples show statistics retrieved from database views that are useful for the discovery phase.

This query shows a single table in the database that's enabled for HCC with `QUERY HIGH` compression mode:

```
select table_name, compression, compress_for from dba_tables where compression =
 'ENABLED';
TABLE_NAME COMPRESS COMPRESS_FOR
------------------------------ -------- ------------
ORDER_ITEMS ENABLED QUERY HIGH
```

This query displays database feature usage, which helps determine feature dependency on Oracle Database Enterprise Edition:

```
select
   name             c1,
   detected_usages  c2,
   first_usage_date c3,
   currently_used   c4
from dba_feature_usage_statistics
```

```
where first_usage_date is not null;
                                                 times    first     used
feature                                          used     used      now
---------------------------------------------- -------- --------- -----
Protection Mode - Maximum Performance            24       18-AUG-20 TRUE
Recovery Area                                    24       18-AUG-20 TRUE
Server Parameter File                            24       18-AUG-20 TRUE
Shared Server                                     4       18-AUG-20 FALSE
Streams (system)                                 24       18-AUG-20 TRUE
Virtual Private Database (VPD)                   24       18-AUG-20 TRUE
Automatic Segment Space Management (system)      24       18-AUG-20 TRUE
Automatic Segment Space Management (user)        24       18-AUG-20 TRUE
Automatic SQL Execution Memory                   24       18-AUG-20 TRUE
Automatic Undo Management                        24       18-AUG-20 TRUE
Character Set                                    24       18-AUG-20 TRUE
Dynamic SGA                                       1       18-AUG-20 FALSE
Locally Managed Tablespaces (system)             24       18-AUG-20 TRUE
Locally Managed Tablespaces (user)               24       18-AUG-20 TRUE
Multiple Block Sizes                              7       25-DEC-20 TRUE
Partitioning (system)                            24       18-AUG-20 TRUE
```

This query shows the total physical read bytes, bytes eligible for cell offloading, and bytes returned from the storage cell for a SQL statement for a specific AWR snapshot:

```
select
  ROUND(physical_read_bytes_delta/EXECUTIONS_DELTA)/1024/1024 phyrd_mb
  , ROUND(IO_OFFLOAD_ELIG_BYTES_TOTAL/EXECUTIONS_DELTA)/1024/1024 elig_mb
  , ROUND(io_interconnect_bytes_delta/EXECUTIONS_DELTA)/1024/1024 ret_mb
  from dba_hist_sqlstat
  where sql_id = 'zg2fg7abfx2y' and snap_id between 12049 and 12050;
PHYRD_MB      ELIG_MB     RET_MB  SAVING%
----------    ---------- ---------- ----------
 10815          10815       3328    69.2%
```

AWS SCT

The AWS Schema Conversion Tool (AWS SCT) makes heterogeneous database migrations predictable. It automatically converts the source database schema and a majority of the database code objects, including views, stored procedures, and functions, to a format that's compatible with the target database. Any objects that can't be automatically converted are clearly marked so you can manually convert them to complete the migration. AWS SCT can predict the efforts required for a heterogeneous migration when a manual action is required to convert database objects. This

tool can also indicate dependencies on Oracle Database Enterprise Edition (EE) features. You can use this analysis to decide whether to consider migrating from EE to SE2. For more information, see the Database editions and versions section earlier in this guide. For information about using AWS SCT for heterogeneous migrations, see the Performing the migration section later in this guide.

Resource requirements for the target platform

We recommend that you size the target database environment on AWS based on the source Exadata utilization, not configuration. Many customers purchase Exadata systems with additional capacity to accommodate anticipated growth for the next three to five years. Typically, when Exadata workloads are migrated to AWS, fewer resources are deployed compared to the configuration of the source Exadata system, so it's misleading to use that original configuration to predict AWS resources.

To estimate the resources required in the target instance, you can use the tools that are discussed in the previous section, such as AWR, database views, OEM, and CellCLI. On AWS, you can scale resources up or down more easily compared with the source Exadata platform. The following sections discuss best practices for estimating resources such as CPU, memory, and IOPS for the target platform. Additionally, AWS account teams and database specialists who have extensive experience in assisting customers with their Exadata migrations can help you size your target environment. AWS has internal tools that the AWS account team can use to estimate the resources required and right-size your target environment on AWS.

CPU and memory requirements

When you migrate your Exadata workloads to an Oracle database deployment option on AWS, such as Amazon RDS for Oracle, you shouldn't base the compute layer resources (CPU and memory) only on the utilization statistics from Exadata database servers. The workload also benefits from Exadata-specific features such as Smart Scan and storage indexes, which offload processing to the storage cells and use the resources of the storage servers. Therefore, you should provision the compute layer in the target instance with additional CPU and memory resources based on your usage of Exadata-specific features and the degree of workload optimization that might be possible during the migration.

It's difficult to accurately estimate the additional CPU resources required. Use the discovery results as a starting point for sizing the target environment. For a rough calculation, consider including

one additional vCPU for every 500 MBps of Smart Scan workloads to the total vCPUs required for the compute layer.

Another approach is to consider the CPU utilization on the storage servers. You could add about 20 percent of the total used CPUs on storage servers to the total vCPUs required for the compute layer as a starting point. You can adjust this percentage based on your use of Exadata features, as determined by tools such as AWR and CellCLI. For example, for low usage, you can add 10 percent for low usage, 20 percent for medium usage, and 40 percent for high usage. If you utilize a total number of 20 CPU threads across all storage servers and Exadata feature usage is observed as medium, you might consider 4 additional vCPUs to compensate for missing Exadata features in the target environment.

After these initial estimates, you should also conduct performance testing on the target environment to determine whether you need to scale the resources allocated. Performance testing could also reveal further workload optimization opportunities that can reduce the resources required.

You might have to increase memory allocation to the Oracle instance to improve the cache hit ratio and reduce the I/O footprint. Your source Exadata platform might not have sufficient memory for large SGA allocations, especially in a consolidated scenario. This might not cause performance issues in Exadata, because I/O operations are generally fast. We recommend that you start with an instance that supports the following memory allocation:

```
Target memory required = larger of (8 GB per vCPUs required, two times the SGA+PGA
 allocation in the source)

SGA+PGA allocation = ~80% of available memory on the instance
```

During performance testing, you can use Oracle features such as buffer pool advisory, SGA target advisory, and PGA memory advisory to fine-tune the SGA and PGA allocation to meet your workload's requirements.

The Amazon EC2 or Amazon RDS instance must have adequate CPU, memory, and I/O resources to handle the anticipated database workload. We recommend that you use a current generation instance class to host your workload on AWS. Current generation instance types, such as instances that are built on the Nitro System, support hardware virtual machines (HVMs). To take advantage of enhanced networking and increased security, you can use Amazon Machine Images (AMIs) for HVMs. Amazon RDS for Oracle currently supports up to 128 vCPU and 3,904 GBs of memory. See DB instance classes in the Amazon RDS documentation for information about hardware

specifications of instance classes available for Amazon RDS for Oracle See Amazon EC2 instance types for a list of EC2 instances with resource details.

I/O requirements

Using AWR reports to estimate resource requirements requires familiarity with workload patterns for peak, off-peak, and average load timings. To estimate IOPS requirements for your workload based on an AWR report collected during peak periods, follow these steps:

1. Review the AWR report to identify physical read I/O requests, physical write I/O requests, physical read total bytes, and physical write total bytes.

   These metrics take the benefits of Exadata-specific features such as storage indexes into account, so they indicate actual IOPS and throughput values that you can use to size the storage I/O layer of your target AWS environment.
2. In the I/O profile section of the AWR report, review the physical read requests optimized and physical write requests optimized values to determine if Smart Scan and other Exadata features related to I/O—such as I/O saved by storage indexes, columnar cache, or Smart Flash Cache—are used. If you see optimized requests in the AWR I/O profile, review system statistics to obtain the details of Smart Scan and storage index metrics such as cell physical I/O bytes eligible for predicate offload, cell physical I/O interconnect bytes returned by Smart Scan, and cell physical I/O bytes saved by storage index.

   Although these metrics aren't directly used to size the target environment, they are useful for understanding how much I/O is saved by Exadata-specific features and tuning techniques to optimize the workload.

```
Total IOPS required for the workload = physical read IO requests + physical write IO
 requests

Total throughput = physical read bytes + physical write bytes
```

The AWR statistics physical read I/O requests, physical write I/O requests, physical read bytes, and physical write bytes reflect the workload's I/O activities, excluding the I/O contributed by non-application components such as RMAN backup and other utilities such as expdp or sqlldr. In those cases, you can consider the AWR statistics physical read total I/O requests, physical write total I/O requests, physical read total bytes, and physical write total bytes to estimate IOPs and throughput requirements.

Databases that run on Exadata typically produce hundreds of thousands of IOPS and very high throughput (over 50 Gbps) because of the factors discussed in earlier sections. However, in most cases, tuning techniques and workload optimization reduce the I/O footprint of the workload drastically.

If I/O requirements are very high, be aware of Amazon EC2 and Amazon RDS limitations. For high Amazon EBS volume throughput, consider using Amazon EC2 instance classes such as x2iedn, x2idn, and r5b, which support up to 260,000 IOPS with a throughput of 10,000 MBps. See Amazon EBS-optimized instances in the Amazon EC2 documentation to review the maximum IOPS and throughput supported by various instances. For Amazon RDS for Oracle, the rb5 instance class supports up to 256,000 IOPS with a throughput of 4,000 MBps. See DB instance classes to review Amazon EBS-optimized instances available for Amazon RDS for Oracle.

You should also understand how IOPS and throughput are measured in the case of different EBS volumes that are available for the target environment. In some cases, Amazon EBS splits or merges I/O operations to maximize the throughput. To learn more, see I/O characteristics and monitoring in the Amazon EC2 documentation and How do I optimize the performance of my Amazon EBS Provisioned IOPS volumes? in the AWS Knowledge Center.

Performance testing on the target platform

You can select the appropriate target instance and storage option on AWS based on the resource information you collect during the discovery phase.

After the target instance is provisioned, we recommend that you conduct load testing to ensure that the provisioned instance and configuration meet your application's performance requirements. You should perform this load testing by using your real application workload for the anticipated number of users and concurrencies instead of using generic load testing tools such as Swingbench. If your target is Amazon RDS for Oracle, Amazon RDS Custom for Oracle, or Amazon EC2, you can use Oracle Real Application Testing, which is a separately licensed feature, to capture production workloads from the source Exadata database and replay them on the target instance to assess performance. For more information about using Real Application Testing on AWS, see the AWS blog posts Use Oracle Real Application Testing features with Amazon RDS for Oracle and Use Oracle Real Application Testing features with Amazon EC2.

If you're planning a heterogeneous migration, where the workload is migrated from Oracle Database to an open source database such as PostgreSQL, it's more challenging to estimate resources because they aren't comparable across different engines. As a general practice, we

recommend that you start with an instance that can support the CPU, memory, and I/O resources that are equivalent to the utilized resources in Exadata, and then right-size the target instance based on load testing results by using AWS scaling options.

Application SLA requirements

During the discovery phase, it's important to determine the SLA requirements of your application that's hosted on Exadata, including recovery time objective (RTO) and recovery point objective (RPO). You should understand these requirements from the business or user point of view instead of copying your current architecture as is to the target platform. For example, your current deployment might be using Oracle Real Application Cluster (RAC) feature, which is integrated with Exadata. However, if your application doesn't really need this feature, it might be feasible to deploy a cost-effective solution on AWS without using RAC.

The following table lists the RTO and RPO that you can achieve with different deployment models on AWS. This information is based on high availability and disaster recovery (HA/DR) options within one AWS Region. You can extend DR capabilities by using a multi-Region deployment model, such as adding a cross-Region read replica in Amazon RDS for Oracle, or using global databases in Amazon Aurora.

Deployment type	RTO (in seconds)	RPO (in seconds)	Comments
Amazon RDS for Oracle with Multi-AZ	~120	0	RTO can vary depending on factors such as the time required for instance recovery.
Amazon RDS Custom for Oracle with self-managed HA solution using Data Guard and Fast Start Failover (FSFO)	~120	0	Building the appropriate HA solution is your responsibility. As a best practice, deploy the standby instance in a different Availabil ity Zone from the primary instance.

Deployment type	RTO (in seconds)	RPO (in seconds)	Comments
Self-managed instances on Amazon EC2 by using Data Guard and FSFO	~120	0	Building the appropriate HA solution is your responsibility. As a best practice, deploy the standby instance in a different Availability Zone from the primary instance.
Aurora PostgreSQL-Compatible Edition	< 30	0	If you use a reader instance, failover can complete in a few seconds.
Amazon RDS for PostgreSQL with Multi-AZ	~120	0	
RAC on AWS with Oracle Active Data Guard	0	0	This deployment type uses one of the RAC deployment options on AWS with Data Guard replication to another Availability Zone.

As with the deployment model, choosing the right migration and rollback strategies and migration tools is critical to meeting the SLA requirements of your business. This topic is covered in the detail in the Performing the migration section of this guide.

Data lifecycle management and retention policy

Organizations typically keep data for a long time to meet their compliance requirements. It's common practice to see the entire dataset for an application, including active data, less frequently

accessed data, and archived data, stored in a single database hosted on Exadata using features such as HCC. It might not be efficient to follow the same practice when you migrate your Exadata workloads to AWS. AWS provides several storage solutions, such as Amazon S3 Intelligent-Tiering and Amazon S3 Glacier to efficiently store, query, and retrieve infrequently accessed and archived data instead of keeping them in the transactional database. For more information about different approaches for handling archived and infrequently accessed data during migration to AWS, see the Performing the migration section of this guide.

Other factors

Understanding the tools and products available for use under current Oracle license agreements will be helpful in choosing the right migration strategy to AWS. For example, if you have a license and the skills to use Oracle GoldenGate, it could be an alternative to using AWS DMS as the migration tool. For more information, see the Performing the migration section of this guide.

In addition, we recommend that you collect the details of all inbound and outbound interfaces to your database on Exadata. This includes all application components that connect to the database, interdatabase connectivity using database links, and foreign database connectivity. You should also include these interfaces in your functional and load testing on the target instance, in case an existing interface requires changes to work in the target architecture. For example, Amazon RDS for Oracle doesn't support foreign database connectivity using Oracle Database Gateway products, so you might need to rearchitect the interface to use other solutions such as AWS Glue or migrate those databases to Amazon RDS Custom for Oracle or self-managed Oracle databases on Amazon EC2.

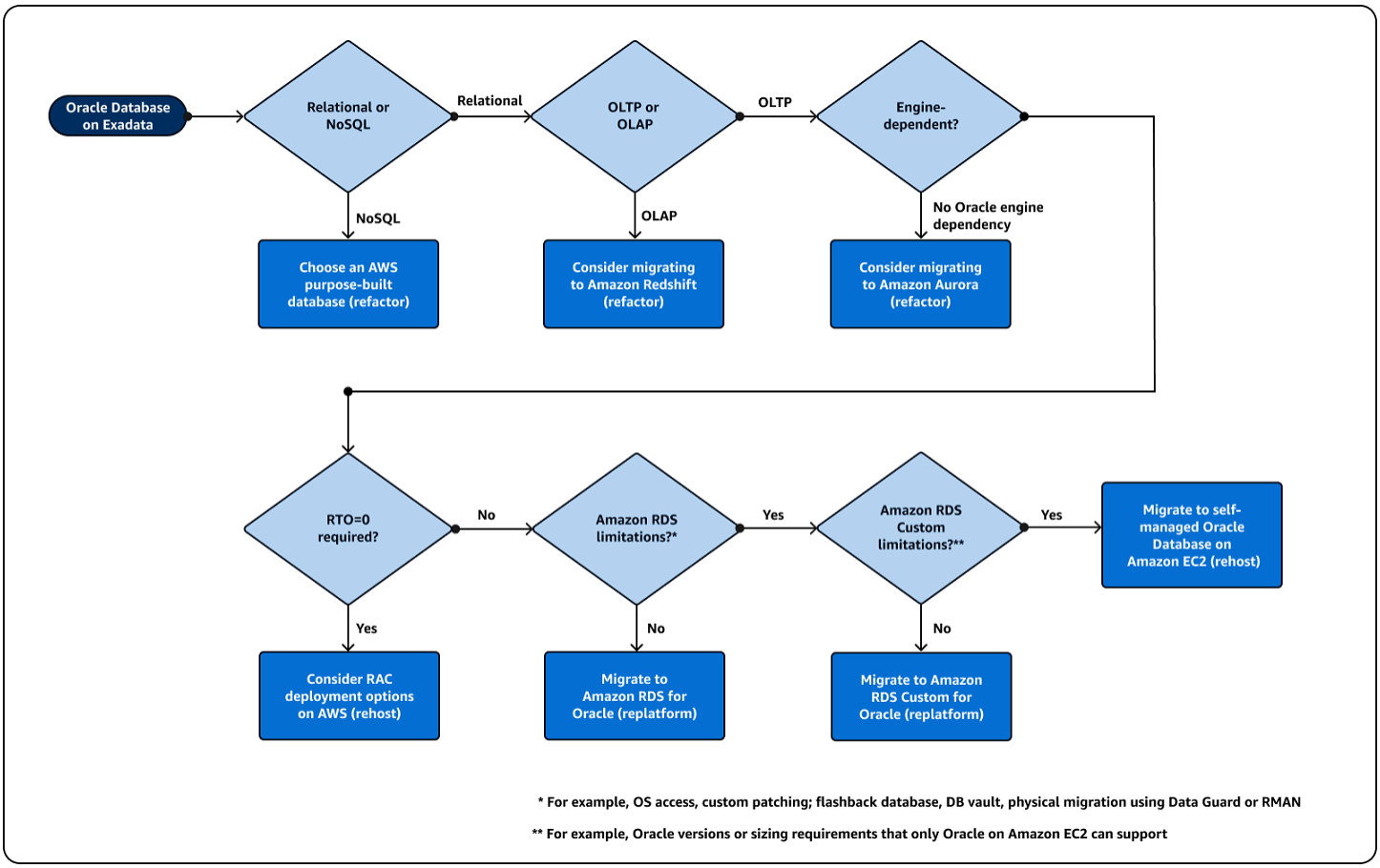
We also recommend that you consider the database features that your workload uses in order to choose the right target environment on AWS. If your workload has a dependency on Oracle Database add-on features, it should be migrated to an AWS service that supports those features. For example, if your workload has a dependency on Oracle Database Vault, you should host it on Amazon RDS Custom for Oracle or as a self-managed database on Amazon EC2.

For information about supported features, see the Amazon RDS for Oracle and Amazon RDS Custom for Oracle documentation.

Decision flowchart

The following diagram provides a simple decision flowchart for migrating an application that runs on Exadata to AWS. The flowchart offers directional guidance on target databases options, but no

two migration projects are the same. We recommend that you perform a discovery assessment or proof of concept (POC) before making any decisions.



Performing the migration

Your Oracle database in AWS must be properly migrated, configured, and tuned to handle the anticipated workload. Before tuning, the first AWS performance test might not meet the performance of the on-premises Exadata environment. However, with proper sizing and tuning, you can meet or exceed the original Exadata performance on AWS in most cases. To achieve your performance objectives, we recommend that you obtain the help of AWS Professional Services or an experienced AWS Partner for your migration.

This section discusses migration approaches, including replatforming, rehosting, and refactoring. It provides an overview of migration services and tools, and covers the best practices for setting up an optimized and performant Oracle database on AWS.

Amazon RDS for Oracle and Amazon EC2 support both Oracle Database Enterprise Edition (EE) and Oracle Database Standard Edition 2 (SE2). Oracle Database EE provides performance, availability, scalability, and security for developing applications such as high-volume online transaction processing (OLTP) applications, query-intensive data warehouses, and demanding internet applications. It provides all the components of Oracle Database and additional enhancements with the purchase of options and packs. On-premises Exadata systems use Oracle Database EE.

> **Note**
>
> Although AWS supports both Oracle editions, this section assumes that Oracle Database EE will be used for the Oracle database in AWS.

For additional information about migration strategies and architectures, see the AWS Prescriptive Guidance guide Migrating Oracle databases to the AWS Cloud.

In this section:

- Refactoring recommendations

Exadata to AWS migration tools

There are more than 15 Exadata to AWS migration approaches. The following table shows the most commonly used tools. The table doesn't include Oracle conventional export/import, Oracle SQL*Loader, Oracle SQL Developer Database Copy, Oracle SQL*Developer Export/Import Wizard, Oracle Transportable Tablespaces, Oracle database links using Create Table as Select (CTAS), Oracle external tables, or extract, transform, and load (ETL) solutions.

Migration approach	Supports migration strategy	Physical or logical	Supports change data capture (CDC)	Requires networking to AWS
AWS DMS	All	Logical	Yes	Yes
Oracle GoldenGate	All	Logical	Yes	Yes
Oracle Data Pump	Rehost, replatform	Logical	No	No
Oracle Recovery Manager (RMAN)	Rehost	Physical	No	If you use `RMAN DUPLICATE` or Oracle Secure Backup to Amazon S3
Oracle Data Guard	Rehost	Physical	Yes	Yes

Oracle Data Guard and Oracle Recovery Manager (RMAN) are excellent options for migrating an Exadata database to Amazon EC2. However, Amazon RDS for Oracle doesn't support either of these tools.

You can implement Oracle Data Guard by using the logical standby or physical standby method. A logical standby database applies data manipulation language (DML) statements on the standby

database to keep data synchronized. Logical standby databases are typically used to offload reporting from the primary database. All Oracle Data Guard references in this section apply directly to physical standby. A physical standby database matches the primary database exactly at the block level.

AWS DMS migrations

AWS Database Migration Service (AWS DMS) is a logical replication solution. It supports homogeneous migrations such as migrating an Oracle on-premises database to an Oracle database on AWS, as well as heterogeneous migrations between different database platforms, such as Oracle to Microsoft SQL Server and Oracle to Amazon Aurora PostgreSQL-Compatible Edition. AWS DMS supports a wide range of sources and targets. Supported AWS DMS targets include Amazon Simple Storage Service (Amazon S3), Amazon DynamoDB, Amazon Redshift, Amazon Kinesis Data Streams, Amazon DocumentDB, and Redis.

You can use AWS DMS to migrate your Exadata workloads to Amazon RDS for Oracle or to an Oracle database on Amazon EC2. AWS DMS handles the initial load as well as change data capture (CDC) updates from Exadata. Exadata is fully operational during the migration process. If you use CDC, the target database remains continuously synchronized with Exadata, so your application cutover can occur at a convenient time.

Native Oracle tools such as Oracle RMAN, Oracle Data Guard, and Oracle Data Pump are more flexible and can load data faster than AWS DMS. If you're migrating large (multi-TiB) Exadata databases, we recommend that you choose these native Oracle utilities instead of AWS DMS for the initial data load.

Oracle Data Pump supports multiple worker processes that can perform inter-table and inter-partition parallelism to load and unload tables in multiple, parallel, or direct-path streams. All import and export processing in Data Pump, including reading and writing dump files, is handled by the server and doesn't involve the client. The Data Pump dump file storage format is the internal stream format of the direct path API. This format is very similar to the format stored in Oracle Database data files inside tablespaces. Therefore, Data Pump doesn't have to perform client-side conversion to `INSERT` statement bind variables. Also, Data Pump supports data access methods, direct path, and external tables, which are faster than conventional SQL. The direct path API provides the fastest single-stream performance. The external tables feature makes efficient use of the parallel queries and parallel DML capabilities of Oracle Database. If your Exadata to Amazon RDS for Oracle migration requires low downtime, a common Exadata migration approach is to use Data Pump for the initial load and then use AWS DMS or Oracle GoldenGate for CDC.

There are limitations when you use Exadata as a source for AWS DMS. For more information about these, see the AWS DMS documentation. Also, network connectivity to the source (Exadata on premises) and target (Oracle database on AWS) is required for AWS DMS.

If you use AWS DMS for the initial load, consider the following best practices:

- You can generally improve performance by selecting a large AWS DMS replication instance. Large tables take longer to load, and transactions on those tables must be cached until the table is loaded. After a table is loaded, these cached transactions are applied and are no longer held on disk. For example, if the load takes five hours and produces 6 GiB of transactions each hour, ensure that 30 GiB of disk space is allocated for cached transactions. When the initial load is complete, before you start CDC, you can modify the AWS DMS replication instance to use a smaller instance.
- For large (multi-TiB) Exadata migrations, we recommend that you use AWS DMS Binary Reader instead of Oracle LogMiner (which is the default). Binary Reader has a lower risk of I/O or CPU impact because logs are mined directly instead of requiring multiple database queries. However, Oracle LogMiner is better when you have a high volume of changes and you're using Oracle ASM. To use Binary Reader to access the redo logs, add the following extra connection attributes for the source endpoint:

  ```
  useLogMinerReader=N;useBfile=Y
  ```

  For a full comparison, see Using Oracle LogMiner or AWS DMS Binary Reader for CDC in the AWS DMS documentation.
- Disable Amazon RDS for Oracle backups or change the archiving mode to `NOARCHIVELOG` if you're migrating to Oracle on Amazon EC2. Enable backups before the CDC phase or after the initial data load.
- Disable all standby databases on AWS. This includes Amazon RDS for Oracle Multi-AZ and read replicas. It also includes Oracle Data Guard or Oracle Active Data Guard standbys if you're migrating to Oracle on Amazon EC2.
- Drop primary key indexes, secondary indexes, referential integrity constraints, and data manipulation language (DML) triggers before initial loads on the target database. Enable these objects before starting the CDC phase.
- For large tables, consider breaking up a single table into multiple AWS DMS tasks by using row filtering, a key, or a partition key. For example, if your database has an integer primary key ID

that ranges from 1 to 8,000,000, create eight tasks by using row filtering to migrate one million records for each AWS DMS task. You can also use this technique with a date column.

- Divide the AWS DMS migration into multiple AWS DMS tasks. Transactional consistency is maintained within a task, so tables in separate tasks should not participate in common transactions.
- By default, AWS DMS loads eight tables at a time. For performance improvements, you can increase this value if you use a large replication server.
- By default, AWS DMS processes changes in a transactional mode, which preserves transactional integrity. Changing to the batch-optimized apply option can improve performance. We recommend that you turn off these constraints during the initial load and turn them back on for the CDC process.
- If the AWS DMS replication instance and the Oracle database on AWS are in different virtual private clouds (VPCs), we recommend that you use VPC peering.
- Enable Amazon CloudWatch logs when you create or modify AWS DMS migration tasks. This parameter is available in the **Task Settings** section when you create an AWS DMS task. Enabling this parameter captures information such as task status, percent complete, elaspsed time, and table statistics during the migration process. For more information, see Monitoring replication tasks using Amazon CloudWatch in the AWS DMS documentation.

For additional best practices, see Using an Oracle database as a source for AWS DMS and Best practices for AWS Database Migration Service in the AWS DMS documentation.

Oracle GoldenGate migrations

Oracle GoldenGate is a logical replication solution. You can use this tool to replicate, filter, and transform data from one database to another. You can move committed transactions across multiple heterogeneous systems and replicate data from Oracle databases to other homogeneous databases and supported heterogeneous databases. Oracle GoldenGate shares many of the positive characteristics and limitations of AWS DMS.

Both tools provide logical replication. However, AWS DMS is a managed service that requires no installation and configuration, whereas Oracle GoldenGate must be installed and configured. You can set it up on premises or on AWS. You can install Oracle GoldenGate on AWS by using a highly available configuration to migrate data from Exadata to AWS. Do not install Oracle GoldenGate directly on Exadata on premises or on an Oracle database node on Amazon EC2; database nodes should be dedicated to processing database workloads.

Another major difference between AWS DMS and Oracle GoldenGate is pricing. AWS DMS charges for replication instance usage and log storage. All data transfers into AWS DMS are free, and data transferred between AWS DMS and databases on Amazon RDS and Amazon EC2 instances in the same Availability Zone are also free. Oracle GoldenGate requires an Oracle GoldenGate license for every core on the source and target databases. You can use Oracle GoldenGate to migrate Exadata workloads to Amazon RDS for Oracle or Oracle on Amazon EC2, for both the initial load and to perform CDC from Exadata. This process allows Exadata to be fully operational during the migration process.

To migrate large (multi-TiB) Exadata databases to Oracle on Amazon EC2, consider using Oracle RMAN, Oracle Data Guard, or Oracle Data Pump instead of Oracle GoldenGate for the following reasons:

- Oracle GoldenGate requires network connectivity between Exadata and AWS.
- Oracle GoldenGate doesn't perform as well as other Oracle migration tools for the initial data load. For example, to migrate large Exadata databases to Amazon RDS for Oracle, consider using Oracle Data Pump instead, because it's more flexible and can load data faster than Oracle GoldenGate.

If your Exadata to Amazon RDS for Oracle migration requires low downtime, a common migration approach is to use Oracle Data Pump for the initial load and Oracle GoldenGate or AWS DMS for CDC. The advantage of Oracle GoldenGate is that it can handle the initial load as well as CDC. CDC allows the target database to remain continuously synchronized with Exadata, so you can switch over at a convenient time.

There are limitations when you use Exadata as a source with Oracle GoldenGate. For information about these, see Understanding What's Supported in the GoldenGate documentation.

If you use Oracle GoldenGate for the initial load, consider the following best practices:

- Use Extract in integrated capture mode to take advantage of the integration with the LogMiner server. Integrated capture allows seamless extraction of more data types than with Extract in classic mode. These additional data types include compressed data, including Basic Compression, online transaction processing (OLTP), and Exadata Hybrid Columnar Compression (HCC). There is no additional configuration required for Extract to read log files that are stored on Oracle ASM.
- Use Integrated Replicat. This option uses the database apply process. It maintains referential integrity and automatically applies DDL operations. Integrated Replicat also offers automatic

parallelism, which automatically increases or decreases based on the current workload and database performance.

- Set `BATCHSQL` in the Replicat parameter file. By default, Integrated Replicat tries to reorder and group DML statements of the same type against the same object within each transaction. Using batches can reduce the CPU and run time of DML statements.
- Configure the GoldenGate heartbeat table to provide end-to-end replication lag views. This enables you to see the end-to-end replication latency by viewing the `GG_LAG` database view.
- Disable Amazon RDS for Oracle backups or change the archiving mode to `NOARCHIVELOG` if you're using Oracle on Amazon EC2. Enable backups before the CDC phase or after the initial data load.
- Disable all standby databases on AWS. This includes Amazon RDS for Oracle Multi-AZ and read replicas. It also includes Oracle Data Guard or Oracle Active Data Guard standbys if you're migrating to Oracle on Amazon EC2.
- Drop primary key indexes, secondary indexes, referential integrity constraints, and data manipulation language (DML) triggers before initial loads on the target database. Enable these objects before starting the CDC phase.
- If the Oracle GoldenGate replication instance and the Oracle database on AWS are in different virtual private clouds (VPCs), we recommend that you use VPC peering.

Oracle Data Pump migrations

You can use Oracle Data Pump to move data from one Oracle database to another. Data Pump provides a wide range of benefits, such as supporting older releases of Oracle Database (back to version 10.1) and supporting platforms that have different formats, database architectures, and versions. You can choose to export your full database or only specific schemas, tablespaces, or tables.

You can control the degree of parallelism, compression, and encryption, and specify which objects and objects types to include or exclude. Data Pump also supports network mode, where you can transfer data by using a database link without the need for intermediate storage.

The Data Pump API provides a fast and reliable way to move data and metadata between Oracle databases. The Data Pump Export and Data Pump Import utilities are based on the Data Pump API. An Amazon RDS for Oracle instance can't be accessed through the Secure Shell (SSH) protocol, so the Data Pump API is the only way to import data if you use Data Pump to migrate from Exadata

to Amazon RDS for Oracle. The Data Pump Command Line Interface (CLI) is not an option for migrating to Amazon RDS for Oracle.

If you use Data Pump for the initial load, consider the following best practices:

- Create the required tablespaces before you import the data.
- If you want to import data into a user account that doesn't exist, create the user account and grant the necessary permissions and roles.
- If you're migrating to Oracle on Amazon EC2, turn off Amazon RDS for Oracle backups or change the archiving mode to `NOARCHIVELOG`. Activate backups before you start the CDC phase or after the initial data load.
- Turn off all standby databases on AWS. This includes Amazon RDS for Oracle Multi-AZ and read replicas. It also includes Oracle Data Guard or Oracle Active Data Guard standbys if you're migrating to Oracle on Amazon EC2.
- Drop primary key indexes, secondary indexes, referential integrity constraints, and DML triggers before initial loads on the target database. Activate these objects before you start the CDC phase.
- To import specific schemas and objects, perform imports in schema or table mode.
- Limit the schemas you import to those that your application requires.
- Load and unload data in parallel by using compression and multiple threads.
- Files in Amazon S3 must be 5 TiB or less. Use the `PARALLEL` option to create multiple Data Pump dump files to avoid this limitation.
- If you're planning to perform CDC after the Data Pump export, use the Oracle system change number (SCN) with Data Pump.
- If you want to load data to Amazon RDS for Oracle, perform these tasks:
  1. Create an AWS Identity and Access Management (IAM) policy to allow Amazon RDS access to an S3 bucket.
  2. Create an IAM role and attach the policy.
  3. Associate the IAM role with the Amazon RDS for Oracle instance.
  4. Configure an Amazon RDS for Oracle option group for Amazon S3 integration and add it to the Amazon RDS for Oracle instance.

  For additional information, see Amazon S3 integration in the Amazon RDS documentation.

Oracle RMAN migrations

Oracle Recovery Manager (RMAN) is a tool for backing up and recovering an Oracle database. It is also used to facilitate database migrations on premises and between on-premises and cloud databases.

Oracle RMAN provides a physical migration approach. For this reason, it supports rehosting (migration to Amazon EC2) but can't be used to replatform your Oracle Database on Amazon RDS for Oracle. Your migration downtime tolerance must be large enough to back up and restore an Oracle RMAN incremental backup.

Migrating to Amazon S3

To back up your Exadata database to Amazon S3, you can use the following options:

- Use the Oracle Secure Backup (OSB) Cloud Module to back up your Exadata database directly to Amazon S3.
- Copy the Oracle RMAN backup sets to Amazon S3 from the Exadata RMAN backup location.
- Use Oracle ZFS Storage Appliances. Oracle RMAN backup sets that are stored on Oracle ZFS Storage Appliances can be transferred directly to Amazon S3 by using the Oracle ZFS Storage Appliance S3 Object API Service.
- Store Oracle RMAN backups directly on the Exadata Storage Server, Oracle Zero Loss Recovery Appliance, and tape libraries. You can then transfer the RMAN backup sets on any of these storage platforms to Amazon S3.

Migrating to Amazon EC2

You can also use RMAN to back up your Exadata database directly to Oracle Database on Amazon EC2 without creating backup sets. To do this, use the Oracle RMAN `DUPLICATE` command to perform a backup and restore. However, Oracle RMAN `DUPLICATE` isn't recommended for large (multi-TiB) Exadata migrations.

RMAN settings are usually configured based on factors such as the backup size, the Exadata CPU, compression, and the parallelism or number of RMAN channels. Using Oracle Service Bus (OSB) and compression (low, medium and high) with RMAN requires Oracle Advanced Compression Option (ACO) licenses. OSB also requires Oracle licenses that are based on the number of RMAN channels that you want to use with OSB.

If you want to use RMAN to migrate Exadata to Oracle on Amazon EC2, consider the following best practices.

> **Note**
>
> The commands provided in this section must be run on the Oracle on Amazon EC2 instance.

- If you want to use different Oracle ASM disk group names on Amazon EC2, run the `set newname` command with the RMAN restore process:

```
set newname for datafile 1 to '+<disk_group>'; set newname for datafile 2 to
 '+<disk_group>';
```

- If the online redo logs will reside in a different location on AWS, rename the redo log files:

```
alter database rename file '/<old_path>/redo01.log' to '+<disk_group>';
alter database rename file '/<old_path>/redo02.log' to '+<disk_group>';
```

- After you open the database successfully on AWS:
  - Remove the redo log groups for redo threads of other instances:

```
alter database disable thread 2;
alter database drop logfile group 4;
alter database clear unarchived logfile group 4;
```

  - Remove the undo tablespaces of other instances:

```
drop tablespace UNDOTBS2 including contents and datafiles;
```

  - Make sure that only one `TEMP` tablespace exists. Remove unnecessary `TEMP` tablespaces and confirm that the existing `TEMP` tablespace is large enough to handle the anticipated database workload.

HCC considerations

If you use Hybrid Columnar Compression (HCC) in Exadata, all tables with HCC must be converted to Oracle ACO or disabled on AWS. Otherwise, SQL statements will fail when you access your Oracle database on Amazon EC2. Oracle ACO requires an Oracle license.

Typically, users can't remove HCC from an on-premises Exadata production database. You can remove HCC when you migrate your database to AWS. To determine whether HCC is activated on a table or partition after you migrate your database to AWS, run the following SQL statement:

```
select TABLE_NAME, COMPRESSION, COMPRESS_FOR
from DBA_TABLES
where OWNER like 'SCHEMA_NAME';

select TABLE_NAME, PARTITION_NAME, COMPRESSION, COMPRESS_FOR
from DBA_TAB_PARTITIONS
where TABLE_OWNER = 'SCHEMA_NAME';
```

If the `compression` column value is set to `ENABLED` and the `compress_for` column has one of the following values, HCC is enabled:

- `QUERY LOW`
- `QUERY HIGH`
- `ARCHIVE LOW`
- `ARCHIVE HIGH`
- `QUERY LOW ROW LEVEL LOCKING`
- `QUERY HIGH ROW LEVEL LOCKING`
- `ARCHIVE LOW ROW LEVEL LOCKING`
- `ARCHIVE HIGH ROW LEVEL LOCKING`
- `NO ROW LEVEL LOCKING`

To turn off HCC on a table or partition, run the following SQL statement:

```
alter table table_name nocompress;
alter table table_name modify partition partition_name nocompress;
```

To activate Oracle ACO on AWS, follow the instructions in the Oracle documentation.

Oracle Data Guard migrations

Oracle Data Guard enables you to create and manage one or more standby databases for high availability and disaster recovery. Data Guard maintains standby databases as copies of the primary (typically production) database. If the production database encounters planned or unplanned

availability issues, Data Guard can switch roles to ensure minimal downtime and application continuity.

You can use both logical standby and physical standby methods to implement Data Guard. In this guide, we assume that you're using a physical standby database that exactly matches the primary database.

Data Guard supports migrations from Exadata to Oracle Database on Amazon EC2 to create a physical standby. It can't be used to migrate to Amazon RDS for Oracle, which requires logical migration approaches such as AWS DMS, Oracle Data Pump, or Oracle GoldenGate.

Data Guard is a simpler and faster approach for migrating an entire Exadata database compared with a CDC mechanism such as AWS DMS or Oracle GoldenGate. It is usually the recommended approach if you have minimal downtime requirements (for example, you have time only for a switchover).

You can configure Data Guard with synchronous or asynchronous transport. In general, Oracle customers have greater success with synchronous transport when round trip network latency is less than 5 ms. For asynchronous transport, Oracle recommends round trip network latency that's less than 30 ms.

Typically, a Data Guard standby would already exist for the production Exadata on-premises database. Oracle on Amazon EC2 usually serves as an additional standby database for the production Exadata on-premises database. We recommend that you create the Data Guard standby database on AWS by using Oracle RMAN.

There are many variables that affect Data Guard performance. We recommend that you perform testing before you draw any conclusions on the impact of Data Guard replication on your workload.

Latency (measured through a ping monitor) isn't significant for Data Guard replication, because the mechanism used is different. The Oracle **oratcptest** utility helps assess network resources. You can download **oratcptest** in JAR format from My Oracle Support (MOS) Note 2064368.1 (requires an Oracle account). The MOS note also provides more information about this utility.

AWS sample migration patterns

Let's say that you have a 50 GiB Exadata database that must be replatformed on AWS (migrated to Amazon RDS for Oracle). The migration approach you use would depend on factors such as your downtime tolerance, your connection method, and your database size.

The following table provides examples of the most effective migration approaches based on key factors. The migration approach that best meets your needs depends on the specific combination of these factors.

Source database	Target database	Database size	Migration downtime tolerance	Networking to AWS	Best migration approach
Exadata 12c	Amazon RDS for Oracle 19c	1 TiB	48 hours	1 Gbps AWS Direct Connect	Use Oracle Data Pump.
Exadata 12c	Amazon RDS for Oracle 21c	5 TiB	2 hours	10 Gbps AWS Direct Connectt	Use Oracle Data Pump for the initial load and AWS DMSfor CDC.
Exadata 19c	Oracle 19c on Amazon EC2	10 TiB	72 hours	10 Gbps AWS Direct Connect	Use Oracle RMAN.
Exadata 19c	Oracle 19c on Amazon EC2	70 TiB	4 hours	1 Gbps AWS Direct Connect	Use AWS Snowball to transfer RMAN backups, archived redo log files, and control files to AWS. Instantiate the Oracle Data Guard standby database on Amazon EC2

Source database	Target database	Database size	Migration downtime tolerance	Networking to AWS	Best migration approach
					from Exadata RMAN backups. Perform a Data Guard switchover after the standby database has been configured on Amazon EC2 and is in sync.
Exadata 19c	Amazon RDS for PostgreSQL 13.4	10 TiB	2 hours	10 Gbps AWS Direct Connect	Use the AWS Schema Conversion Tool (AWS SCT) to create PostgreSQL schemas. Use AWS DMS for both full load and CDC.

Exadata-specific feature considerations

Exadata has proprietary software that runs on storage cells to improve query performance, lower redo log latency, compress data, and improve other database operations. Many of these features aren't available for an Oracle database on AWS. We recommend that you consider performing

the tasks that are discussed later in this section to achieve equivalent performance and similar functionality.

You can disable Exadata functionality on non-production Exadata systems to get a baseline of how the database would perform without this functionality. You can compare this baseline to the first performance test on AWS for a realistic comparison.

The following instructions describe how to disable Exadata functionality on an existing Exadata system. We recommend that you perform these steps in a non-production environment to capture a baseline of how a non-Exadata database will perform.

- **To disable Exadata Storage Server cell offload processing:** The mechanism depends on the scope of the change (statement-level, session-level, or database-level).
  - For a SQL statement, use the following SQL hint:

```
select /*+ OPT_PARAM('cell_offload_processing' 'false') */ max(ORDER_DATE)
from SALES;
```

  - For an Oracle session, set the following Oracle database initialization parameter:

```
alter session set CELL_OFFLOAD_PROCESSING=FALSE;
```

  - For the entire Exadata database, set the following Oracle database initialization parameter:

```
alter system set CELL_OFFLOAD_PROCESSING=FALSE;
```

- **To disable Exadata storage indexing:** To turn off Exadata storage indexing for the entire Exadata database, set the following Oracle database initialization parameter:

```
alter system set KCFISSTORAGEIDX_DISABLED=TRUE scope=both;
```

- **To disable decryption offload to Exadata Storage Server:** By default, the decryption of both encrypted tablespaces and encrypted columns are offloaded to Exadata Storage Server. To disable decryption offload to Exadata Storage Server, run the following command:

```
alter system set CELL_OFFLOAD_DECRYPTION=FALSE;
```

- **Smart Flash Cache:** Oracle doesn't recommend turning off Exadata Smart Flash Cache unless directed by Oracle Support or Oracle Development.

In agile product development, a sprint is a set period of time during which specific work has to be completed and made ready for review. After you migrate your Exadata database to AWS and complete three or four sprints, it is not uncommon for IOPS to be reduced by 30-70 percent. Additionally, storage throughput could be reduced by up to 90 percent of the Exadata-reported value. As previously mentioned, you can test IOPS and throughput on an Exadata non-production system that is a copy of the Exadata production system. You can turn off Exadata Storage Server cell offload processing, Exadata Storage Server decryption, and Exadata storage indexes. Additionally, you might have to complete the following on the Exadata non-production system after you migrate Exadata to AWS:

- Add indexes to improve unindexed queries. If indexes were changed to invisible, you might have to make them visible by using an `ALTER INDEX` statement. Each index requires maintenance for insert, update, and delete statements.
- Rewrite queries that can't be improved with indexes.
- Determine if you can run some SQL statements less frequently.

After several development sprints, an AWS customer who moved their Exadata system to Amazon EC2 on AWS reported the following results, based on the averages across Oracle Automatic Workload Repository (AWR) snapshots. The Oracle database on AWS performed on average 220 percent better than the Exadata on-premises database, although the peak IOPS and peak throughput (MBps) was lower. Also, the AWS database had only 20 percent of the cores compared with Exadata on premises.

Environment	Peak IOPS	Peak throughput (MBps)
Exadata on premises	201,470	62,617
Oracle on Amazon EC2	66,420	4,640

Homogeneous database migration considerations

This section discusses key best practices for homogeneous migrations. When you migrate your database from Exadata on premises to Amazon RDS for Oracle or Oracle on Amazon EC2, consider the guidelines that are discussed in the following subsections.

Encryption

Data security is top priority at AWS. AWS has implemented rigorous contractual, technical, and organizational measures to protect customers' confidentiality, integrity, and availability. For databases, encryption is critical because it protects private information and sensitive data. Oracle on Amazon EC2 and Amazon RDS for Oracle support two encryption methods for data at rest:

- AWS Key Management Service (AWS KMS) to encrypt Amazon EBS volumes.
- Oracle Advanced Security Option Transparent Data Encryption (TDE) to encrypt sensitive information that is stored in data files. Oracle TDE requires an Oracle license.

Both options encrypt user data in the Oracle database and in all database backups. Encryption is also transparent to the DML statements issued from applications.

For data in transit, Oracle on Amazon EC2 and Amazon RDS for Oracle support Oracle Native Network Encryption (NNE). For more information about NNE support, see the Amazon RDS documentation.

Data partitioning

With Oracle Partitioning, a single logical object in the database, such as a table or an index, is divided into smaller physical database objects, which helps improve manageability, performance, and availability. Oracle Partitioning requires an Oracle license.

If you have large database workloads, consider partitioning your tables. Partition pruning enables the Oracle Database optimizer to analyze `FROM` and `WHERE` clauses in SQL statements to eliminate unneeded partitions when building the partition access list. Oracle Database performs operations only on the partitions that are relevant to the SQL statement, which typically improves performance.

Partitioning also helps with availability. If a partition goes offline and a SQL statement doesn't need the offline partition in order to complete an operation, the SQL statement will succeed. However, if a data block is lost within an Oracle Database table that hasn't been partitioned, the entire table will be unavailable until the restore operation is complete.

Data compression

For data compression, Oracle offers both HCC and Advanced Compression. Advanced Compression improves performance and reduces storage costsby reducing the database storage footprint for

relational data (tables), unstructured data (files), indexes, Data Guard redo data, network data, RMAN backups, and other types of data. Advanced Compression can also improve the performance of database infrastructure components, including memory and network bandwidth.

According to Oracle documentation, Advanced Compression has an average compression rate of at least 2x. Therefore, 100 GiB of data can typically reside in 50 GiB of storage space. When you migrate your Oracle database to AWS, you can use Advanced Compression inAmazon RDS for Oracle and Oracle on Amazon EC2, with both OLTP and data warehousing databases. You can consider using Advanced Compression with your Oracle database on AWS to improve performance and lower Amazon EBS storage costs even if you didn't use it with Exadata. Advanced Compression requires an Oracle license.

ILM strategy

Information Lifecycle Management (ILM) provides processes, policies, and components that help manage the information in a database based on its usage frequency. When you migrate from Exadata to Oracle on AWS, you should determine whether you can purge any data before or after migrating it to AWS. On AWS, you can apply rules to maintain data for a specific period of time only. You can implement Oracle Partitioning and Oracle Advanced Compression to set up data lifecycle policies. This can improve performance while maintaining only the data that is required to support your business.

For example, let's say you have a table that consumes multiple tebibytes of uncompressed data. You currently have 12 years of data, and you must keep data for 14 years. About 90 percent of all queries access data that is less than two years old. You typically compare data usage month over month, quarter over quarter, and year over year. Data cannot be updated after 30 months, but you sometimes have to access historical data that's up to 12 years old. In this case, you might consider the following ILM policies:

- Implement Advanced Compression. Take advantage of Oracle Heat Map and Automatic Data Optimization (ADO) with Advanced Compression.
- Set up interval partitioning on the date column.
- Use a function that drops partitions that are older than 14 years on a monthly basis.
- Use read-only tablespaces to hold data that's more than 30 months old. The primary purpose of read-only tablespaces is to eliminate the need to perform backup and recovery of large, static portions of a database (when you use Oracle RMAN with Oracle on Amazon EC2). Read-only tablespaces also provide a way to protect historical data so that users cannot modify it. Making

a tablespace read-only prevents updates on all tables in the tablespace, regardless of a user's update privilege level.

Users often store active data, infrequently accessed data, and archive data in a single Oracle database. During your Oracle database migration to AWS, you can migrate infrequently accessed data, historical audit data, and archive data directly into Amazon S3 or Amazon S3 Glacier. This helps you meet your governance and compliance needs for long-term data retention without impacting database performance. As data ages in the relational database, it can be archived to Amazon S3 or Amazon S3 Glacier. You can easily query the archived data by using Amazon Athena or Amazon S3 Glacier Select.

OEM integration

When you migrate your Oracle workloads to AWS, you might want to implement Oracle Enterprise Manager (OEM) Cloud Control on AWS. OEM is Oracle's management platform that provides a single interface for managing Oracle environments.

Oracle on Amazon EC2 and Amazon RDS for Oracle can be targets for an OEM environment. Oracle on Amazon EC2 follows the same process as Oracle on premises to integrate with OEM. To activate OEM on Amazon RDS for Oracle:

1. Sign in to the AWS Management Console and open the Amazon RDS console at https://console.aws.amazon.com/rds/.
2. In the navigation pane, choose **Option Groups**.
3. Add the `OEM_AGENT` option to a new or existing option group.
4. Add OEM configuration information, including the OEM management server hostname, port, and OEM agent registration password.

Amazon RDS for Oracle and Oracle on Amazon EC2 can also be targets for an OEM environment that's running on premises. However, this requires all OEM ports to be accessible through your firewall.

Amazon CloudWatch integration

Amazon CloudWatch collects monitoring and operational data in the form of logs, metrics, and events. It visualizes data by using automated dashboards that provide a unified view of AWS

resources, applications, and services that run on AWS and on premises. Oracle databases that are hosted on Amazon EC2 and Amazon RDS for Oracle can use CloudWatch.

CloudWatch and Amazon Simple Notification Service (Amazon SNS) are integrated so you can collect, view, and analyze metrics for every active Amazon SNS notification. For example, you can set an alarm to send an email notification or SMS if a specified action, such as a specific Oracle error message in the Oracle Database alert log, occurs.

To use CloudWatch and Amazon SNS with Oracle on Amazon EC2, you must install a CloudWatch agent to push the Oracle alert log, audit logs, trace logs, OEM logs, and listener logs to CloudWatch. If you deploy Amazon RDS for Oracle, you must modify the Oracle instance to enable these logs to be sent to CloudWatch. For more information about CloudWatch integration, see Monitoring Amazon SNS topis using CloudWatch in the Amazon SNS documentation.

Amazon RDS for Oracle also has built-in CloudWatch alarms for dozens of events, including CPU utilization, number of database connections, available memory, free storage space, storage IOPS, disk throughput, and replication lag.

Most users who migrate from Exadata on premises to AWS continue to use OEM and also integrate CloudWatch with their Oracle databases on AWS.

Database optimizer statistics

Oracle Database optimizer statistics provide information about the database and its tables, columns, indexes, and the system. The optimizer uses this information to estimate the number of rows and bytes that are retrieved from a table, partition, or index for a query, to estimate the cost of access, and to pick the SQL execution plan that has the lowest cost.

If you restore an Exadata on-premises database to Amazon EC2 through Oracle RMAN, Oracle automatically provides statistics that reflect the Exadata environment. As soon as you restore the Exadata databases to Amazon EC2 or the initial load is completed in Amazon RDS for Oracle, it is best practice to gather statistics as soon as possible. This can be accomplished by running the Oracle DBMS_STATS package.

AWR settings

The Oracle Automatic Workload Repository (AWR) stores performance-related statistics for an Oracle database. By default, Oracle Database generates snapshots once every hour, and retains the snapshots for 8 days. You can manually create or drop snapshots, and modify snapshot settings.

For production Oracle databases, you should increase the AWR retention period to 60 or 90 days and reduce the AWR interval to 15 or 30 minutes. These settings support month-over-month comparisons and provide more granularity when you view AWR data. These changes consume relatively small (measured in gibibytes) database space and provide the benefits of additional history. To set the AWR retention period to 60 days and the AWR interval to 15 minutes, run the following command (parameter values are in minutes):

```
BEGIN
DBMS_WORKLOAD_REPOSITORY.modify_snapshot_settings
        (interval => 15,
         retention => 86400
        );
END;
/
```

If you migrate your Exadata on-premises database to Oracle on Amazon EC2 by using Oracle RMAN or Oracle Data Guard, you should drop the AWR snapshots captured while the database was running on Exadata. To do this, use the `DBMS_WORKLOAD_REPOSITORY.DROP_SNAPSHOT_RANGE` procedure on AWS.

Oracle RAC considerations

By default, Exadata uses Oracle Real Application Clusters (RAC), which enable you to run a single Oracle database across multiple servers in order to maximize availability and enable horizontal scalability. Oracle RAC uses shared storage. The smallest Exadata offering includes two nodes that are configured by using Oracle RAC.

If you have an RPO requirement of zero and an RTO requirement of two minutes or less, you can implement Amazon RDS for Oracle with Multi-AZ. This configuration provides a monthly uptime commitment of 99.95%, which is equivalent to or better than any managed Oracle cloud database in the industry, including managed Oracle databases that use Oracle RAC.

Additionally, Oracle on Amazon EC2 lets you implement a highly available database by using many of the components in the Oracle Maximum Availability Architecture (MAA). These components include, but are not limited to, Active Data Guard, RMAN, Flashback Technologies, Edition-Based Redefinition, and GoldenGate.

There are also various alternatives for implementing Oracle RAC on AWS. To find out more about RAC options on AWS, we recommend that you contact your AWS account team.

Additional best practices for homogeneous migrations

Developers often ignore SQL tuning techniques and best practices when they implement Exadata. Exadata hides many design issues, so SQL statements might get deployed into production without assessing their execution plans or resource consumption, because they complete within acceptable elapsed times. Follow these additional practices when you migrate your Exadata on-premises database to Oracle on AWS.

- Apply the latest Oracle Release Update (RU) or Release Update Revision (RUR).
- Make sure that the `COMPATIBLE` initialization parameter contains only three levels (for example, 19.0.0). If an upgrade takes place after you migrate to AWS, make sure that this parameter is modified during the upgrade process.
- Consider caching sequence numbers to minimize I/O. The default value is 20. If there is insufficient caching of sequence numbers, contention can occur, which will show up as an increase in service times for DML.
- If you use sequences, validate the sequence values against the source database (Exadata on premises) to avoid sequence inconsistency.
- If connection pooling isn't implemented on the application tier or the number of application tiers results in a very large number of database connections, consider implementing Oracle Database Resident Connection Pooling (DRCP). This feature handles memory and compute resources on the database server efficiently.
- Consider using HugePages. Oracle recommends that you use standard HugePages for Linux. Enabling HugePages makes it possible for the operating system to support memory pages that are greater than the default (usually 4 KB). Using very large page sizes can improve system performance by reducing the amount of system resources required to access page table entries.
- If the Oracle database on AWS has database links, confirm that the `OPEN_LINKS` and `OPEN_LINKS_PER_INSTANCE` initialization parameters aren't set to the default value (4). If this value is too low, SQL statements that have database links begin to queue when the maximum value is reached, which negatively impacts performance.
- The initial data load might not be able to be transmitted over the network. For example, theoretically it takes at least nine days with no interruptions to transfer 100 TiB over a 1 Gbps link. A better approach would be to use an AWS Snow Family device to migrate the database to AWS.

- Remove any Exadata-specific hidden parameters (see Oracle MOS Note 1274318.1). These hidden Exadata initialization parameters shouldn't be activated on AWS. They can cause instability, performance problems, corruption, and crashes.
- Try to resolve all non-SYS and SYSTEM invalid objects after you migrate the data to Oracle on AWS.
- Consider caching static, frequently accessed tables in the Oracle System Global Area (SGA).
- Choose memory optimized instances with larger Oracle SGA configurations to mitigate the challenge of additional I/O on AWS. You can use the Oracle SGA Advisory report during load testing in the target instance to find the optimal Oracle SGA configuration.
- Create indexes on tables that handle many full table scans. The V$SEGMENT_STATISTICS view lists candidate segments.
- Identify top resource-intensive queries and optimize them for better execution plans. Oracle SQL Tuning Advisor, which is licensed under the Oracle Tuning Pack, can be useful for automatic SQL tuning. In some cases, you might need to rewrite queries or break down a complex query into smaller chunks.
- Consider implementing caching solutions such as Amazon ElastiCache and Amazon RDS for Oracle read replicas, such as Oracle Active Data Guard, to serve read-only workloads.
- Train your developers in query optimization techniques, and build standard operating procedures to assess queries before they are deployed to production.
- Make sure that the database object count in AWS is the same as in the Exadata on-premises database. Validate tables, indexes, procedures, triggers, functions, packages, constraints, and other objects.
- Consider application modifications if possible. (In some cases, applications can't be modified as with packaged ISV applications.) Avoid unnecessary calls and try to reduce the frequency of required calls. Try to minimize the data volume retrieved by SQL statements. Make sure that the commit frequency is appropriate for the business logic, but not excessive. Try to improve the use of application-level caching.
- The database should reside in a private virtual private cloud (VPC) on AWS. Restrict network access for inbound and outbound traffic to a least privilege model. The security group source should refer to a security group in the AWS account, prefix lists, or a specific set of IP addresses (using the x.x.x.x/32 format). The security group source shouldn't use CIDR and security groups shouldn't be accessible from the public internet (0.0.0.0/0).

Replatforming recommendations

Most users choose Amazon RDS for Oracle when they migrate from an Exadata on-premises database to take advantage of a managed database service and to improve agility and elasticity. Amazon RDS for Oracle should always be your first option for running Oracle databases on AWSbecause of its automation and management features.

Amazon EBS volume type considerations

Amazon RDS for Oracle offers two EBS volume types: General Purpose solid state drive (SSD) and Provisioned IOPS SSD. Your database size, IOPS requirements, and estimated throughput help you determine the appropriate EBS volume type to use.

When your applications don't need high storage performance, you can use General Purpose SSD (gp2) storage. Baseline I/O performance for gp2 storage is 3 IOPS for each GiB, with a minimum of 100 IOPS. This means that larger volumes have better performance. For example, baseline performance for one 100 GiB volume is 300 IOPS. Baseline performance for one 1,000 GiB volume is 3,000 IOPS. Maximum baseline performance for one gp2 volume (5334 GiB and greater) is 16,000 IOPS. Individual gp2 volumes below 1,000 GiB in size also have the ability to burst to 3,000 IOPS for extended periods of time.

General Purpose SSD (gp3) volumes support a maximum of 16,000 IOPS per EBS volume. An Amazon EBS gp3 volume can range in size from one GiB to 16 TiB. When you use gp3 volumes, you can achieve a maximum of 64,000 IOPS for your Amazon RDS for Oracle instance. By using gp3 storage volumes, you can customize storage performance independently of storage capacity.*Storage performance* is the combination of I/O operations per second (IOPS) and how fast the storage volume can perform read and write operations (storage throughput). On gp3 storage volumes, Amazon RDS provides a baseline storage performance of 3,000 IOPS and 125 MiB/s.

For every Amazon RDS DB engine except for Amazon RDS for SQL Server, when the storage size for gp3 volumes reaches a certain threshold, the baseline storage performance increases to 12,000 IOPS and 500 MiB/s. This is because of *volume striping*, where the storage uses four volumes instead of one.

Provisioned IOPS SSD volumes

Provisioned IOPS SSD (io1) volumes are designed to meet the needs of I/O-intensive workloads that are sensitive to storage performance and consistency. Amazon EBS io1 volumes deliver single-

digit millisecond latencies. When you select Amazon EBS io1 volumes for Amazon RDS for Oracle, you have to provide the allocated storage value and the provisioned IOPS value. An io1 volume can range in size from 4 GiB to 16 TiB. The maximum IOPS per io1 volume is 64,000. When you use io1 volumes, you can achieve a maximum of 256,000 IOPS and a maximum throughput of 4 Gbps (requires 256 KB IOPS) for the Amazon RDS for Oracle instance. The maximum write throughput for an Amazon RDS for Oracle instance with Multi-AZ enabled is 625 MBps.

io2 Block Express is a newer Provisioned IOPS SSD storage option. An io2 volume can range in size from 4 GiB to 64 TiB. The maximum IOPS per io2 volume is 256,000. io2 Block Express also provides a sub-millisecond average latency and therefore outperforms io1. When using Provisioned IOPS SSD storage, io2 is the recommended option to use. You can upgrade from io1 volumes to io2 Block Express volumes without any downtime, and significantly improve the performance and reliability of your applications without increasing storage costs. For more information, see the AWS blog post Amazon RDS now supports i02 Block Express volumes for mission-critical database workloads.

Amazon RDS for Oracle best practices

Consider the following best practices when you migrate from Exadata on premises to Amazon RDS for Oracle:

- Before you migrate data from Exadata to Amazon RDS for Oracle, increase the size of the redo logs from the default value of 128 MB. Otherwise, redo log switching might occur too often and cause performance degradation.
- Enable Performance Insights (which has a default 7-day data retention period) after the initial data load.
- Set up Multi-AZ for the production database after the initial data load.
- Integrate Amazon RDS for Oracle with Amazon CloudWatch (at a minimum, use alert logs, listeners, and OEM agent) after the initial data load.
- Install the Oracle Enterprise Manager (OEM) Agent in the associated Amazon RDS for Oracle option group. This requires a functional OEM that already exists on AWS or on premises. You can set up OEM in a highly available mode on AWS.
- Implement Amazon RDS alarms for the following to notify administrators before a maximum capacity is breached:
  - CPU utilization, write IOPS, read IOPS, write throughput
  - Read throughput, freeable memory, swap usage

- Amazon RDS uploads transaction logs for DB instances to Amazon S3 every five minutes. To see the latest restorable time for a DB instance, use the AWS CLI describe-db-instances command and look at the value returned in the `LatestRestorableTime` field for the DB instance. Amazon RDS can upload transaction logs more frequently if your point-in-time recovery requirement is less than five minutes. To change the default value, modify the `ARCHIVE_LAG_TARGET` initialization parameter in the associated Amazon RDS for Oracle parameter group. You can set this parameter's value to 60, 120, 180, 240, or 300 seconds. However, there are tradeoffs if you set a lower value: More redo log files will be generated and log file switching will occur more often.
- Implement Oracle Unified Auditing, which is Oracle's recommended audit framework, in mixed mode. By default, unified auditing isn't enabled on Amazon RDS (`AUDIT_TRAIL=NONE`). You can enable it by setting `AUDIT_TRAIL=DB` or `AUDIT_TRAIL=DB, EXTENDED`. For more information, see the AWS blog post Security auditing in Amazon RDS for Oracle: Part 1.
- To protect against internal threats, configure database activity streams if applicable. This feature works with Oracle unified auditing and provides a near real-time stream of all audited statements (`SELECT`, `DML`, `DDL`, `DCL`, `TCL`) that run in the DB instance. The audit data is collected from the unified database audit location, whereas the storage and processing of database activity is managed outside the database in Amazon Kinesis Data Streams. For more information, see the AWS blog post Security auditing in Amazon RDS for Oracle: Part 2.
- If you prefer standard auditing, you can integrate audit statements with Amazon CloudWatch after the initial data load. When you enable standard auditing by setting the `AUDIT_TRAIL` parameter to `OS`, `XML`, or `XML, EXTENDED`, Amazon RDS for Oracle generates audit records that are stored as `.AUD` or `.XML` operating system files in the Amazon RDS for Oracle instance. These audit files are typically retained in the Amazon RDS for Oracle instance for seven days. You can configure Amazon RDS for Oracle to publish these files to CloudWatch, where they can perform real-time analysis of the log data, store the data in highly durable storage, and manage the data with the CloudWatch log agents. AWS retains log data published to CloudWatch logs for an indefinite period in the AWS account unless you specify a retention period.

Rehosting recommendations

When you rehost Oracle on Amazon EC2, you install and configure the Oracle database and perform all maintenance operations, including minor Oracle upgrades, major Oracle upgrades, operating system patching, operating system configuration, database configuration, memory allocation, storage allocation, and storage configuration.

Amazon EC2 instance type considerations

The EC2 instance must have adequate CPU, memory, and storage to handle the anticipated database workload. We recommend that you use a current generation EC2 instance class for the Oracle database. These instance types, such as instances built on the Nitro System, support Hardware Virtual Machine (HVM). HVM Amazon Machine Images (AMIs) are required to take advantage of enhanced networking, and they also offer increased security.

The virtualized instances built on the Nitro System include R5b, X2idn, and X2iedn. For high Amazon EBS volume throughput, consider Amazon EC2 R5b and X2 instance types. These instances support up to 260,000 IOPS. The maximum throughput for an Amazon EC2 R5b instance is 7,500 MBps. The maximum throughput for Amazon EC2 X2idn and X2iedn instances is 10,000 MBps. For more information, review Amazon EBS-optimized instances and maximum IOPS in the Amazon EC2 documentation.

Amazon EBS volume type considerations

Amazon EBS General Purpose (gp3) volumes are less expensive than Amazon EBS Provisioned IOPS (io2) volumes. If gp3 volumes meet your I/O and throughput requirements, they should be your preferred solution. A single gp3 volume cannot exceed 16,000 IOPS per volume. You must also consider the maximum number of EBS volumes that can be assigned to the EC2 instance. This number varies based on the EC2 instance type; however, the maximum number of EBS volumes for a Nitro System instance is 28. Typically, no more than 24 EBS volumes should be dedicated for the Oracle database.

If your disk I/O requirements are high, consider Amazon EBS io2 Block Express volumes. These are designed to offer up to 4,000 MBps throughput per volume, 256,000 IOPS per volume, 64 TiB storage capacity, sub-millisecond latency, and 99.999% durability. We recommend that you use Amazon EBS io2 Block Express volumes in the following scenarios:

- The database allocated space exceeds 384 TiB. This includes, but is not limited to, database files, redo logs, `TEMP` space, `UNDO` space, Flashback Recovery Area space, and the data staging area. Amazon EBS io2 Block Express volumes can support up to 1.536 PiB with a single EC2 instance.
- You require storage latency in the sub-millisecond range.
- You require a database that's designed for 999% durability, compared with 99.9% durability with Amazon EBS gp3 volumes.
- You need a virtual storage array to deliver 1 million IOPS or more to a single EC2 instance.

- Exadata Smart Flash Cache and Exadata Smart Flash Logging are extremely high in your Exadata on-premises system. The I/O latency for Exadata Smart Flash Cache is typically less than 400 microseconds for read operations. The I/O latency for Amazon EBS io2 Block Express typically ranges between 400 and 600 microseconds.

Oracle ASM considerations

When you use Oracle on Amazon EC2, Oracle and AWS recommend that you implement Oracle Automatic Storage Management (ASM) external redundancy to avoid Amazon EBS failure rates. However, if an EBS volume becomes unavailable in ASM external redundancy mode, the associated ASM disk group goes into a forced dismount. All disks must be located to successfully mount an ASM disk group. Therefore, the database becomes unavailable until all EBS volumes are available. ASM external redundancy effectively provides RAID level 0 reliability, so the chance of impact to the ASM disk group increases with each EBS volume added, and the overall failure rate is the multiple of each individual EBS volume failure rate.

Amazon EBS volumes are replicated within an AWS Availability Zone. However, EBS volumes can still experience a failure. For example, gp3 volumes have a 0.1–0.2 percent annual failure rate, and io2 volumes have an 0.001 percent annual failure rate. You can implement ASM disk groups with normal redundancy or high redundancy to reduce outages that are caused by a single EBS volume failure. However, this is not a best practice, because EBS volumes are replicated within an Availability Zone, and ASM failure group EBS volumes can also be on the same physical hosts as the ASM primary group EBS volumes.

Additional ASM considerations:

- Use Oracle ASM Filter Driver (ASMFD) to implement ASM.
- Make sure that all Oracle ASM disks in a disk group have similar storage performance and availability characteristics. In storage configurations that have mixed speed drives, such as flash memory and hard disk drives (HDD), I/O performance is constrained by the slowest speed drive.
- Make sure that Oracle ASM disks in a disk group have the same capacity to maintain balance.
- Oracle ASM distributes data randomly into selected sets of ASM disks. When you configure the system's storage, consider the initial capacity of the system and plans for future growth. Oracle ASM simplifies the task of accommodating growth. As mentioned earlier, an Amazon EC2 Nitro System instance supports up to 28 volumes. If the DATA ASM disk group requires 96 TiB, four 24 TiB Amazon EBS io2 Block Express volumes would be a better choice than sixteen 6 TiB Amazon EBS io2 Block Express volumes.

- Set up at least two control files across two ASM disk groups.

Oracle on Amazon EC2 best practices

After you migrate data from Exadata on premises to Oracle on Amazon EC2, and before you provide access to end users, consider the following best practices:

- Enable EC2 instance termination protection. This prevents an EC2 instance from being accidentally terminated by requiring the user to disable the protection before terminating the instance.
- Enable the Amazon EC2 automatic recovery feature, which resolves issues if the hardware that hosts an EC2 instance becomes impaired. This feature recovers the instance on different underlying hardware and reduces the need for manual intervention.
- Amazon EC2 offers instances that have up to 24 TiB of memory. These instances support extremely large Oracle SGAs and should be your first choice if you're using multi-TiB Oracle SGAs. However, many EC2 instances and Amazon RDS for Oracle instances also support local instance storage. If you use an Amazon EC2 or Amazon RDS for Oracle instance with NVMe SSD instance storage, you can use ephemeral storage to extend the Oracle SGA database block buffers. This approach enables you to cache objects by using instance storage and provides an average I/O latency of 100 microseconds for read operations. Smart Flash Cache and/ Level 2 Flash Cache work only on instances that use instance storage and require the Oracle Linux operating system. OLTP and data warehouse environments can benefit from this technology. Set the Oracle initialization parameters `DB_FLASH_CACHE_FILE` and `DB_FLASH_CACHE_SIZE` to use Smart Flash Cache.
- Use Oracle Linux as the operating system for your instance. If Oracle Linux isn't an option, consider Red Hat Enterprise Linux (RHEL). EC2 instances that are based on the Graviton processor don't support Oracle databases, because Oracle hasn't released Oracle Database binaries that are compiled for ARM processors. In addition, Amazon Linux isn't supported for Oracle databases.
- Use the latest release of the Oracle software to install Oracle Grid Infrastructure. You can deploy the latest release of the Oracle Grid Infrastructure with an older version of Oracle Database. For example, Oracle Grid Infrastructure 21c supports Oracle Database 19c.
- If you use Oracle RMAN or Oracle Data Guard to migrate from an older release of Oracle Database on Exadata, consider upgrading the database release to the most recent version after migration. If you use Oracle Data Pump, install the latest Oracle Database release on AWS before migration.

- Use an Oracle flash recovery area (FRA) to quickly restore your database without using an RMAN backup. If possible, set the FRA to a minimum of one day. You must set the Oracle initialization parameters `DB_RECOVERY_FILE_DEST_SIZE`, `DB_RECOVERY_FILE_DEST`, and `DB_FLASHBACK_RETENTION_TARGET` (represents the amount of time, in minutes).
- If you migrate multiple database workloads into a single EC2 instance, consider implementing Oracle Database Resource Manager to manage database resource allocation.
- Implement an Oracle `SPFILE` instead of a standalone `PFILE`. An `SPFILE` is a binary file that permits dynamic modifications without requiring an instance restart. Do not specify `PFILE` when using the `STARTUP` command if an `SPFILE` is in use.
- Enable Oracle Automatic Shared Memory Manager (ASMM), which simplifies SGA memory management. Oracle Database automatically distributes memory among SGA components to ensure the most effective memory utilization.
- You might experience an Oracle **db file parallel write** wait event with the database writer process (DBWR). This wait indicates the time that DBWR spends waiting for I/O completion. To resolve this problem, confirm that asynchronous I/O is enabled (Oracle initialization parameter `DISK_ASYNCH_IO`), increase the IOPS for the EBS volumes, and verify that the database buffer cache is large enough to prevent thrashing.
- Run a scan periodically (every two weeks at a minimum) against the EC2 instances and verify compliance. You can use Amazon Inspector for this scan. Amazon Inspector is an automated security assessment service that helps improve the security and compliance of applications that are deployed on AWS. It automatically assesses applications for exposure, vulnerabilities, and deviations from best practices. After performing an assessment, it produces a detailed list of security findings prioritized by level of severity. You can review these findings directly or in the detailed assessment reports that are available through the Amazon Inspector console or API.
- Set up Amazon CloudWatch alarms for AWS CloudTrail. For example, a CloudWatch alarm should be activated when configuration changes occur on security groups. This alerts the operations team when someone tries to gain access to the EC2 instances.
- If your organization requires a zero or near-zero recovery point objective (RPO), use Oracle Data Guard or Oracle Active Data Guard in maximum availability mode. The standby database should reside in a different Availability Zone from the primary database. The maximum protection and maximum availability modes provide an automatic failover environment that is designed for no data loss. Maximum performance mode provides an automatic failover environment that is designed to lose no more than the amount of data (in seconds) specified by the `FastStartFailoverLagLimit` configuration property. We also recommend that you implement Data Guard Broker with Oracle Data Guard or Oracle Active Data Guard. Data Guard

Broker automates configuration and monitoring tasks for Data Guard. Active Data Guard requires an Oracle license.

- Consider using Oracle Active Data Guard automatic block media recovery. If a corrupt data block is encountered when you access a primary database, the block is automatically replaced with an uncorrupt copy of that block from a physical standby database. However, to use this feature, Active Data Guard must run in maximum availability mode and have the Oracle initialization parameter `LOG_ARCHIVE_DEST_n` set to the `SYNC` redo transport mode. Maximum performance mode doesn't support this feature.
- If your organization requires cross-Region disaster recovery, consider implementing Oracle Far Sync. Far Sync requires an Oracle Active Data Guard license.
- Use Oracle Secure Backup (OSB) to back up your database to Amazon S3 by using Oracle RMAN. OSB requires an Oracle license. OSB pricing is based on the number of Oracle RMAN channels in use. You can also use AWS Storage Gateway to back up your database to Amazon S3 directly. You can apply lifecycle policies to the backups in Amazon S3 to move older backups to Amazon S3 Glacier for archiving.

Refactoring recommendations

AWS offers two tools that activate heterogeneous migrations from Oracle to Amazon RDS for PostgreSQL or Amazon Aurora PostgreSQL-Compatible Edition. These tools are the AWS Schema Conversion Tool (AWS SCT) and AWS Database Migration Service (AWS DMS).

AWS SCT automatically converts the source database schema and a majority of the custom code to a format that's compatible with the target database. During a database migration from Oracle to PostgreSQL, AWS SCT automates the conversion of Oracle PL/SQL code to equivalent PL/pgSQL code in PostgreSQL. The custom code that the tool converts includes views, stored procedures, and functions. When a code fragment can't be automatically converted to the target language, AWS SCT documents all locations that require manual input from the application developer. AWS DMS uses CDC to migrate Oracle to PostgreSQL or MySQL.

To migrate an Oracle database to PostgreSQL or MySQL, you usually need to complete both automated and manual tasks. AWS offers migration playbooks that provide step-by-step instructions for basic and complex code conversion strategies. For information about refactoring your Oracle databases, see the following playbooks:

- Oracle to Aurora PostgreSQL Migration Playbook
- Oracle to Aurora MySQL Migration Playbook

Post-migration activities

After you migrate your Exadata workloads to AWS, there are additional tasks and best practices that are essential to maintain a reliable, highly available, performant, and cost-optimized database environment. This section outlines key post-migration best practices and tips.

In this section:

Continuous monitoring

Monitoring is an important part of maintaining the reliability, availability, and performance of databases on AWS. To more easily debug multi-point failures, we recommend collecting monitoring data from all parts of your database environment on AWS.

This section explores the AWS services and tools that provide advanced performance diagnostics capabilities. Before you use those tools, we recommend that you define a clear monitoring plan.

Monitoring plan

We recommend that you address the following questions before you create your monitoring plan:

- What are your monitoring goals?
- Which resources will you use for monitoring?
- How often will these resources be monitored?
- Which monitoring tools will you use?
- Who will perform the monitoring tasks?
- Who should be notified when something goes wrong?

After you define your monitoring plan, establish a baseline for key metrics, to measure whether your monitoring goals are being met.

Performance baseline

Measure performance under different load conditions at various times. You can monitor metrics such as the following:

- CPU utilization
- Network throughput
- Client connections
- I/O for read or write operations
- Burst credit balances

When performance falls outside your established baseline, you might have to make changes to optimize database availability for the workload. For example, these changes might include changing the instance class of your DB instance or changing the number of DB instances and read replicas that are available for clients.

Key performance guidelines

In general, acceptable values for performance metrics depend on what the application is doing relative to the baseline. Investigate consistent or trending variances from the baseline. The following metrics are often the source of performance issues:

- **High CPU or RAM consumption**. High values for CPU or RAM consumption might be appropriate, if they're consistent with application goals such as throughput or concurrency, and are expected.
- **Disk space consumption**. Investigate disk space consumption if the space used is consistently at or above 85 percent of the total disk space. Evaluate whether it is possible to delete data from the instance or archive data to a different system to free up space.
- **Network traffic**. For network traffic, work with your systems administrator to determine the expected throughput for the domain network and internet connections. We recommend that you investigate network traffic if throughput is consistently lower than expected.
- **Database connections**. If you encounter a high number of user connections along with decreases in instance performance and response time, you might consider limiting database connections. The optimum number of user connections for a DB instance varies based on the instance class and the complexity of the operations that are performed.

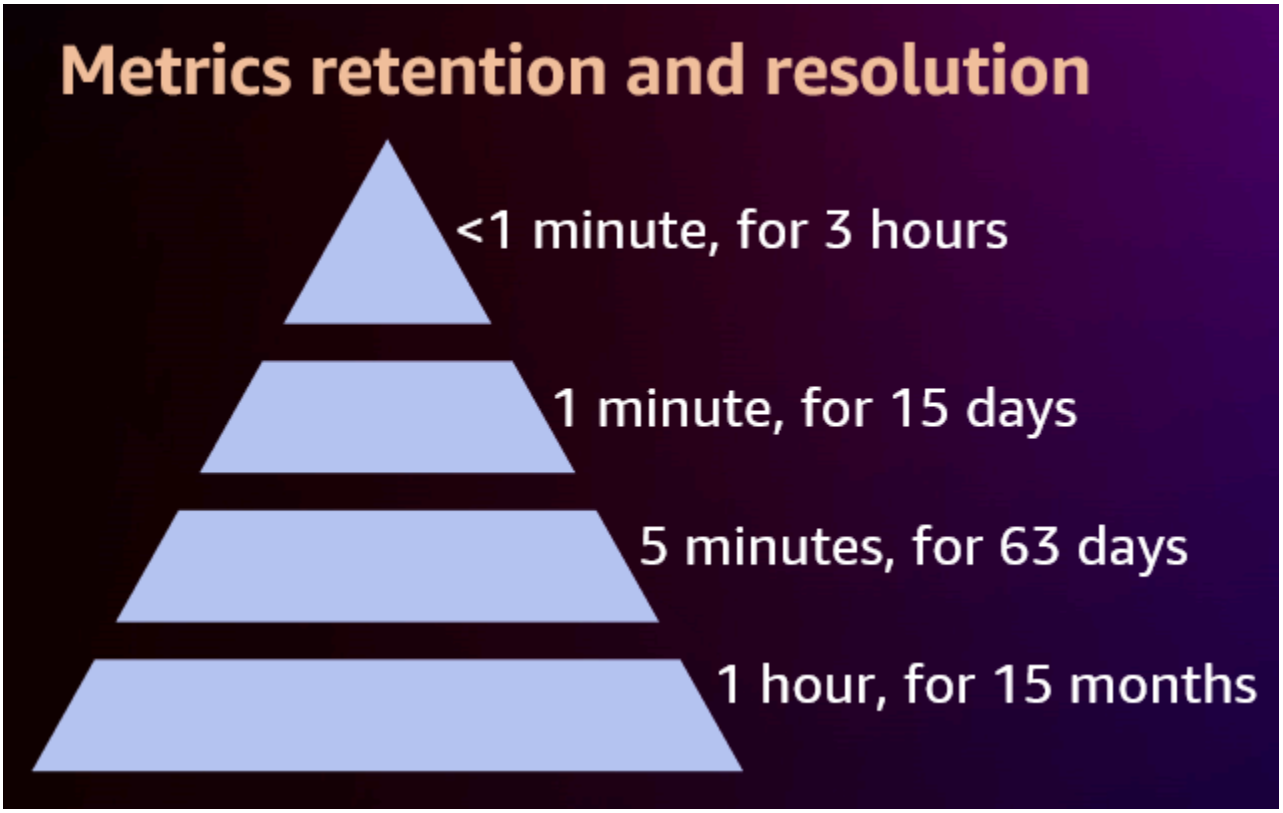
- **IOPS metrics**. When you migrate from Oracle Exadata, IOPS monitoring is essential. Oracle Exadata is known to deliver high storage throughput and IOPS. We recommend that you determine the baseline for typical I/O activity to ensure the best configuration on AWS.

Monitoring tools

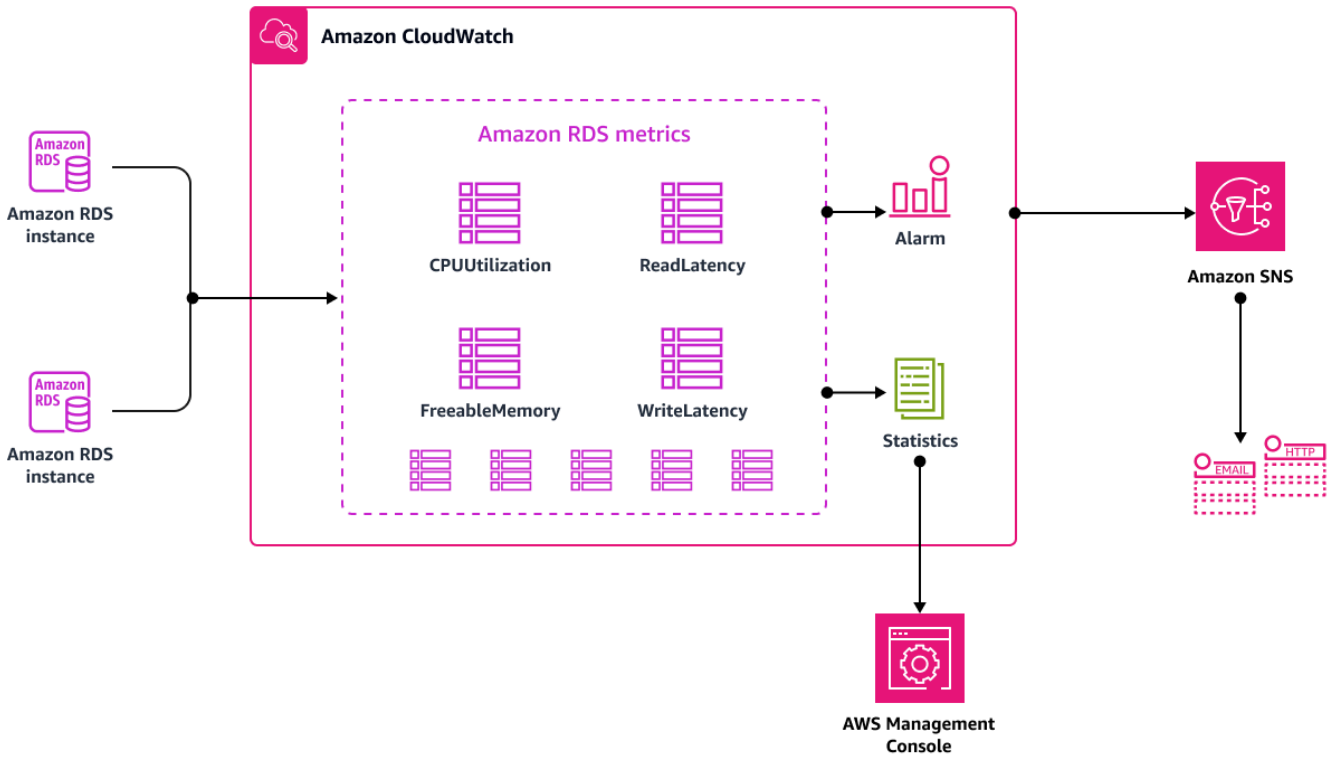
This section discusses monitoring tools from Amazon and Oracle that you can use during the post-migration phase to maintain a reliable, highly available, performant, and cost-optimized database environment.

Amazon CloudWatch

Amazon CloudWatch is a monitoring and observability service that provides a unified view of operational health and gives you complete visibility into the AWS resources, applications, and services running on AWS and on premises. You can use CloudWatch to detect anomalous behavior in your environments, set alarms, visualize logs and metrics side by side, take automated actions, troubleshoot issues, and discover insights to keep your applications running smoothly. The best analogy for CloudWatch metrics resolution and retention is a pyramid structure that's illustrated in the following diagram. The top level represents the most granular frequency (up to 1 second) but also the lowest retention of metrics. As users explore more historical monitoring data, the less granular the data points will be. For example, for maximum retention (between 63 days and 15 months), granularity will be one hour, as illustrated in the bottom level of the pyramid.



As the following diagram shows, you can set up alarms for CloudWatch metrics. For example, you might create an alarm that is activated when the CPU utilization for an instance exceeds 70 percent.



You can configure Amazon Simple Notification Service (Amazon SNS) to send an email or SMS whenever the threshold is passed. You can also use Amazon SNS to notify additional protocols or services such as Amazon Simple Queue Service (Amazon SQS), AWS Lambda, or HTTP/HTTPS. For example, you might create an alarm that is activated if the total IOPS used exceeds 90 percent of the maximum that's configured for the instance. The alarm action might be a Lambda function that increases the amount of provisioned IOPS (PIOPS) if the alarm state is **Alarm**. For additional information, see the presentation Take a load off: Diagnose & resolve performance issues with Amazon RDS (AWS re:Invent 2023).

Enhanced Monitoring

Some users who migrate from Oracle Exadata are used to having OS-level visibility into physical devices that are mapped into their ASM disk groups, and viewing granular OS-level metrics such as huge pages, swap activity, and process/thread list details. Amazon CloudWatch doesn't provide that level of visibility, but Amazon RDS and Amazon Aurora offer Enhanced Monitoring, which provides granular, OS-level monitoring for your databases. Enhanced Monitoring provides a default retention of 30 days and a one-minute sampling frequency, but both settings are configurable.

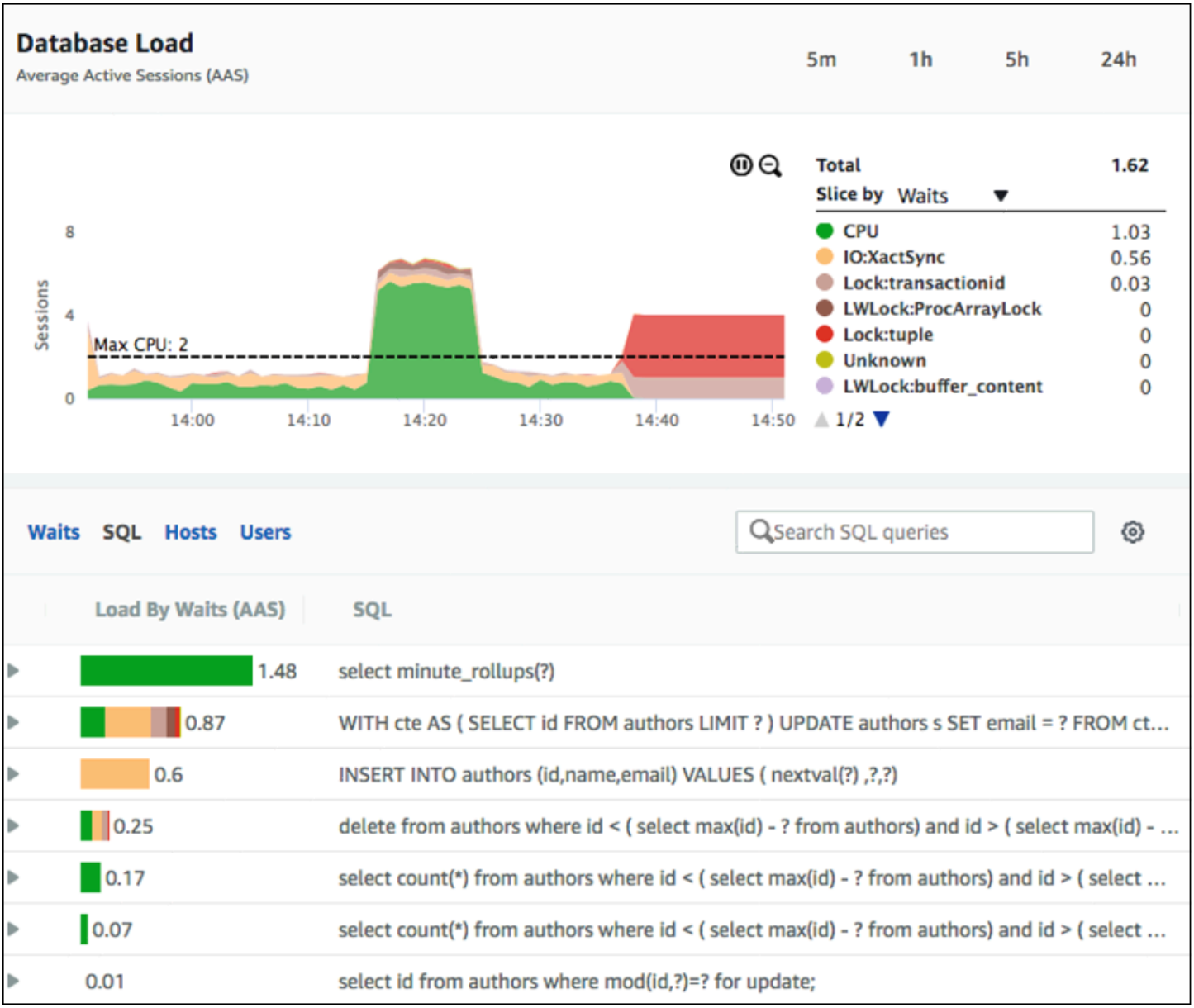
For more information, see the *Monitoring OS metrics with Enhanced Monitoring* sections of the Amazon RDS and Aurora documentation.

> **Note**
>
> Enhanced Monitoring doesn't currently support Oracle databases on Amazon EC2. For these databases, you can use third-party partner solutions or native solutions such as Oracle Enterprise Manager, as discussed in a later section.

Performance Insights

Both Amazon CloudWatch and Amazon RDS Enhanced Monitoring are great tools for instance-level and OS-level monitoring. However, these tools don't provide database engine-level deep-dive performance diagnostics capabilities. Database engine metrics help DBAs identify database bottlenecks such as intensive SQL queries and clearly visualize database load over time. In Amazon RDS and Amazon Aurora, the Performance Insights dashboard displays database load by using a metric named *average active sessions (AAS)*.

The following example shows a maximum of two vCPUs in the monitored Amazon RDS instance. However, two major spikes exceed the number of vCPUs and could indicate a performance bottleneck. One spike represents a major CPU load, shown in green, and the other spike represents a major SQL statements bottleneck, shown in red.



Performance Insights provides that level of visibility by sampling every second of database sessions, looking for active sessions, and ignoring idle sessions. For each active session, Performance Insights collects the following:

- SQL statements
- Wait events such as CPU, I/O, locks, and commit log waits
- Additional dimensions such as hosts and users

Based on this data, you can visualize your database workload and troubleshoot performance issues easily. You can also filter the activity by various dimensions such as hosts and users for additional root-cause analysis. Each database engine has its own set of supported dimensions.

One of the key benefits of Performance Insights is that it doesn't rely on the Oracle Diagnostics Pack, so you can use it to monitor Oracle Database SE2 and other non-Enterprise editions running on Amazon RDS. For more information, see the *Performance Insights* sections of the Amazon RDS and Aurora documentation.

> **Note**
>
> Performance Insights doesn't currently support Oracle databases on Amazon EC2. For these databases, you can use third-party partner solutions or native solutions such as Oracle Enterprise Manager, as discussed in the next section.

Oracle Enterprise Manager

In some cases, Oracle Exadata users might prefer to work with Oracle Enterprise Manager (OEM). Amazon RDS supports OEM through the following options:

Option	Option ID	Supported OEM releases	Supported Oracle Database releases
OEM Database Express	`OEM`	OEM Database Express 12c	Oracle Database 19c (non-CDB only) and Oracle Database 12c
OEM Management Agent	`OEM_AGENT`	• OEM Cloud Control for 13c • OEM Cloud Control for 12c	Oracle Database 19c (non-CDB only) and Oracle Database 12c

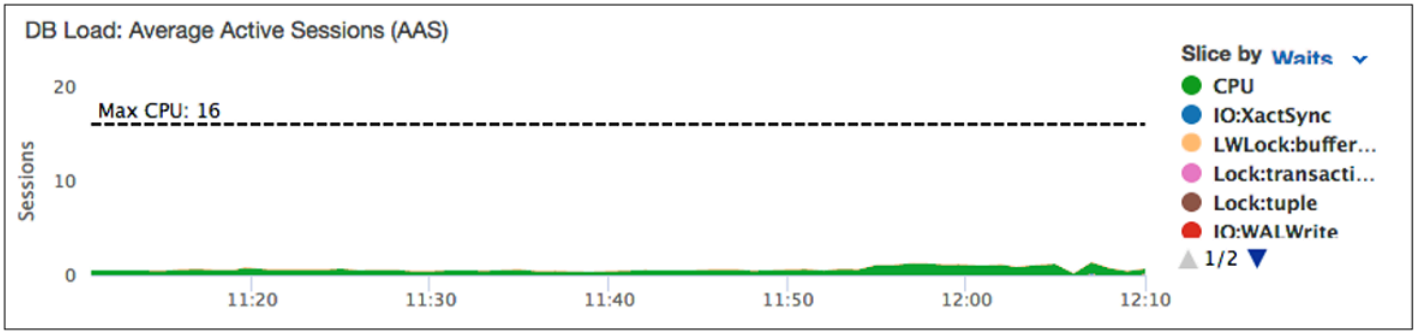
Continuous cost optimization

There are various practices for optimizing database costs on AWS. These include techniques such as instance right-sizing, moving to Oracle Database SE2, using reserved instances, using Amazon with Graviton2 processors, and optimizing SQL statements.

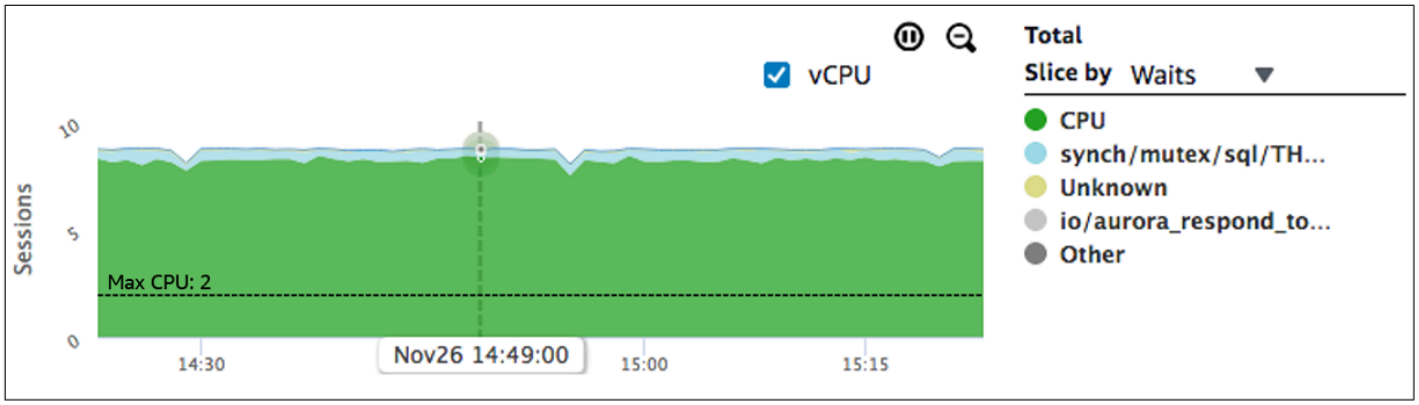
Right-size your instance

Right-sizing is the process of choosing instance and storage types that meet your specific workload performance and capacity requirements at the lowest cost. It is is a key part of optimizing AWS costs.

The previous section covered Performance Insights, which you can use for performance diagnostics as well as right-sizing and cost optimization. For example, if the CPU load is significantly lower than the number of vCPUs, as shown in the following screen illustration, your instance is oversized and you have a significant cost-savings opportunity.



On the other hand, if the CPU load is significantly higher than the number of vCPUs, your instance undersized, as shown in the following screen illustration. In this case, you have a performance optimization opportunity that requires either optimizing your SQL statements to reduce the average active sessions, or moving to a bigger instance that can meet the load requirements.



Consider moving to Oracle Database SE2

Oracle Database Enterprise Edition (EE) has become the standard for many organizations. However, when you perform an in-depth database assessment, you might find that your application might not need all the features of Oracle Database EE.

Oracle Database Standard Edition (SE) is now available as Oracle Database Standard Edition 2 (SE2) for Oracle 12c and 19c. Oracle Database SE2 is a relational database management system (RDBMS) that includes the core features of Oracle Database. These include features that companies can use to support enterprise-class workloads. Given the additional features provided by Amazon RDS and Amazon Aurora, which are available for both EE and SE2 (such as Amazon RDS Multi-AZ and Amazon RDS for cross-Region automated backups, Amazon RDS encryption at rest and in transit, and database activity streams), you might consider using SE2 to save costs.

By switching to SE2, you can optimize Oracle Database license usage. You can provision Oracle Database SE2 for use with Amazon RDS by using both Bring Your Own License (BYOL) and Oracle License Included (LI) options. However, before you decide on such a major change, we recommend that you assess which EE features are being used, which features can be replaced by using Amazon RDS or Aurora capabilities, and which features are mandatory and cannot be replaced or removed, which might prevent you from changing the database edition.

For more information, see Evaluate downgrading Oracle databases to Standard Edition 2 on AWS on the AWS Prescriptive Guidance website.

Use reserved DB instances

You can use Amazon RDS reserved DB instances to reserve a DB instance for a one-year or three-year term, and, in turn, receive a significant discount compared with on-demand DB instances.

You can choose between three payment options when you purchase a reserved instance: All Upfront, Partial Upfront, and No Upfront. With the All Upfront option, you pay for the entire reserved instance before you start using it. This option provides the largest discount compared with on-demand pricing. The Partial Upfront option requires a low upfront payment and a discounted hourly rate for the instance for the duration of the term. The No Upfront option provides a discounted hourly rate for the duration of the term with no upfront payment.

Reserved DB instance types are available in both Amazon RDS and Aurora, for the MySQL, MariaDB, PostgreSQL, Oracle, and SQL Server database engines.

Use AWS Graviton processors

If you migrate from Oracle Exadata to one of the Amazon RDS and Aurora open source databases, you can benefit from the better cost-performance of AWS Graviton2 and Graviton3 processors for Amazon RDS.

Optimize your SQL queries

We recommend that you monitor your database performance and identify top SQL statements that consume significant database resources—for example, by using Amazon RDS Performance Insights—on a regular basis. After you identify resource-intensive SQL statements, apply SQL tuning practices to improve database performance. These tuning practices include, but are not limited to, operations such as index creation or deletion, SQL query rewrites, schema modeling, and features such as materialized views.

SQL optimization improves performance, which results in better application response times and better user experiences, and lowers database costs. For example, a query might consume 60 percent of the database load because of its associated high IOPS and CPU, which might require 200,000 provisioned IOPS (PIOPS) and a large Amazon RDS instance (`r5b.24xl`). By optimizing the query—for example, by creating an index—you can right-size it. As a result, you might be able to pay less for a smaller Amazon RDS DB instance that has less PIOPS.

Automated monitoring

This section discusses key automation capabilities for monitoring your Exadata workloads on AWS.

Amazon CloudWatch alarms and anomaly detection

Creating alarms and invoking alarm actions are best practices for proactive monitoring. When you set up an alarm, a typical question is the threshold for the metrics that you want to monitor. For example, you might create an alarm that is changed to an `ALARM` state when the CPU utilization for an instance exceeds a threshold of 70 percent.

Determining the threshold value isn't always easy, especially because many companies monitor dozens, sometimes hundreds, of metrics across many database instances. This is where Amazon CloudWatch anomaly detection could be useful.

When you use anomaly detection for a metric, CloudWatch applies statistical and machine learning (ML) algorithms. These algorithms continuously analyze system and application metrics, generate a range of expected values that represent typical metric behavior, and surface anomalies with minimal user intervention. These types of alarms don't have a static threshold for determining alarm state. Instead, they compare the metric's value to the expected value based on the anomaly detection model. You can choose whether the alarm responds when the metric value is above the band of expected values, below the band, or both. For more information about using anomaly detection, see the CloudWatch documentation.

For example, you can specify an alarm based on the ReadIOPS metric for an Amazon RDS for Oracle instance by using the wizard in CloudWatch and choosing the anomaly detection option instead of the static option. For instructions, see the Amazon CloudWatch documentation.

Amazon DevOps Guru for Amazon RDS

Amazon DevOps Guru for Amazon RDS is an ML-powered capability that helps you quickly detect, diagnose, and remediate a wide variety of database-related issues. When DevOps Guru for Amazon RDS automatically detects a database-related issue such as resource over-utilization or misbehavior of SQL queries, the service immediately notifies you and provides diagnostic information, details on the extent of the problem, and intelligent recommendations to help you quickly resolve the issue.

> **Note**
>
> DevOps Guru for Amazon RDS currently supports heterogeneous migrations from Oracle Exadata to Amazon Aurora MySQL-Compatible Edition, Aurora PostgreSQL-Compatible

Edition, and Amazon RDS for PostgreSQL. It doesn't support Oracle databases on Amazon EC2, Amazon RDS, or Aurora.

For example, consider an online bookstore. Let's assume that the bookstore website has a high concurrency spike because a large number of users wanted to purchase a book after it was promoted on TV. Each customer purchase reduces the availability of that book. Here is an example of a SQL statement that runs behind the scenes after each purchase:

```
update book_inventory
set available = available -1
where book_series =: series and book_title =: title;
```

The high concurrency from many DML statements accessing the same rows at the same time could result in table locks. However, Amazon CloudWatch won't display any major spikes in CPU load, because locks usually do not consume significant CPU resources. In this scenario, DevOps Guru can automatically identify an unusual spike in database activity by looking at the average active sessions metric and detecting values that deviate from the typical baseline.

For more information, see Analyzing performance anomalies with Amazon DevOps Guru for Amazon RDS in the Amazon RDSdocumentation.

Automated auditing

Implementing security auditing has become increasingly important because of compliance requirements and security threats. Many users prefer to continue the auditing activities they perform with Oracle on Exadata. AWS provides two auditing options for your databases: basic Amazon RDS auditing and database activity streams.

Basic Amazon RDS auditing

Amazon RDS for Oracle provides the following auditing features:

- `log` **and** `listener.log` **files**. You can push these critical log files automatically to Amazon CloudWatch for longer retention and analysis.
- **Standard auditing**. You can use this native Oracle feature to audit SQL statements, privileges, schemas, objects, network, and multi-tier activity. Oracle recommends using standard auditing on versions before Oracle Database 12c release 1 (12.1). Standard auditing can be difficult to

manage because of multiple audit trails that have different parameters to control auditing behavior and the lack of granular auditing options.

- **Unified auditing.** Oracle Database 12.1 and later versions offer unified auditing. This feature provides audit data in a single location and in a single format. Amazon RDS for Oracle supports mixed-mode auditing, which is enabled by default to support both standard auditing and unified auditing.

Database activity streams

Database activity streams provide a real-time data stream of all database activity. This feature helps companies monitor, audit, and protect databases from unauthorized access and meet compliance and regulatory requirements. It reduces the work required to satisfy compliance goals and facilitates migration to managed database services on AWS. Database activity streams provide real-time data that's integrated into the existing monitoring and alert infrastructure, so you can use your existing processes, tools, and reports. Here is a typical use case:

1. Grant access to Partner applications for Amazon Kinesis Data Streams and AWS Key Management Service (AWS KMS) to monitor database activity.
2. Connect Amazon Kinesis Data Streams to Amazon Data Firehose to save activities to Amazon S3 for long-term retention.
3. Connect to AWS Lambda to analyze or monitor database activities.

> **Note**
>
> The database activity streams feature is available in Amazon RDS and Amazon Aurora. It supports both heterogeneous and homogeneous database migration scenarios.

Summary

To build modern applications and to maximize business agility and cost savings, you need a data infrastructure that can meet the unique needs of your application and its microservices. When you modernize your application, we recommend that you consider factors such as your resource requirements, your feature usage, and your monitoring and auditing needs before you determine your target migration path.

This guide covered key aspects of an Exadata to AWS migration project, including pre-migration discovery, performing the migration, and maintaining a reliable, highly available, performance-efficient and cost-optimized database environment after migration. To start the modernization journey, contact the AWS account team to set up a complimentary discovery session.

Resources

This section summarizes AWS tools, programs, and other resources that can assist in your migration from Oracle Exadata to AWS.

Tools and services

- AWS Database Migration Service (AWS DMS) helps you migrate your databases quickly and securely to AWS. Your source database remains fully operational during the migration, which minimizes downtime for applications that rely on the database. AWS DMS supports widely used commercial and open-source databases, including homogeneous migrations, such as on-premises Oracle Database to Oracle Database in the cloud, and heterogeneous migrations between different database platforms, such as Oracle Database or Microsoft SQL Server to Amazon Aurora. You can also use AWS DMS to continuously replicate data with low latency from any supported source to any supported target. For example, you can replicate data from multiple sources to Amazon S3 to build a highly available and scalable data lake solution. You can also consolidate databases into a petabyte-scale data warehouse by streaming data to Amazon Redshift. AWS DMS can be particularly useful if you require minimum downtime during migration, which typically involves a change data capture (CDC) solution. AWS DMS has advantages over other CDC solutions such as Oracle GoldenGate, because it is a native AWS service. It's also cost-efficient: Your costs are limited to the underlying EC2 instance that runs the AWS DMS replication instance, and possibly additional storage and data transfer costs. Furthermore, because AWS DMS is a fully managed service, resource requirements and operational costs associated with it are minimal compared with most other data migration and replication solutions.
- AWS Schema Conversion Tool (AWS SCT) offers predictable heterogeneous database migrations. It automatically converts the source database schema and a majority of the database code objects, including views, stored procedures, and functions, to a format that's compatible with the target database. Any objects that can't be automatically converted are marked for manual conversion. AWS SCT can also scan application source code for embedded SQL statements and convert them as part of a database schema conversion project. During this process, AWS SCT performs cloud-native code optimization by converting legacy Oracle and SQL Server functions to their AWS service equivalents, which helps you modernize your applications.

Programs

- The AWS Migration Acceleration Program (MAP) is a comprehensive cloud migration program that's based on our experiences at AWSmigrating thousands of enterprise customers to the AWS Cloud. Enterprise migrations can be complex and time-consuming, but MAP can help accelerate your cloud migration and modernization tasks by using an outcome-driven methodology. MAP provides tools that reduce costs and automate and accelerate implementation, tailored training approaches and content, expertise from AWS Professional Services, a global AWS Partner community, and AWS investment. MAP also uses a proven three-phased framework (assess, mobilize, and migrate and modernize) to help companies achieve migration goals.
- AWS Optimization and Licensing Assessment (AWS OLA) helps you save on third-party licensing costs and run resources more efficiently. AWS OLA is a free program for new and existing customers to assess and optimize current on-premises and cloud environments, based on actual resource utilization, third-party licensing, and application dependencies. Use AWS OLA to build your migration and licensing strategy on AWS. This program provides a report that models deployment options by using existing licensing entitlements. These results can help you explore available cost savings across flexible licensing options.
- Amazon Database Migration Accelerator (DMA) brings together AWS DMS, AWS SCT, and AWS database experts to help customers migrate away from traditional commercial databases and analytics services. This program offers migration advisory services, such as creating migration strategies, solutions, and implementation plans, or unblocking ongoing stalled or delayed migrations. Amazon DMA has supported thousands of customers, including BMC Software and Thomson Reuters, by modernizing their databases to Amazon Aurora, Amazon RDS for PostgreSQL or MySQL, Amazon Redshift, Amazon DynamoDB, and others.

Case studies

- The AWS blog post EDF Completes Groundbreaking Migration to Run Oracle Utilities Solution on Amazon RDS describes how the electricity provider EDF migrated from Oracle Exadata to AWS. It provides a real-world example of a successful migration that uses some of the best practices and tools that are covered in this guide.

AWS Prescriptive Guidance content

- Migrating Oracle databases to the AWS Cloud describes the options, tools, and best practices for migrating your Oracle on-premises databases to AWS.
- Migrating bulky Oracle databases to AWS for cross-platform environments explains how you can reduce migration downtime for Oracle databases that are larger than 100 TB by using AWS Snowball, AWS Direct Connect, and Amazon FSx with Oracle XTTS and RMAN incremental backups.
- Transferring Oracle Database dump files from on premises to AWS explains how to migrate Oracle Database dump files to AWS by using Amazon S3, Amazon EFS, and Oracle database links.
- Choosing a DR capability for standard editions of Amazon RDS for Oracle and SQL Server discusses active-active and active-passive disaster recovery (DR) scenarios, and the benefits and limitations of each option for standard editions of Amazon RDS for Oracle and SQL Server.
- Evaluate downgrading Oracle databases to Standard Edition 2 on AWS provides guidance for assessing your Oracle databases and determining whether they can be downgraded to reduce Oracle licensing costs.
- Prioritization guide for refactoring Microsoft SQL Server and Oracle databases on AWS discusses the process for identifying candidate databases to move to open source engines such as PostgreSQL and MySQL on AWS.
- Migration strategy for relational databases focuses on strategies and frameworks for migrating on-premises relational databases, such as Oracle or Microsoft SQL Server, to AWS.
- See also: Migration and modernization patterns for Oracle Database.

Contributors

The following authors and co-authors contributed to this guide:

- Pini Dibask, Senior Database Solutions Architect, AWS
- Tom Harper, NoSQL Solutions Architect Manager, AWS
- Jobin Joseph, Senior Amazon RDS for Oracle Solutions Architect, AWS
- Marvin Vinson, Principal Database Solutions Architect, AWS

Document history

The following table describes significant changes to this guide. If you want to be notified about future updates, you can subscribe to an RSS feed.

Change	Description	Date
Updated information about Amazon EBS volume types	Updated the Replatforming recommendations section with information about io2 Block Express storage options.	July 12, 2024
Updated information about database consolidation	Updated the Consolidation of databases section to clarify that Amazon RDS for Oracle now supports multitenant architectures with multiple pluggable databases.	February 28, 2024
Initial publication	—	January 24, 2024

AWS Prescriptive Guidance glossary

The following are commonly used terms in strategies, guides, and patterns provided by AWS Prescriptive Guidance. To suggest entries, please use the **Provide feedback** link at the end of the glossary.

Numbers

7 Rs

Seven common migration strategies for moving applications to the cloud. These strategies build upon the 5 Rs that Gartner identified in 2011 and consist of the following:

- Refactor/re-architect – Move an application and modify its architecture by taking full advantage of cloud-native features to improve agility, performance, and scalability. This typically involves porting the operating system and database. Example: Migrate your on-premises Oracle database to the Amazon Aurora PostgreSQL-Compatible Edition.
- Replatform (lift and reshape) – Move an application to the cloud, and introduce some level of optimization to take advantage of cloud capabilities. Example: Migrate your on-premises Oracle database to Amazon Relational Database Service (Amazon RDS) for Oracle in the AWS Cloud.
- Repurchase (drop and shop) – Switch to a different product, typically by moving from a traditional license to a SaaS model. Example: Migrate your customer relationship management (CRM) system to Salesforce.com.
- Rehost (lift and shift) – Move an application to the cloud without making any changes to take advantage of cloud capabilities. Example: Migrate your on-premises Oracle database to Oracle on an EC2 instance in the AWS Cloud.
- Relocate (hypervisor-level lift and shift) – Move infrastructure to the cloud without purchasing new hardware, rewriting applications, or modifying your existing operations. You migrate servers from an on-premises platform to a cloud service for the same platform. Example: Migrate a Microsoft Hyper-V application to AWS.
- Retain (revisit) – Keep applications in your source environment. These might include applications that require major refactoring, and you want to postpone that work until a later time, and legacy applications that you want to retain, because there's no business justification for migrating them.

- Retire – Decommission or remove applications that are no longer needed in your source environment.

A

ABAC

See attribute-based access control.

abstracted services

See managed services.

ACID

See atomicity, consistency, isolation, durability.

active-active migration

A database migration method in which the source and target databases are kept in sync (by using a bidirectional replication tool or dual write operations), and both databases handle transactions from connecting applications during migration. This method supports migration in small, controlled batches instead of requiring a one-time cutover. It's more flexible but requires more work than active-passive migration.

active-passive migration

A database migration method in which in which the source and target databases are kept in sync, but only the source database handles transactions from connecting applications while data is replicated to the target database. The target database doesn't accept any transactions during migration.

aggregate function

A SQL function that operates on a group of rows and calculates a single return value for the group. Examples of aggregate functions include `SUM` and `MAX`.

AI

See artificial intelligence.

AIOps

See artificial intelligence operations.

anonymization

The process of permanently deleting personal information in a dataset. Anonymization can help protect personal privacy. Anonymized data is no longer considered to be personal data.

anti-pattern

A frequently used solution for a recurring issue where the solution is counter-productive, ineffective, or less effective than an alternative.

application control

A security approach that allows the use of only approved applications in order to help protect a system from malware.

application portfolio

A collection of detailed information about each application used by an organization, including the cost to build and maintain the application, and its business value. This information is key to the portfolio discovery and analysis process and helps identify and prioritize the applications to be migrated, modernized, and optimized.

artificial intelligence (AI)

The field of computer science that is dedicated to using computing technologies to perform cognitive functions that are typically associated with humans, such as learning, solving problems, and recognizing patterns. For more information, see What is Artificial Intelligence?

artificial intelligence operations (AIOps)

The process of using machine learning techniques to solve operational problems, reduce operational incidents and human intervention, and increase service quality. For more information about how AIOps is used in the AWS migration strategy, see the operations integration guide.

asymmetric encryption

An encryption algorithm that uses a pair of keys, a public key for encryption and a private key for decryption. You can share the public key because it isn't used for decryption, but access to the private key should be highly restricted.

atomicity, consistency, isolation, durability (ACID)

A set of software properties that guarantee the data validity and operational reliability of a database, even in the case of errors, power failures, or other problems.

attribute-based access control (ABAC)

The practice of creating fine-grained permissions based on user attributes, such as department, job role, and team name. For more information, see ABAC for AWS in the AWS Identity and Access Management (IAM) documentation.

authoritative data source

A location where you store the primary version of data, which is considered to be the most reliable source of information. You can copy data from the authoritative data source to other locations for the purposes of processing or modifying the data, such as anonymizing, redacting, or pseudonymizing it.

Availability Zone

A distinct location within an AWS Region that is insulated from failures in other Availability Zones and provides inexpensive, low-latency network connectivity to other Availability Zones in the same Region.

AWS Cloud Adoption Framework (AWS CAF)

A framework of guidelines and best practices from AWS to help organizations develop an efficient and effective plan to move successfully to the cloud. AWS CAF organizes guidance into six focus areas called perspectives: business, people, governance, platform, security, and operations. The business, people, and governance perspectives focus on business skills and processes; the platform, security, and operations perspectives focus on technical skills and processes. For example, the people perspective targets stakeholders who handle human resources (HR), staffing functions, and people management. For this perspective, AWS CAF provides guidance for people development, training, and communications to help ready the organization for successful cloud adoption. For more information, see the AWS CAF website and the AWS CAF whitepaper.

AWS Workload Qualification Framework (AWS WQF)

A tool that evaluates database migration workloads, recommends migration strategies, and provides work estimates. AWS WQF is included with AWS Schema Conversion Tool (AWS SCT). It analyzes database schemas and code objects, application code, dependencies, and performance characteristics, and provides assessment reports.

B

bad bot

A bot that is intended to disrupt or cause harm to individuals or organizations.

BCP

See business continuity planning.

behavior graph

A unified, interactive view of resource behavior and interactions over time. You can use a behavior graph with Amazon Detective to examine failed logon attempts, suspicious API calls, and similar actions. For more information, see Data in a behavior graph in the Detective documentation.

big-endian system

A system that stores the most significant byte first. See also endianness.

binary classification

A process that predicts a binary outcome (one of two possible classes). For example, your ML model might need to predict problems such as “Is this email spam or not spam?" or "Is this product a book or a car?"

bloom filter

A probabilistic, memory-efficient data structure that is used to test whether an element is a member of a set.

blue/green deployment

A deployment strategy where you create two separate but identical environments. You run the current application version in one environment (blue) and the new application version in the other environment (green). This strategy helps you quickly roll back with minimal impact.

bot

A software application that runs automated tasks over the internet and simulates human activity or interaction. Some bots are useful or beneficial, such as web crawlers that index information on the internet. Some other bots, known as *bad bots*, are intended to disrupt or cause harm to individuals or organizations.

botnet

Networks of bots that are infected by malware and are under the control of a single party, known as a *bot herder* or *bot operator*. Botnets are the best-known mechanism to scale bots and their impact.

branch

A contained area of a code repository. The first branch created in a repository is the *main branch*. You can create a new branch from an existing branch, and you can then develop features or fix bugs in the new branch. A branch you create to build a feature is commonly referred to as a *feature branch*. When the feature is ready for release, you merge the feature branch back into the main branch. For more information, see About branches (GitHub documentation).

break-glass access

In exceptional circumstances and through an approved process, a quick means for a user to gain access to an AWS account that they don't typically have permissions to access. For more information, see the Implement break-glass procedures indicator in the AWS Well-Architected guidance.

brownfield strategy

The existing infrastructure in your environment. When adopting a brownfield strategy for a system architecture, you design the architecture around the constraints of the current systems and infrastructure. If you are expanding the existing infrastructure, you might blend brownfield and greenfield strategies.

buffer cache

The memory area where the most frequently accessed data is stored.

business capability

What a business does to generate value (for example, sales, customer service, or marketing). Microservices architectures and development decisions can be driven by business capabilities. For more information, see the Organized around business capabilities section of the Running containerized microservices on AWS whitepaper.

business continuity planning (BCP)

A plan that addresses the potential impact of a disruptive event, such as a large-scale migration, on operations and enables a business to resume operations quickly.

C

CAF

See AWS Cloud Adoption Framework.

canary deployment

The slow and incremental release of a version to end users. When you are confident, you deploy the new version and replace the current version in its entirety.

CCoE

See Cloud Center of Excellence.

CDC

See change data capture.

change data capture (CDC)

The process of tracking changes to a data source, such as a database table, and recording metadata about the change. You can use CDC for various purposes, such as auditing or replicating changes in a target system to maintain synchronization.

chaos engineering

Intentionally introducing failures or disruptive events to test a system's resilience. You can use AWS Fault Injection Service (AWS FIS) to perform experiments that stress your AWS workloads and evaluate their response.

CI/CD

See continuous integration and continuous delivery.

classification

A categorization process that helps generate predictions. ML models for classification problems predict a discrete value. Discrete values are always distinct from one another. For example, a model might need to evaluate whether or not there is a car in an image.

client-side encryption

Encryption of data locally, before the target AWS service receives it.

Cloud Center of Excellence (CCoE)

A multi-disciplinary team that drives cloud adoption efforts across an organization, including developing cloud best practices, mobilizing resources, establishing migration timelines, and leading the organization through large-scale transformations. For more information, see the CCoE posts on the AWS Cloud Enterprise Strategy Blog.

cloud computing

The cloud technology that is typically used for remote data storage and IoT device management. Cloud computing is commonly connected to edge computing technology.

cloud operating model

In an IT organization, the operating model that is used to build, mature, and optimize one or more cloud environments. For more information, see Building your Cloud Operating Model.

cloud stages of adoption

The four phases that organizations typically go through when they migrate to the AWS Cloud:

- Project – Running a few cloud-related projects for proof of concept and learning purposes
- Foundation – Making foundational investments to scale your cloud adoption (e.g., creating a landing zone, defining a CCoE, establishing an operations model)
- Migration – Migrating individual applications
- Re-invention – Optimizing products and services, and innovating in the cloud

These stages were defined by Stephen Orban in the blog post The Journey Toward Cloud-First & the Stages of Adoption on the AWS Cloud Enterprise Strategy blog. For information about how they relate to the AWS migration strategy, see the migration readiness guide.

CMDB

See configuration management database.

code repository

A location where source code and other assets, such as documentation, samples, and scripts, are stored and updated through version control processes. Common cloud repositories include GitHub or Bitbucket Cloud. Each version of the code is called a *branch*. In a microservice structure, each repository is devoted to a single piece of functionality. A single CI/CD pipeline can use multiple repositories.

cold cache

A buffer cache that is empty, not well populated, or contains stale or irrelevant data. This affects performance because the database instance must read from the main memory or disk, which is slower than reading from the buffer cache.

cold data

Data that is rarely accessed and is typically historical. When querying this kind of data, slow queries are typically acceptable. Moving this data to lower-performing and less expensive storage tiers or classes can reduce costs.

computer vision (CV)

A field of AI that uses machine learning to analyze and extract information from visual formats such as digital images and videos. For example, Amazon SageMaker AI provides image processing algorithms for CV.

configuration drift

For a workload, a configuration change from the expected state. It might cause the workload to become noncompliant, and it's typically gradual and unintentional.

configuration management database (CMDB)

A repository that stores and manages information about a database and its IT environment, including both hardware and software components and their configurations. You typically use data from a CMDB in the portfolio discovery and analysis stage of migration.

conformance pack

A collection of AWS Config rules and remediation actions that you can assemble to customize your compliance and security checks. You can deploy a conformance pack as a single entity in an AWS account and Region, or across an organization, by using a YAML template. For more information, see Conformance packs in the AWS Config documentation.

continuous integration and continuous delivery (CI/CD)

The process of automating the source, build, test, staging, and production stages of the software release process. CI/CD is commonly described as a pipeline. CI/CD can help you automate processes, improve productivity, improve code quality, and deliver faster. For more information, see Benefits of continuous delivery. CD can also stand for *continuous deployment*. For more information, see Continuous Delivery vs. Continuous Deployment.

CV

See computer vision.

D

data at rest

Data that is stationary in your network, such as data that is in storage.

data classification

A process for identifying and categorizing the data in your network based on its criticality and sensitivity. It is a critical component of any cybersecurity risk management strategy because it helps you determine the appropriate protection and retention controls for the data. Data classification is a component of the security pillar in the AWS Well-Architected Framework. For more information, see Data classification.

data drift

A meaningful variation between the production data and the data that was used to train an ML model, or a meaningful change in the input data over time. Data drift can reduce the overall quality, accuracy, and fairness in ML model predictions.

data in transit

Data that is actively moving through your network, such as between network resources.

data mesh

An architectural framework that provides distributed, decentralized data ownership with centralized management and governance.

data minimization

The principle of collecting and processing only the data that is strictly necessary. Practicing data minimization in the AWS Cloud can reduce privacy risks, costs, and your analytics carbon footprint.

data perimeter

A set of preventive guardrails in your AWS environment that help make sure that only trusted identities are accessing trusted resources from expected networks. For more information, see Building a data perimeter on AWS.

data preprocessing

To transform raw data into a format that is easily parsed by your ML model. Preprocessing data can mean removing certain columns or rows and addressing missing, inconsistent, or duplicate values.

data provenance

The process of tracking the origin and history of data throughout its lifecycle, such as how the data was generated, transmitted, and stored.

data subject

An individual whose data is being collected and processed.

data warehouse

A data management system that supports business intelligence, such as analytics. Data warehouses commonly contain large amounts of historical data, and they are typically used for queries and analysis.

database definition language (DDL)

Statements or commands for creating or modifying the structure of tables and objects in a database.

database manipulation language (DML)

Statements or commands for modifying (inserting, updating, and deleting) information in a database.

DDL

See database definition language.

deep ensemble

To combine multiple deep learning models for prediction. You can use deep ensembles to obtain a more accurate prediction or for estimating uncertainty in predictions.

deep learning

An ML subfield that uses multiple layers of artificial neural networks to identify mapping between input data and target variables of interest.

defense-in-depth

An information security approach in which a series of security mechanisms and controls are thoughtfully layered throughout a computer network to protect the confidentiality, integrity, and availability of the network and the data within. When you adopt this strategy on AWS, you add multiple controls at different layers of the AWS Organizations structure to help secure resources. For example, a defense-in-depth approach might combine multi-factor authentication, network segmentation, and encryption.

delegated administrator

In AWS Organizations, a compatible service can register an AWS member account to administer the organization's accounts and manage permissions for that service. This account is called the *delegated administrator* for that service. For more information and a list of compatible services, see Services that work with AWS Organizations in the AWS Organizations documentation.

deployment

The process of making an application, new features, or code fixes available in the target environment. Deployment involves implementing changes in a code base and then building and running that code base in the application's environments.

development environment

See environment.

detective control

A security control that is designed to detect, log, and alert after an event has occurred. These controls are a second line of defense, alerting you to security events that bypassed the preventative controls in place. For more information, see Detective controls in *Implementing security controls on AWS*.

development value stream mapping (DVSM)

A process used to identify and prioritize constraints that adversely affect speed and quality in a software development lifecycle. DVSM extends the value stream mapping process originally designed for lean manufacturing practices. It focuses on the steps and teams required to create and move value through the software development process.

digital twin

A virtual representation of a real-world system, such as a building, factory, industrial equipment, or production line. Digital twins support predictive maintenance, remote monitoring, and production optimization.

dimension table

In a star schema, a smaller table that contains data attributes about quantitative data in a fact table. Dimension table attributes are typically text fields or discrete numbers that behave like text. These attributes are commonly used for query constraining, filtering, and result set labeling.

disaster

An event that prevents a workload or system from fulfilling its business objectives in its primary deployed location. These events can be natural disasters, technical failures, or the result of human actions, such as unintentional misconfiguration or a malware attack.

disaster recovery (DR)

The strategy and process you use to minimize downtime and data loss caused by a disaster. For more information, see Disaster Recovery of Workloads on AWS: Recovery in the Cloud in the AWS Well-Architected Framework.

DML

See database manipulation language.

domain-driven design

An approach to developing a complex software system by connecting its components to evolving domains, or core business goals, that each component serves. This concept was introduced by Eric Evans in his book, *Domain-Driven Design: Tackling Complexity in the Heart of Software* (Boston: Addison-Wesley Professional, 2003). For information about how you can use domain-driven design with the strangler fig pattern, see Modernizing legacy Microsoft ASP.NET (ASMX) web services incrementally by using containers and Amazon API Gateway.

DR

See disaster recovery.

drift detection

Tracking deviations from a baselined configuration. For example, you can use AWS CloudFormation to detect drift in system resources, or you can use AWS Control Tower to detect changes in your landing zone that might affect compliance with governance requirements.

DVSM

See development value stream mapping.

E

EDA

See exploratory data analysis.

EDI

See electronic data interchange.

edge computing

The technology that increases the computing power for smart devices at the edges of an IoT network. When compared with cloud computing, edge computing can reduce communication latency and improve response time.

electronic data interchange (EDI)

The automated exchange of business documents between organizations. For more information, see What is Electronic Data Interchange.

encryption

A computing process that transforms plaintext data, which is human-readable, into ciphertext.

encryption key

A cryptographic string of randomized bits that is generated by an encryption algorithm. Keys can vary in length, and each key is designed to be unpredictable and unique.

endianness

The order in which bytes are stored in computer memory. Big-endian systems store the most significant byte first. Little-endian systems store the least significant byte first.

endpoint

See service endpoint.

endpoint service

A service that you can host in a virtual private cloud (VPC) to share with other users. You can create an endpoint service with AWS PrivateLink and grant permissions to other AWS accounts or to AWS Identity and Access Management (IAM) principals. These accounts or principals can connect to your endpoint service privately by creating interface VPC endpoints. For more

information, see Create an endpoint service in the Amazon Virtual Private Cloud (Amazon VPC) documentation.

enterprise resource planning (ERP)

A system that automates and manages key business processes (such as accounting, MES, and project management) for an enterprise.

envelope encryption

The process of encrypting an encryption key with another encryption key. For more information, see Envelope encryption in the AWS Key Management Service (AWS KMS) documentation.

environment

An instance of a running application. The following are common types of environments in cloud computing:

- development environment – An instance of a running application that is available only to the core team responsible for maintaining the application. Development environments are used to test changes before promoting them to upper environments. This type of environment is sometimes referred to as a *test environment*.
- lower environments – All development environments for an application, such as those used for initial builds and tests.
- production environment – An instance of a running application that end users can access. In a CI/CD pipeline, the production environment is the last deployment environment.
- upper environments – All environments that can be accessed by users other than the core development team. This can include a production environment, preproduction environments, and environments for user acceptance testing.

epic

In agile methodologies, functional categories that help organize and prioritize your work. Epics provide a high-level description of requirements and implementation tasks. For example, AWS CAF security epics include identity and access management, detective controls, infrastructure security, data protection, and incident response. For more information about epics in the AWS migration strategy, see the program implementation guide.

ERP

See enterprise resource planning.

exploratory data analysis (EDA)

The process of analyzing a dataset to understand its main characteristics. You collect or aggregate data and then perform initial investigations to find patterns, detect anomalies, and check assumptions. EDA is performed by calculating summary statistics and creating data visualizations.

F

fact table

The central table in a star schema. It stores quantitative data about business operations. Typically, a fact table contains two types of columns: those that contain measures and those that contain a foreign key to a dimension table.

fail fast

A philosophy that uses frequent and incremental testing to reduce the development lifecycle. It is a critical part of an agile approach.

fault isolation boundary

In the AWS Cloud, a boundary such as an Availability Zone, AWS Region, control plane, or data plane that limits the effect of a failure and helps improve the resilience of workloads. For more information, see AWS Fault Isolation Boundaries.

feature branch

See branch.

features

The input data that you use to make a prediction. For example, in a manufacturing context, features could be images that are periodically captured from the manufacturing line.

feature importance

How significant a feature is for a model's predictions. This is usually expressed as a numerical score that can be calculated through various techniques, such as Shapley Additive Explanations (SHAP) and integrated gradients. For more information, see Machine learning model interpretability with AWS.

feature transformation

To optimize data for the ML process, including enriching data with additional sources, scaling values, or extracting multiple sets of information from a single data field. This enables the ML model to benefit from the data. For example, if you break down the “2021-05-27 00:15:37” date into “2021”, “May”, “Thu”, and “15”, you can help the learning algorithm learn nuanced patterns associated with different data components.

few-shot prompting

Providing an LLM with a small number of examples that demonstrate the task and desired output before asking it to perform a similar task. This technique is an application of in-context learning, where models learn from examples (*shots*) that are embedded in prompts. Few-shot prompting can be effective for tasks that require specific formatting, reasoning, or domain knowledge. See also zero-shot prompting.

FGAC

See fine-grained access control.

fine-grained access control (FGAC)

The use of multiple conditions to allow or deny an access request.

flash-cut migration

A database migration method that uses continuous data replication through change data capture to migrate data in the shortest time possible, instead of using a phased approach. The objective is to keep downtime to a minimum.

FM

See foundation model.

foundation model (FM)

A large deep-learning neural network that has been training on massive datasets of generalized and unlabeled data. FMs are capable of performing a wide variety of general tasks, such as understanding language, generating text and images, and conversing in natural language. For more information, see What are Foundation Models.

G

generative AI

A subset of AI models that have been trained on large amounts of data and that can use a simple text prompt to create new content and artifacts, such as images, videos, text, and audio. For more information, see What is Generative AI.

geo blocking

See geographic restrictions.

geographic restrictions (geo blocking)

In Amazon CloudFront, an option to prevent users in specific countries from accessing content distributions. You can use an allow list or block list to specify approved and banned countries. For more information, see Restricting the geographic distribution of your content in the CloudFront documentation.

Gitflow workflow

An approach in which lower and upper environments use different branches in a source code repository. The Gitflow workflow is considered legacy, and the trunk-based workflow is the modern, preferred approach.

golden image

A snapshot of a system or software that is used as a template to deploy new instances of that system or software. For example, in manufacturing, a golden image can be used to provision software on multiple devices and helps improve speed, scalability, and productivity in device manufacturing operations.

greenfield strategy

The absence of existing infrastructure in a new environment. When adopting a greenfield strategy for a system architecture, you can select all new technologies without the restriction of compatibility with existing infrastructure, also known as brownfield. If you are expanding the existing infrastructure, you might blend brownfield and greenfield strategies.

guardrail

A high-level rule that helps govern resources, policies, and compliance across organizational units (OUs). *Preventive guardrails* enforce policies to ensure alignment to compliance standards. They are implemented by using service control policies and IAM permissions boundaries.

Detective guardrails detect policy violations and compliance issues, and generate alerts for remediation. They are implemented by using AWS Config, AWS Security Hub, Amazon GuardDuty, AWS Trusted Advisor, Amazon Inspector, and custom AWS Lambda checks.

H

HA

See high availability.

heterogeneous database migration

Migrating your source database to a target database that uses a different database engine (for example, Oracle to Amazon Aurora). Heterogeneous migration is typically part of a re-architecting effort, and converting the schema can be a complex task. AWS provides AWS SCT that helps with schema conversions.

high availability (HA)

The ability of a workload to operate continuously, without intervention, in the event of challenges or disasters. HA systems are designed to automatically fail over, consistently deliver high-quality performance, and handle different loads and failures with minimal performance impact.

historian modernization

An approach used to modernize and upgrade operational technology (OT) systems to better serve the needs of the manufacturing industry. A *historian* is a type of database that is used to collect and store data from various sources in a factory.

holdout data

A portion of historical, labeled data that is withheld from a dataset that is used to train a machine learning model. You can use holdout data to evaluate the model performance by comparing the model predictions against the holdout data.

homogeneous database migration

Migrating your source database to a target database that shares the same database engine (for example, Microsoft SQL Server to Amazon RDS for SQL Server). Homogeneous migration is typically part of a rehosting or replatforming effort. You can use native database utilities to migrate the schema.

hot data

Data that is frequently accessed, such as real-time data or recent translational data. This data typically requires a high-performance storage tier or class to provide fast query responses.

hotfix

An urgent fix for a critical issue in a production environment. Due to its urgency, a hotfix is usually made outside of the typical DevOps release workflow.

hypercare period

Immediately following cutover, the period of time when a migration team manages and monitors the migrated applications in the cloud in order to address any issues. Typically, this period is 1–4 days in length. At the end of the hypercare period, the migration team typically transfers responsibility for the applications to the cloud operations team.

I

IaC

See infrastructure as code.

identity-based policy

A policy attached to one or more IAM principals that defines their permissions within the AWS Cloud environment.

idle application

An application that has an average CPU and memory usage between 5 and 20 percent over a period of 90 days. In a migration project, it is common to retire these applications or retain them on premises.

IIoT

See industrial Internet of Things.

immutable infrastructure

A model that deploys new infrastructure for production workloads instead of updating, patching, or modifying the existing infrastructure. Immutable infrastructures are inherently more consistent, reliable, and predictable than mutable infrastructure. For more information, see the Deploy using immutable infrastructure best practice in the AWS Well-Architected Framework.

inbound (ingress) VPC

In an AWS multi-account architecture, a VPC that accepts, inspects, and routes network connections from outside an application. The AWS Security Reference Architecture recommends setting up your Network account with inbound, outbound, and inspection VPCs to protect the two-way interface between your application and the broader internet.

incremental migration

A cutover strategy in which you migrate your application in small parts instead of performing a single, full cutover. For example, you might move only a few microservices or users to the new system initially. After you verify that everything is working properly, you can incrementally move additional microservices or users until you can decommission your legacy system. This strategy reduces the risks associated with large migrations.

Industry 4.0

A term that was introduced by Klaus Schwab in 2016 to refer to the modernization of manufacturing processes through advances in connectivity, real-time data, automation, analytics, and AI/ML.

infrastructure

All of the resources and assets contained within an application's environment.

infrastructure as code (IaC)

The process of provisioning and managing an application's infrastructure through a set of configuration files. IaC is designed to help you centralize infrastructure management, standardize resources, and scale quickly so that new environments are repeatable, reliable, and consistent.

industrial Internet of Things (IIoT)

The use of internet-connected sensors and devices in the industrial sectors, such as manufacturing, energy, automotive, healthcare, life sciences, and agriculture. For more information, see Building an industrial Internet of Things (IIoT) digital transformation strategy.

inspection VPC

In an AWS multi-account architecture, a centralized VPC that manages inspections of network traffic between VPCs (in the same or different AWS Regions), the internet, and on-premises networks. The AWS Security Reference Architecture recommends setting up your Network account with inbound, outbound, and inspection VPCs to protect the two-way interface between your application and the broader internet.

Internet of Things (IoT)

The network of connected physical objects with embedded sensors or processors that communicate with other devices and systems through the internet or over a local communication network. For more information, see What is IoT?

interpretability

A characteristic of a machine learning model that describes the degree to which a human can understand how the model's predictions depend on its inputs. For more information, see Machine learning model interpretability with AWS.

IoT

See Internet of Things.

IT information library (ITIL)

A set of best practices for delivering IT services and aligning these services with business requirements. ITIL provides the foundation for ITSM.

IT service management (ITSM)

Activities associated with designing, implementing, managing, and supporting IT services for an organization. For information about integrating cloud operations with ITSM tools, see the operations integration guide.

ITIL

See IT information library.

ITSM

See IT service management.

L

label-based access control (LBAC)

An implementation of mandatory access control (MAC) where the users and the data itself are each explicitly assigned a security label value. The intersection between the user security label and data security label determines which rows and columns can be seen by the user.

landing zone

A landing zone is a well-architected, multi-account AWS environment that is scalable and secure. This is a starting point from which your organizations can quickly launch and deploy workloads and applications with confidence in their security and infrastructure environment. For more information about landing zones, see Setting up a secure and scalable multi-account AWS environment.

large language model (LLM)

A deep learning AI model that is pretrained on a vast amount of data. An LLM can perform multiple tasks, such as answering questions, summarizing documents, translating text into other languages, and completing sentences. For more information, see What are LLMs.

large migration

A migration of 300 or more servers.

LBAC

See label-based access control.

least privilege

The security best practice of granting the minimum permissions required to perform a task. For more information, see Apply least-privilege permissions in the IAM documentation.

lift and shift

See 7 Rs.

little-endian system

A system that stores the least significant byte first. See also endianness.

LLM

See large language model.

lower environments

See environment.

M

machine learning (ML)

A type of artificial intelligence that uses algorithms and techniques for pattern recognition and learning. ML analyzes and learns from recorded data, such as Internet of Things (IoT) data, to generate a statistical model based on patterns. For more information, see Machine Learning.

main branch

See branch.

malware

Software that is designed to compromise computer security or privacy. Malware might disrupt computer systems, leak sensitive information, or gain unauthorized access. Examples of malware include viruses, worms, ransomware, Trojan horses, spyware, and keyloggers.

managed services

AWS services for which AWS operates the infrastructure layer, the operating system, and platforms, and you access the endpoints to store and retrieve data. Amazon Simple Storage Service (Amazon S3) and Amazon DynamoDB are examples of managed services. These are also known as *abstracted services*.

manufacturing execution system (MES)

A software system for tracking, monitoring, documenting, and controlling production processes that convert raw materials to finished products on the shop floor.

MAP

See Migration Acceleration Program.

mechanism

A complete process in which you create a tool, drive adoption of the tool, and then inspect the results in order to make adjustments. A mechanism is a cycle that reinforces and improves itself as it operates. For more information, see Building mechanisms in the AWS Well-Architected Framework.

member account

All AWS accounts other than the management account that are part of an organization in AWS Organizations. An account can be a member of only one organization at a time.

MES

See manufacturing execution system.

Message Queuing Telemetry Transport (MQTT)

A lightweight, machine-to-machine (M2M) communication protocol, based on the publish/subscribe pattern, for resource-constrained IoT devices.

microservice

A small, independent service that communicates over well-defined APIs and is typically owned by small, self-contained teams. For example, an insurance system might include microservices that map to business capabilities, such as sales or marketing, or subdomains, such as purchasing, claims, or analytics. The benefits of microservices include agility, flexible scaling, easy deployment, reusable code, and resilience. For more information, see Integrating microservices by using AWS serverless services.

microservices architecture

An approach to building an application with independent components that run each application process as a microservice. These microservices communicate through a well-defined interface by using lightweight APIs. Each microservice in this architecture can be updated, deployed, and scaled to meet demand for specific functions of an application. For more information, see Implementing microservices on AWS.

Migration Acceleration Program (MAP)

An AWS program that provides consulting support, training, and services to help organizations build a strong operational foundation for moving to the cloud, and to help offset the initial cost of migrations. MAP includes a migration methodology for executing legacy migrations in a methodical way and a set of tools to automate and accelerate common migration scenarios.

migration at scale

The process of moving the majority of the application portfolio to the cloud in waves, with more applications moved at a faster rate in each wave. This phase uses the best practices and lessons learned from the earlier phases to implement a *migration factory* of teams, tools, and processes to streamline the migration of workloads through automation and agile delivery. This is the third phase of the AWS migration strategy.

migration factory

Cross-functional teams that streamline the migration of workloads through automated, agile approaches. Migration factory teams typically include operations, business analysts and owners,

migration engineers, developers, and DevOps professionals working in sprints. Between 20 and 50 percent of an enterprise application portfolio consists of repeated patterns that can be optimized by a factory approach. For more information, see the discussion of migration factories and the Cloud Migration Factory guide in this content set.

migration metadata

The information about the application and server that is needed to complete the migration. Each migration pattern requires a different set of migration metadata. Examples of migration metadata include the target subnet, security group, and AWS account.

migration pattern

A repeatable migration task that details the migration strategy, the migration destination, and the migration application or service used. Example: Rehost migration to Amazon EC2 with AWS Application Migration Service.

Migration Portfolio Assessment (MPA)

An online tool that provides information for validating the business case for migrating to the AWS Cloud. MPA provides detailed portfolio assessment (server right-sizing, pricing, TCO comparisons, migration cost analysis) as well as migration planning (application data analysis and data collection, application grouping, migration prioritization, and wave planning). The MPA tool (requires login) is available free of charge to all AWS consultants and APN Partner consultants.

Migration Readiness Assessment (MRA)

The process of gaining insights about an organization's cloud readiness status, identifying strengths and weaknesses, and building an action plan to close identified gaps, using the AWS CAF. For more information, see the migration readiness guide. MRA is the first phase of the AWS migration strategy.

migration strategy

The approach used to migrate a workload to the AWS Cloud. For more information, see the 7 Rs entry in this glossary and see Mobilize your organization to accelerate large-scale migrations.

ML

See machine learning.

modernization

Transforming an outdated (legacy or monolithic) application and its infrastructure into an agile, elastic, and highly available system in the cloud to reduce costs, gain efficiencies, and take advantage of innovations. For more information, see Strategy for modernizing applications in the AWS Cloud.

modernization readiness assessment

An evaluation that helps determine the modernization readiness of an organization's applications; identifies benefits, risks, and dependencies; and determines how well the organization can support the future state of those applications. The outcome of the assessment is a blueprint of the target architecture, a roadmap that details development phases and milestones for the modernization process, and an action plan for addressing identified gaps. For more information, see Evaluating modernization readiness for applications in the AWS Cloud.

monolithic applications (monoliths)

Applications that run as a single service with tightly coupled processes. Monolithic applications have several drawbacks. If one application feature experiences a spike in demand, the entire architecture must be scaled. Adding or improving a monolithic application's features also becomes more complex when the code base grows. To address these issues, you can use a microservices architecture. For more information, see Decomposing monoliths into microservices.

MPA

See Migration Portfolio Assessment.

MQTT

See Message Queuing Telemetry Transport.

multiclass classification

A process that helps generate predictions for multiple classes (predicting one of more than two outcomes). For example, an ML model might ask "Is this product a book, car, or phone?" or "Which product category is most interesting to this customer?"

mutable infrastructure

A model that updates and modifies the existing infrastructure for production workloads. For improved consistency, reliability, and predictability, the AWS Well-Architected Framework recommends the use of immutable infrastructure as a best practice.

O

OAC

See origin access control.

OAI

See origin access identity.

OCM

See organizational change management.

offline migration

A migration method in which the source workload is taken down during the migration process. This method involves extended downtime and is typically used for small, non-critical workloads.

OI

See operations integration.

OLA

See operational-level agreement.

online migration

A migration method in which the source workload is copied to the target system without being taken offline. Applications that are connected to the workload can continue to function during the migration. This method involves zero to minimal downtime and is typically used for critical production workloads.

OPC-UA

See Open Process Communications - Unified Architecture.

Open Process Communications - Unified Architecture (OPC-UA)

A machine-to-machine (M2M) communication protocol for industrial automation. OPC-UA provides an interoperability standard with data encryption, authentication, and authorization schemes.

operational-level agreement (OLA)

An agreement that clarifies what functional IT groups promise to deliver to each other, to support a service-level agreement (SLA).

operational readiness review (ORR)

A checklist of questions and associated best practices that help you understand, evaluate, prevent, or reduce the scope of incidents and possible failures. For more information, see Operational Readiness Reviews (ORR) in the AWS Well-Architected Framework.

operational technology (OT)

Hardware and software systems that work with the physical environment to control industrial operations, equipment, and infrastructure. In manufacturing, the integration of OT and information technology (IT) systems is a key focus for Industry 4.0 transformations.

operations integration (OI)

The process of modernizing operations in the cloud, which involves readiness planning, automation, and integration. For more information, see the operations integration guide.

organization trail

A trail that's created by AWS CloudTrail that logs all events for all AWS accounts in an organization in AWS Organizations. This trail is created in each AWS account that's part of the organization and tracks the activity in each account. For more information, see Creating a trail for an organization in the CloudTrail documentation.

organizational change management (OCM)

A framework for managing major, disruptive business transformations from a people, culture, and leadership perspective. OCM helps organizations prepare for, and transition to, new systems and strategies by accelerating change adoption, addressing transitional issues, and driving cultural and organizational changes. In the AWS migration strategy, this framework is called *people acceleration*, because of the speed of change required in cloud adoption projects. For more information, see the OCM guide.

origin access control (OAC)

In CloudFront, an enhanced option for restricting access to secure your Amazon Simple Storage Service (Amazon S3) content. OAC supports all S3 buckets in all AWS Regions, server-side encryption with AWS KMS (SSE-KMS), and dynamic PUT and DELETE requests to the S3 bucket.

origin access identity (OAI)

In CloudFront, an option for restricting access to secure your Amazon S3 content. When you use OAI, CloudFront creates a principal that Amazon S3 can authenticate with. Authenticated principals can access content in an S3 bucket only through a specific CloudFront distribution. See also OAC, which provides more granular and enhanced access control.

ORR

See operational readiness review.

OT

See operational technology.

outbound (egress) VPC

In an AWS multi-account architecture, a VPC that handles network connections that are initiated from within an application. The AWS Security Reference Architecture recommends setting up your Network account with inbound, outbound, and inspection VPCs to protect the two-way interface between your application and the broader internet.

P

permissions boundary

An IAM management policy that is attached to IAM principals to set the maximum permissions that the user or role can have. For more information, see Permissions boundaries in the IAM documentation.

personally identifiable information (PII)

Information that, when viewed directly or paired with other related data, can be used to reasonably infer the identity of an individual. Examples of PII include names, addresses, and contact information.

PII

See personally identifiable information.

playbook

A set of predefined steps that capture the work associated with migrations, such as delivering core operations functions in the cloud. A playbook can take the form of scripts, automated runbooks, or a summary of processes or steps required to operate your modernized environment.

PLC

See programmable logic controller.

PLM

See product lifecycle management.

policy

An object that can define permissions (see identity-based policy), specify access conditions (see resource-based policy), or define the maximum permissions for all accounts in an organization in AWS Organizations (see service control policy).

polyglot persistence

Independently choosing a microservice's data storage technology based on data access patterns and other requirements. If your microservices have the same data storage technology, they can encounter implementation challenges or experience poor performance. Microservices are more easily implemented and achieve better performance and scalability if they use the data store best adapted to their requirements. For more information, see Enabling data persistence in microservices.

portfolio assessment

A process of discovering, analyzing, and prioritizing the application portfolio in order to plan the migration. For more information, see Evaluating migration readiness.

predicate

A query condition that returns `true` or `false`, commonly located in a `WHERE` clause.

predicate pushdown

A database query optimization technique that filters the data in the query before transfer. This reduces the amount of data that must be retrieved and processed from the relational database, and it improves query performance.

preventative control

A security control that is designed to prevent an event from occurring. These controls are a first line of defense to help prevent unauthorized access or unwanted changes to your network. For more information, see Preventative controls in *Implementing security controls on AWS*.

principal

An entity in AWS that can perform actions and access resources. This entity is typically a root user for an AWS account, an IAM role, or a user. For more information, see *Principal* in Roles terms and concepts in the IAM documentation.

privacy by design

A system engineering approach that takes privacy into account through the whole development process.

private hosted zones

A container that holds information about how you want Amazon Route 53 to respond to DNS queries for a domain and its subdomains within one or more VPCs. For more information, see Working with private hosted zones in the Route 53 documentation.

proactive control

A security control designed to prevent the deployment of noncompliant resources. These controls scan resources before they are provisioned. If the resource is not compliant with the control, then it isn't provisioned. For more information, see the Controls reference guide in the AWS Control Tower documentation and see Proactive controls in *Implementing security controls on AWS*.

product lifecycle management (PLM)

The management of data and processes for a product throughout its entire lifecycle, from design, development, and launch, through growth and maturity, to decline and removal.

production environment

See environment.

programmable logic controller (PLC)

In manufacturing, a highly reliable, adaptable computer that monitors machines and automates manufacturing processes.

prompt chaining

Using the output of one LLM prompt as the input for the next prompt to generate better responses. This technique is used to break down a complex task into subtasks, or to iteratively refine or expand a preliminary response. It helps improve the accuracy and relevance of a model's responses and allows for more granular, personalized results.

pseudonymization

The process of replacing personal identifiers in a dataset with placeholder values. Pseudonymization can help protect personal privacy. Pseudonymized data is still considered to be personal data.

publish/subscribe (pub/sub)

A pattern that enables asynchronous communications among microservices to improve scalability and responsiveness. For example, in a microservices-based MES, a microservice can publish event messages to a channel that other microservices can subscribe to. The system can add new microservices without changing the publishing service.

Q

query plan

A series of steps, like instructions, that are used to access the data in a SQL relational database system.

query plan regression

When a database service optimizer chooses a less optimal plan than it did before a given change to the database environment. This can be caused by changes to statistics, constraints, environment settings, query parameter bindings, and updates to the database engine.

R

RACI matrix

See responsible, accountable, consulted, informed (RACI).

RAG

See Retrieval Augmented Generation.

ransomware

A malicious software that is designed to block access to a computer system or data until a payment is made.

RASCI matrix

See responsible, accountable, consulted, informed (RACI).

RCAC

See row and column access control.

read replica

A copy of a database that's used for read-only purposes. You can route queries to the read replica to reduce the load on your primary database.

re-architect

See 7 Rs.

recovery point objective (RPO)

The maximum acceptable amount of time since the last data recovery point. This determines what is considered an acceptable loss of data between the last recovery point and the interruption of service.

recovery time objective (RTO)

The maximum acceptable delay between the interruption of service and restoration of service.

refactor

See 7 Rs.

Region

A collection of AWS resources in a geographic area. Each AWS Region is isolated and independent of the others to provide fault tolerance, stability, and resilience. For more information, see Specify which AWS Regions your account can use.

regression

An ML technique that predicts a numeric value. For example, to solve the problem of "What price will this house sell for?" an ML model could use a linear regression model to predict a house's sale price based on known facts about the house (for example, the square footage).

rehost

See 7 Rs.

release

In a deployment process, the act of promoting changes to a production environment.

relocate

See 7 Rs.

replatform

See 7 Rs.

repurchase

See 7 Rs.

resiliency

An application's ability to resist or recover from disruptions. High availability and disaster recovery are common considerations when planning for resiliency in the AWS Cloud. For more information, see AWS Cloud Resilience.

resource-based policy

A policy attached to a resource, such as an Amazon S3 bucket, an endpoint, or an encryption key. This type of policy specifies which principals are allowed access, supported actions, and any other conditions that must be met.

responsible, accountable, consulted, informed (RACI) matrix

A matrix that defines the roles and responsibilities for all parties involved in migration activities and cloud operations. The matrix name is derived from the responsibility types defined in the matrix: responsible (R), accountable (A), consulted (C), and informed (I). The support (S) type is optional. If you include support, the matrix is called a *RASCI matrix*, and if you exclude it, it's called a *RACI matrix*.

responsive control

A security control that is designed to drive remediation of adverse events or deviations from your security baseline. For more information, see Responsive controls in *Implementing security controls on AWS*.

retain

See 7 Rs.

retire

See 7 Rs.

Retrieval Augmented Generation (RAG)

A generative AI technology in which an LLM references an authoritative data source that is outside of its training data sources before generating a response. For example, a RAG model might perform a semantic search of an organization's knowledge base or custom data. For more information, see What is RAG.

rotation

The process of periodically updating a secret to make it more difficult for an attacker to access the credentials.

row and column access control (RCAC)

The use of basic, flexible SQL expressions that have defined access rules. RCAC consists of row permissions and column masks.

RPO

See recovery point objective.

RTO

See recovery time objective.

runbook

A set of manual or automated procedures required to perform a specific task. These are typically built to streamline repetitive operations or procedures with high error rates.

S

SAML 2.0

An open standard that many identity providers (IdPs) use. This feature enables federated single sign-on (SSO), so users can log into the AWS Management Console or call the AWS API operations without you having to create user in IAM for everyone in your organization. For more information about SAML 2.0-based federation, see About SAML 2.0-based federation in the IAM documentation.

SCADA

See supervisory control and data acquisition.

SCP

See service control policy.

secret

In AWS Secrets Manager, confidential or restricted information, such as a password or user credentials, that you store in encrypted form. It consists of the secret value and its metadata.

The secret value can be binary, a single string, or multiple strings. For more information, see What's in a Secrets Manager secret? in the Secrets Manager documentation.

security by design

A system engineering approach that takes security into account through the whole development process.

security control

A technical or administrative guardrail that prevents, detects, or reduces the ability of a threat actor to exploit a security vulnerability. There are four primary types of security controls: preventative, detective, responsive, and proactive.

security hardening

The process of reducing the attack surface to make it more resistant to attacks. This can include actions such as removing resources that are no longer needed, implementing the security best practice of granting least privilege, or deactivating unnecessary features in configuration files.

security information and event management (SIEM) system

Tools and services that combine security information management (SIM) and security event management (SEM) systems. A SIEM system collects, monitors, and analyzes data from servers, networks, devices, and other sources to detect threats and security breaches, and to generate alerts.

security response automation

A predefined and programmed action that is designed to automatically respond to or remediate a security event. These automations serve as detective or responsive security controls that help you implement AWS security best practices. Examples of automated response actions include modifying a VPC security group, patching an Amazon EC2 instance, or rotating credentials.

server-side encryption

Encryption of data at its destination, by the AWS service that receives it.

service control policy (SCP)

A policy that provides centralized control over permissions for all accounts in an organization in AWS Organizations. SCPs define guardrails or set limits on actions that an administrator can delegate to users or roles. You can use SCPs as allow lists or deny lists, to specify which services or actions are permitted or prohibited. For more information, see Service control policies in the AWS Organizations documentation.

service endpoint

The URL of the entry point for an AWS service. You can use the endpoint to connect programmatically to the target service. For more information, see AWS service endpoints in *AWS General Reference*.

service-level agreement (SLA)

An agreement that clarifies what an IT team promises to deliver to their customers, such as service uptime and performance.

service-level indicator (SLI)

A measurement of a performance aspect of a service, such as its error rate, availability, or throughput.

service-level objective (SLO)

A target metric that represents the health of a service, as measured by a service-level indicator.

shared responsibility model

A model describing the responsibility you share with AWS for cloud security and compliance. AWS is responsible for security *of* the cloud, whereas you are responsible for security *in* the cloud. For more information, see Shared responsibility model.

SIEM

See security information and event management system.

single point of failure (SPOF)

A failure in a single, critical component of an application that can disrupt the system.

SLA

See service-level agreement.

SLI

See service-level indicator.

SLO

See service-level objective.

split-and-seed model

A pattern for scaling and accelerating modernization projects. As new features and product releases are defined, the core team splits up to create new product teams. This helps scale your

organization's capabilities and services, improves developer productivity, and supports rapid innovation. For more information, see Phased approach to modernizing applications in the AWS Cloud.

SPOF

See single point of failure.

star schema

A database organizational structure that uses one large fact table to store transactional or measured data and uses one or more smaller dimensional tables to store data attributes. This structure is designed for use in a data warehouse or for business intelligence purposes.

strangler fig pattern

An approach to modernizing monolithic systems by incrementally rewriting and replacing system functionality until the legacy system can be decommissioned. This pattern uses the analogy of a fig vine that grows into an established tree and eventually overcomes and replaces its host. The pattern was introduced by Martin Fowler as a way to manage risk when rewriting monolithic systems. For an example of how to apply this pattern, see Modernizing legacy Microsoft ASP.NET (ASMX) web services incrementally by using containers and Amazon API Gateway.

subnet

A range of IP addresses in your VPC. A subnet must reside in a single Availability Zone.

supervisory control and data acquisition (SCADA)

In manufacturing, a system that uses hardware and software to monitor physical assets and production operations.

symmetric encryption

An encryption algorithm that uses the same key to encrypt and decrypt the data.

synthetic testing

Testing a system in a way that simulates user interactions to detect potential issues or to monitor performance. You can use Amazon CloudWatch Synthetics to create these tests.

system prompt

A technique for providing context, instructions, or guidelines to an LLM to direct its behavior. System prompts help set context and establish rules for interactions with users.

T

tags

Key-value pairs that act as metadata for organizing your AWS resources. Tags can help you manage, identify, organize, search for, and filter resources. For more information, see Tagging your AWS resources.

target variable

The value that you are trying to predict in supervised ML. This is also referred to as an *outcome variable*. For example, in a manufacturing setting the target variable could be a product defect.

task list

A tool that is used to track progress through a runbook. A task list contains an overview of the runbook and a list of general tasks to be completed. For each general task, it includes the estimated amount of time required, the owner, and the progress.

test environment

See environment.

training

To provide data for your ML model to learn from. The training data must contain the correct answer. The learning algorithm finds patterns in the training data that map the input data attributes to the target (the answer that you want to predict). It outputs an ML model that captures these patterns. You can then use the ML model to make predictions on new data for which you don't know the target.

transit gateway

A network transit hub that you can use to interconnect your VPCs and on-premises networks. For more information, see What is a transit gateway in the AWS Transit Gateway documentation.

trunk-based workflow

An approach in which developers build and test features locally in a feature branch and then merge those changes into the main branch. The main branch is then built to the development, preproduction, and production environments, sequentially.

trusted access

Granting permissions to a service that you specify to perform tasks in your organization in AWS Organizations and in its accounts on your behalf. The trusted service creates a service-linked role in each account, when that role is needed, to perform management tasks for you. For more information, see Using AWS Organizations with other AWS services in the AWS Organizations documentation.

tuning

To change aspects of your training process to improve the ML model's accuracy. For example, you can train the ML model by generating a labeling set, adding labels, and then repeating these steps several times under different settings to optimize the model.

two-pizza team

A small DevOps team that you can feed with two pizzas. A two-pizza team size ensures the best possible opportunity for collaboration in software development.

U

uncertainty

A concept that refers to imprecise, incomplete, or unknown information that can undermine the reliability of predictive ML models. There are two types of uncertainty: *Epistemic uncertainty* is caused by limited, incomplete data, whereas *aleatoric uncertainty* is caused by the noise and randomness inherent in the data. For more information, see the Quantifying uncertainty in deep learning systems guide.

undifferentiated tasks

Also known as *heavy lifting*, work that is necessary to create and operate an application but that doesn't provide direct value to the end user or provide competitive advantage. Examples of undifferentiated tasks include procurement, maintenance, and capacity planning.

upper environments

See environment.

V

vacuuming

A database maintenance operation that involves cleaning up after incremental updates to reclaim storage and improve performance.

version control

Processes and tools that track changes, such as changes to source code in a repository.

VPC peering

A connection between two VPCs that allows you to route traffic by using private IP addresses. For more information, see What is VPC peering in the Amazon VPC documentation.

vulnerability

A software or hardware flaw that compromises the security of the system.

W

warm cache

A buffer cache that contains current, relevant data that is frequently accessed. The database instance can read from the buffer cache, which is faster than reading from the main memory or disk.

warm data

Data that is infrequently accessed. When querying this kind of data, moderately slow queries are typically acceptable.

window function

A SQL function that performs a calculation on a group of rows that relate in some way to the current record. Window functions are useful for processing tasks, such as calculating a moving average or accessing the value of rows based on the relative position of the current row.

workload

A collection of resources and code that delivers business value, such as a customer-facing application or backend process.

workstream

Functional groups in a migration project that are responsible for a specific set of tasks. Each workstream is independent but supports the other workstreams in the project. For example, the portfolio workstream is responsible for prioritizing applications, wave planning, and collecting migration metadata. The portfolio workstream delivers these assets to the migration workstream, which then migrates the servers and applications.

WORM

See write once, read many.

WQF

See AWS Workload Qualification Framework.

write once, read many (WORM)

A storage model that writes data a single time and prevents the data from being deleted or modified. Authorized users can read the data as many times as needed, but they cannot change it. This data storage infrastructure is considered immutable.

Z

zero-day exploit

An attack, typically malware, that takes advantage of a zero-day vulnerability.

zero-day vulnerability

An unmitigated flaw or vulnerability in a production system. Threat actors can use this type of vulnerability to attack the system. Developers frequently become aware of the vulnerability as a result of the attack.

zero-shot prompting

Providing an LLM with instructions for performing a task but no examples (*shots*) that can help guide it. The LLM must use its pre-trained knowledge to handle the task. The effectiveness of zero-shot prompting depends on the complexity of the task and the quality of the prompt. See also few-shot prompting.

zombie application

An application that has an average CPU and memory usage below 5 percent. In a migration project, it is common to retire these applications.