Migrating Oracle databases to the AWS Cloud

AWS Prescriptive Guidance

AWS Prescriptive Guidance: Migrating Oracle databases to the AWS Cloud

Table of Contents

Migrating Oracle databases to the AWS Cloud

Sagar Patel, Amazon Web Services (AWS)

July 2023 (document history)

Amazon Web Services (AWS) provides a comprehensive set of services and tools for deploying Oracle Database on the reliable and secure AWS Cloud infrastructure. This guide explains the options available for migrating your Oracle on-premises databases to the AWS Cloud. It also dives into the best practices and scenarios for exercising these migration options.

This guide is for program or project managers, product owners, database administrators, database engineers, and operations or infrastructure managers who are planning to migrate their on-premises Oracle databases to AWS.

Overview

Before you migrate your Oracle databases to AWS, you should understand and evaluate your migration strategy by using the framework discussed in Migration Strategy for Relational Databases.

The first step is to perform an analysis of your application and Oracle Database workloads to understand the complexity, compatibility, and cost of migration. Here are some of the top points you should consider when you plan to migrate:

- Check the database current size and overall capacity growth. For example, if you're planning to migrate your Oracle database to Amazon Relational Database Service (Amazon RDS) or Amazon RDS Custom, you can create DB instances with up to 64 TiB of storage. For the latest information, see Amazon RDS DB instance storage in the Amazon RDS documentation.
- Review Oracle Automatic Workload Repository (AWR) reports to check the resource usage and database health of your on-premises database.
- Check current database dependencies on other databases. If your database is dependent on other databases, you can either migrate them together or create dependencies after you migrate your main database.
- Check for application dependencies. If your database supports legacy, custom, or packaged applications, Amazon RDS Custom for Oracle might be a good choice. This service lets you retain control over database configurations, shared file systems, and operating system patches.

- Determine the IOPS and throughput of your databases. If you're planning to migrate to Amazon RDS, consider the I/O performance of Amazon RDS DB instances.
- Review your current architecture and auditing or compliance needs, to make sure you can satisfy these requirements after moving to either Amazon RDS or Amazon Elastic Compute Cloud (Amazon EC2).
- Check the version and edition of your Oracle Database software to make sure it's supported if you're planning to move to Amazon RDS for Oracle (see currently supported versions for Amazon RDS and Amazon RDS Custom).
- Check the network connectivity between your on-premises environment and AWS, to make sure that it provides enough bandwidth for fast transfers of data between on premises and AWS.
- Determine the amount of downtime you have available for migration so you can plan your migration approach and decide whether you want to use online or offline migration.
- Identify your recovery time objective (RTO), recovery point objective (RPO), and service-level agreement (SLA) requirements for your existing database workloads.
- Check the chipset endian platform of the database workload. AWS supports x86-x64 little-endian platforms. Other platforms, such as Sun SPARC, HP Tru64, or IBM zSeries-based big-endian platforms, require cross-platform migration.
- AWS supports Linux (32-bit and 64-bit) and Windows operating systems. It doesn't support Solaris, HP-UX, or IBM AIX operating systems, which are commonly used for Oracle databases. Migrating Oracle databases from these operating systems requires platform conversion.

Oracle database migration strategies

At a high level, there are two options for migrating an Oracle database from on premises to the AWS Cloud: either stay on Oracle (*homogeneous migration*) or move off Oracle (*heterogeneous migration*). In a homogeneous migration, you don't change the database engine (that is, your target database is also an Oracle database). In a heterogeneous migration, you switch either to an open-source database engine such as MySQL, PostgreSQL, or MariaDB, or to an AWS Cloud-native database such as Amazon Aurora, Amazon DynamoDB, or Amazon RedShift.

There are three common strategies for migrating your Oracle databases to AWS: rehost, replatform, and re-architect (refactor). These are part of the 7 Rs of application migration strategies and are described in the following table.

Strategy	Type	When to choose	Example
Rehost	Homogeneous	You want to migrate your Oracle database as is, with or without changing the operating system, database software, or configuration.	Oracle Database to Amazon EC2 (Browse rehost patterns)
Replatform	Homogeneous	You want to reduce the time you spend managing database instances by using a database-as-a-service (DBaaS) offering.	Oracle Database to Amazon RDS for Oracle (Browse replatform patterns)
Re-architect (refactor)	Heterogeneous	You want to restructure, rewrite, and re-architect your database and application to take advantage of open-	Oracle Database to Amazon Aurora PostgreSQL, MySQL, or MariaDB (Browse re-architect patterns)

source and cloud-nat
ive database features.

Choosing the right migration strategy

Choosing the correct strategy depends on your business requirements, your resource constraints, your migration timeframe, and cost considerations. The following diagram shows the effort and complexity involved in migrations, including six of the strategies.

Refactoring your Oracle database and migrating to an open-source or AWS Cloud-native database such as Amazon Aurora PostgreSQL-Compatible Edition or Amazon Aurora MySQL-Compatible Edition can help you modernize and optimize your database. By moving to an open-source database, you can avoid expensive licenses (resulting in lower costs), vendor lock-in periods, and audits, and you won't have to pay additional fees for new features. However, depending on the complexity of your workload, refactoring your Oracle database can be a complicated, time-consuming, and resource-intensive effort.

To reduce complexity, instead of migrating your database in a single step, you might consider a phased approach. In the first phase, you can focus on core database functionality. In the next phase, you can integrate additional AWS services into your cloud environment, to reduce costs, and to optimize performance, productivity, and compliance. For example, if your goal is to replace your on-premises Oracle database with Aurora PostgreSQL-Compatible, you might consider rehosting your database on Amazon EC2 or replatforming your database on Amazon RDS for Oracle in the first phase, and then refactor to Aurora PostgreSQL-Compatible in a subsequent phase. This approach helps reduce costs, resources, and risks during the migration phase and focuses on optimization and modernization in the second phase.

Online and offline migration

You can use two methods to migrate Oracle Database from an on-premises environment to the AWS Cloud, based on your migration timeline and how much downtime you can allow: online migration or offline migration.

- **Offline migration:** This method is used when your application can afford a planned downtime. In offline migration, the source database is offline during the migration period. While the source database is offline, it is migrated over to the target database on AWS. After the migration is complete, validation and verification checks are performed to ensure data consistency with the source database. When the database passes all validation checks, you perform a cutover to AWS by connecting your application to the target database on AWS.
- **Online migration**: This method is used when your application requires near zero to minimal downtime. In 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 the cutover, the application switches its connections over to the target database on AWS, leaving no connections to the source database. You can use AWS Database Migration Service (AWS DMS), Oracle GoldenGate, Quest SharePlex, or tools available from AWS Marketplace (such as Attunity) to synchronize the source and target databases.

Homogeneous database migration for Oracle databases

AWS offers you the ability to run Oracle Database in a cloud environment. For developers and database administrators, running Oracle Database in the AWS Cloud is very similar to running Oracle Database in a data center. This section describes options for migrating Oracle Database from an on-premises environment or a data center to the AWS Cloud.

AWS offers four options for running Oracle Database on AWS, as described in the following table.

Option	Highlights	More information
Oracle Database on Amazon RDS	Managed service, provides easy provisioning and licensing	Amazon RDS for Oracle section
Oracle Database on Amazon RDS Custom	Managed service, but you retain administrative rights to the database and the underlying operating system	Amazon RDS Custom for Oracle section
Oracle Database on Amazon EC2	Self-managed, provides full control and flexibility	Amazon EC2 for Oracle section
Oracle Database on VMware Cloud on AWS	Minimal disruption, easy to manage	VMware Cloud on AWS for Oracle section

Notice

As of April 30, 2024, VMware Cloud on AWS is no longer resold by AWS or its channel partners. The service will continue to be available through Broadcom. We encourage you to reach out to your AWS representative for details.

Your application requirements, database features, functionality, growth capacity, and overall architecture complexity will determine which option to choose. If you are migrating multiple Oracle databases to AWS, some of them might be a great fit for Amazon RDS whereas others might be better suited to run directly on Amazon EC2. You might have databases that are running on

Oracle Enterprise Edition (EE) but are a good fit for Oracle Standard Edition Two (SE2). You can save on cost and licenses for those databases. Many AWS customers run multiple Oracle Database workloads across Amazon RDS, Amazon EC2, and VMware Cloud on AWS. If you're moving to Amazon RDS Custom, make sure to review the requirements and limitations for Amazon RDS Custom for Oracle.

Amazon RDS for Oracle

Amazon RDS for Oracle is a managed database service that simplifies the provisioning and management of Oracle Database on AWS. Amazon RDS makes it easy to set up, operate, and scale Oracle Database deployments in the cloud. You can deploy your database in minutes and choose either General Purpose (SSD) storage or Provisioned IOPS storage. (For details, see Amazon RDS storage types in the AWS documentation.)

Amazon RDS frees you up to focus on application development, because it manages time-consuming database administration tasks, including provisioning, backups, software patching, monitoring, and hardware scaling. Amazon RDS for Oracle easily provisions read replica and Multi-AZ databases to enhance availability, performance, and reliability for production workloads.

For more information about migrating from Oracle to Amazon RDS, see the replatform patterns on the AWS Prescriptive Guidance website.

When to choose Amazon RDS

Amazon RDS for Oracle is a good migration option when:

- You want to focus on your business and applications, and you want AWS to take care of undifferentiated heavy lifting tasks such as the provisioning of the database, management of backup and recovery tasks, management of security patches, minor Oracle version upgrades, and storage management.
- You need a highly available database solution, and you want to take advantage of the push-button, synchronous Multi-AZ replication offered by Amazon RDS, without having to manually set up and maintain a standby database.
- You want to have synchronous replication to a standby instance, to provide high availability for your Oracle Database Standard Edition One (SE1) or Standard Edition Two (SE2) database, instead of having to pay for Oracle Database Enterprise Edition (EE).
- You want to pay for the Oracle license as part of the instance cost on an hourly basis instead of making a large, upfront investment.

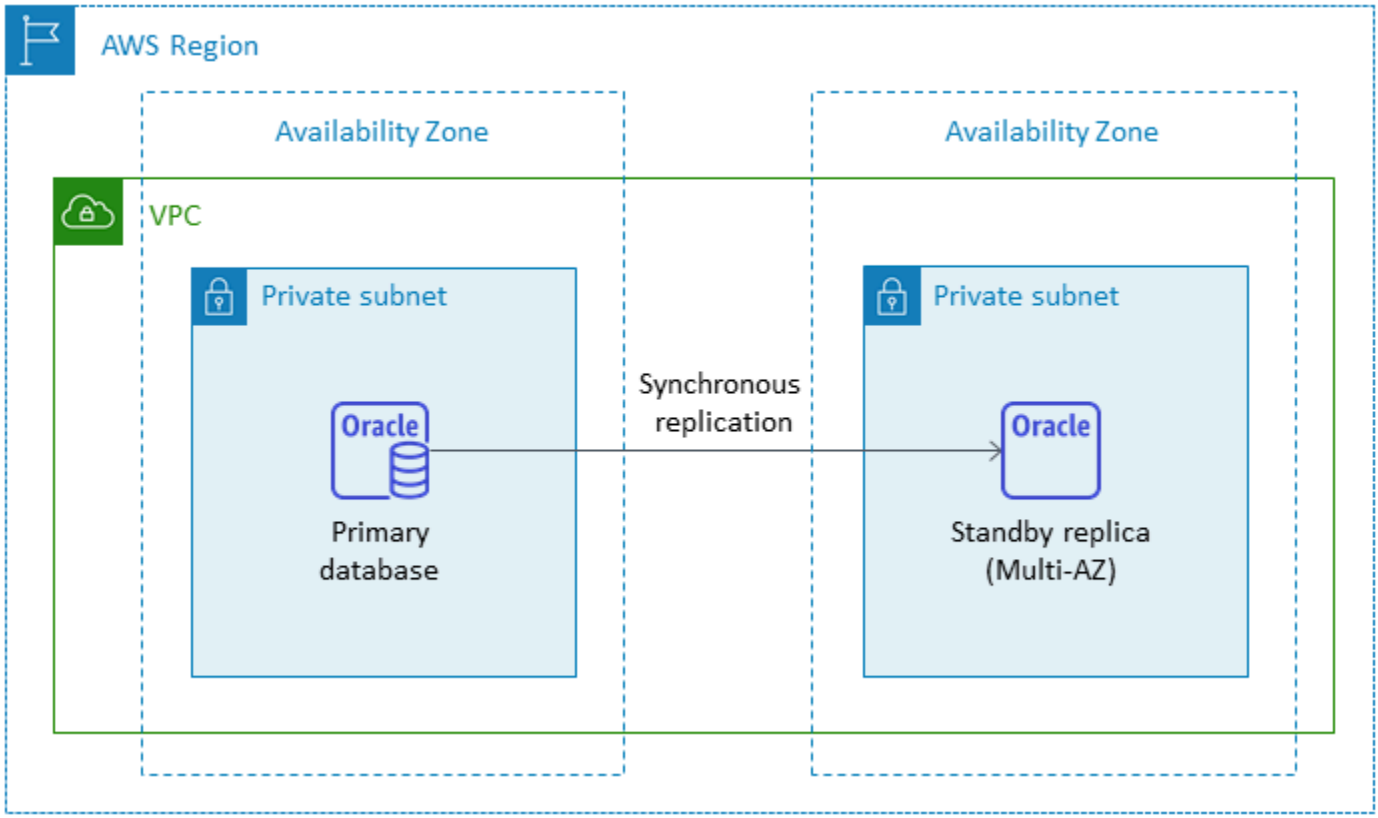
- Your database size and IOPS needs are supported by Amazon RDS for Oracle. See Amazon RDS DB instance storage in the AWS documentation for the current maximum limits.
- You don't want to manage backups or point-in-time recoveries of your database.
- You would rather focus on high-level tasks, such as performance tuning and schema optimization, instead of the daily administration of the database.
- You want to scale the instance type up or down based on your workload patterns without being concerned about licensing complexities.

After assessing your database and project requirements, if you decide to migrate to Amazon RDS for Oracle, see the details provided in the following sections, and review the migration best practices we discuss later in this guide.

High availability

Amazon RDS provides high availability and failover support for databases that are deployed with the Multi-AZ option. When you provision your database with the Multi-AZ option, Amazon RDS automatically provisions and maintains a synchronous standby replica in a different Availability Zone. The primary database synchronously replicates the data to the standby replica across Availability Zones. In case of infrastructure failure or Availability Zone disruption, Amazon RDS performs an automatic failover to the standby replica so you can resume database operations as soon as the failover is complete. This provides high redundancy, durability, and enhanced availability of your primary database. It also offloads your primary database by taking automated backups from the standby replica. For more information, see High Availability (Multi-AZ) for Amazon RDS in the AWS documentation.

The following diagram illustrates the Amazon RDS for Oracle Multi-AZ deployment option. The database application and users connect to the primary Oracle database, and all changes are synchronously replicated to the secondary database, which is in a different Availability Zone. The secondary database is not available to users until the failover is complete. After failover, the endpoint remains the same, so users and database applications can resume database operations without any manual intervention.



Read replicas

A read replica is a special type of Amazon RDS for Oracle DB instance that helps reduce the load on your primary DB instance. Updates made to your primary DB instance are asynchronously copied to the read replica, which you can set up in the same AWS Region or in another AWS Region.

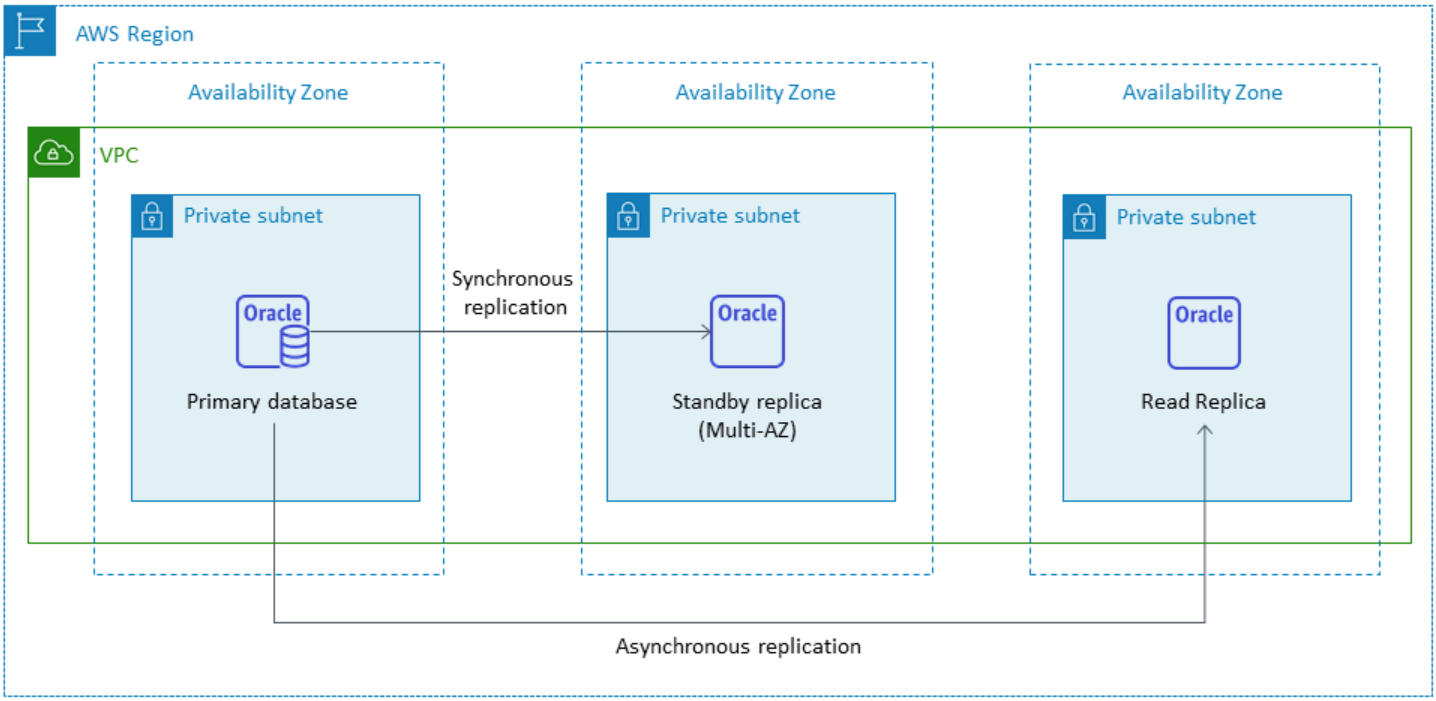
You can provision an Amazon RDS for Oracle database with read replicas by using Oracle Active Data Guard to offload your read-only workload from the primary Oracle database. Oracle Active Data Guard replicates database changes from the source DB instance to the read replicas. This feature supports managed disaster recovery for mission-critical databases by allowing a read replica in another AWS Region to be promoted as a new, standalone, production database. You can provision up to five read replicas for your Amazon RDS for Oracle database.

Amazon RDS for Oracle makes it easy to create the read replicas by managing the configuration of Active Data Guard and maintaining secure network connections between a primary DB instance and its read replicas. For more information, see Working with read replicas for Amazon RDS for Oracle in the Amazon RDS documentation.

To use the read replica feature, you must use the Bring Your Own License (BYOL) model with Oracle Database Enterprise Edition (EE) and also have an Active Data Guard license.

Using a read replica in the same AWS Region

The following diagram illustrates an Amazon RDS for Oracle DB instance in a Multi-AZ environment with a read replica in another Availability Zone within the same AWS Region. Not all AWS Regions offer more than two Availability Zones, so you should check the Region you're planning to use before adopting this strategy.



Using a read replica in another AWS Region

Amazon RDS for Oracle also supports cross-Region read replicas. It uses Oracle Active Data Guard to create and manage the configuration of physical standby DB instances in different AWS Regions from the primary DB instance. It replicates data over secure network connections between a primary DB instance and its read replicas across Regions.

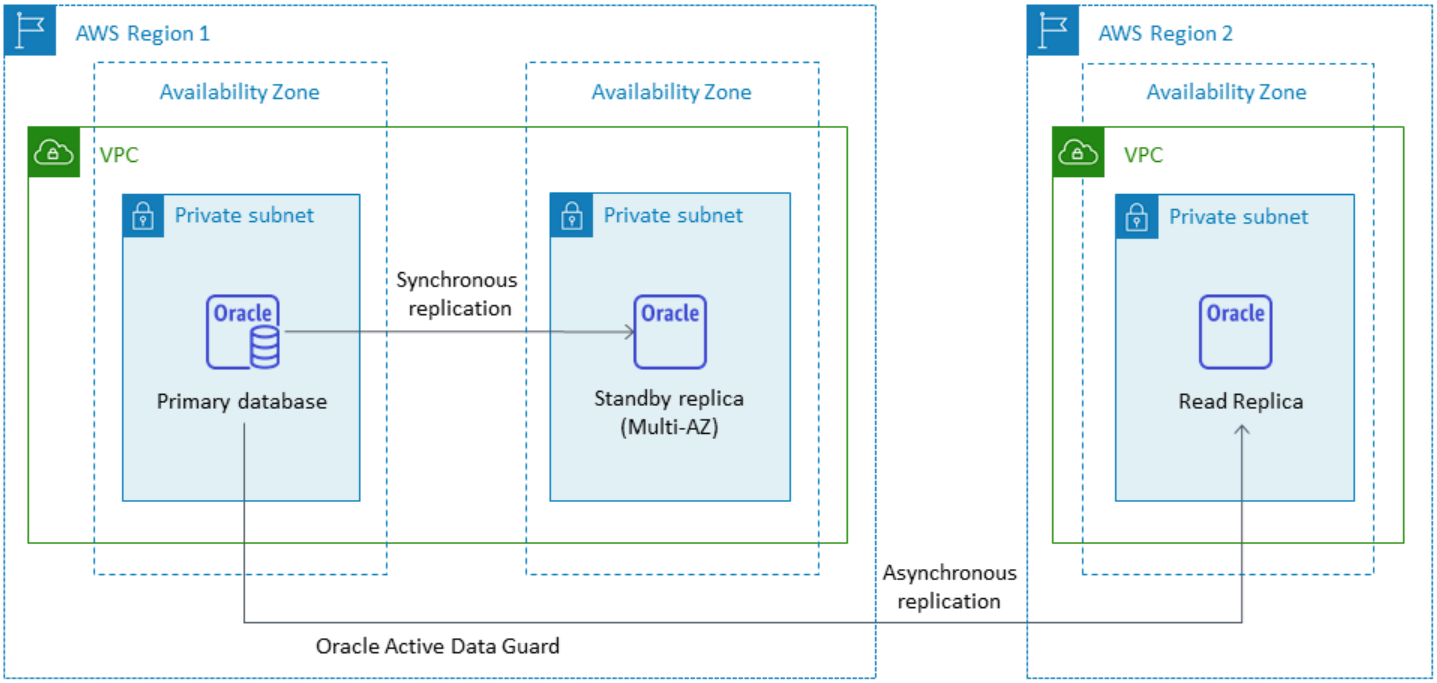
Cross-Region read replicas provide:

- High availability and data protection against single-Region failure.
- The ability to scale read operations to another AWS Region that's closer to your application's users.

You can promote an Oracle read replica to a standalone DB instance explicitly, or you can promote it implicitly by deleting its source DB instance. When you promote a read replica, the DB instance

is rebooted before it becomes available. The promoted read replica behaves the same as any other Oracle DB instance.

The following diagram shows the configuration of Amazon RDS for Oracle cross-Region read replicas.



The data transferred for cross-Region replication incurs Amazon RDS data transfer charges.

For more information about using read replicas, see Working with DB instance read replicas and Working with read replicas for Amazon RDS for Oracle in the AWS documentation. For more information about data transfer pricing, see Amazon RDS pricing.

Amazon RDS Custom for Oracle

If you're unable to move to a fully managed service such as Amazon RDS because of customization requirements, you can migrate to Amazon RDS Custom for Oracle. With Amazon RDS Custom, you can retain administrative rights to the database and its underlying operating system.

When to choose Amazon RDS Custom for Oracle

Amazon RDS Custom for Oracle is a good migration option when:

- You have legacy, custom, and packaged applications that require access to the underlying operating system and database environment.

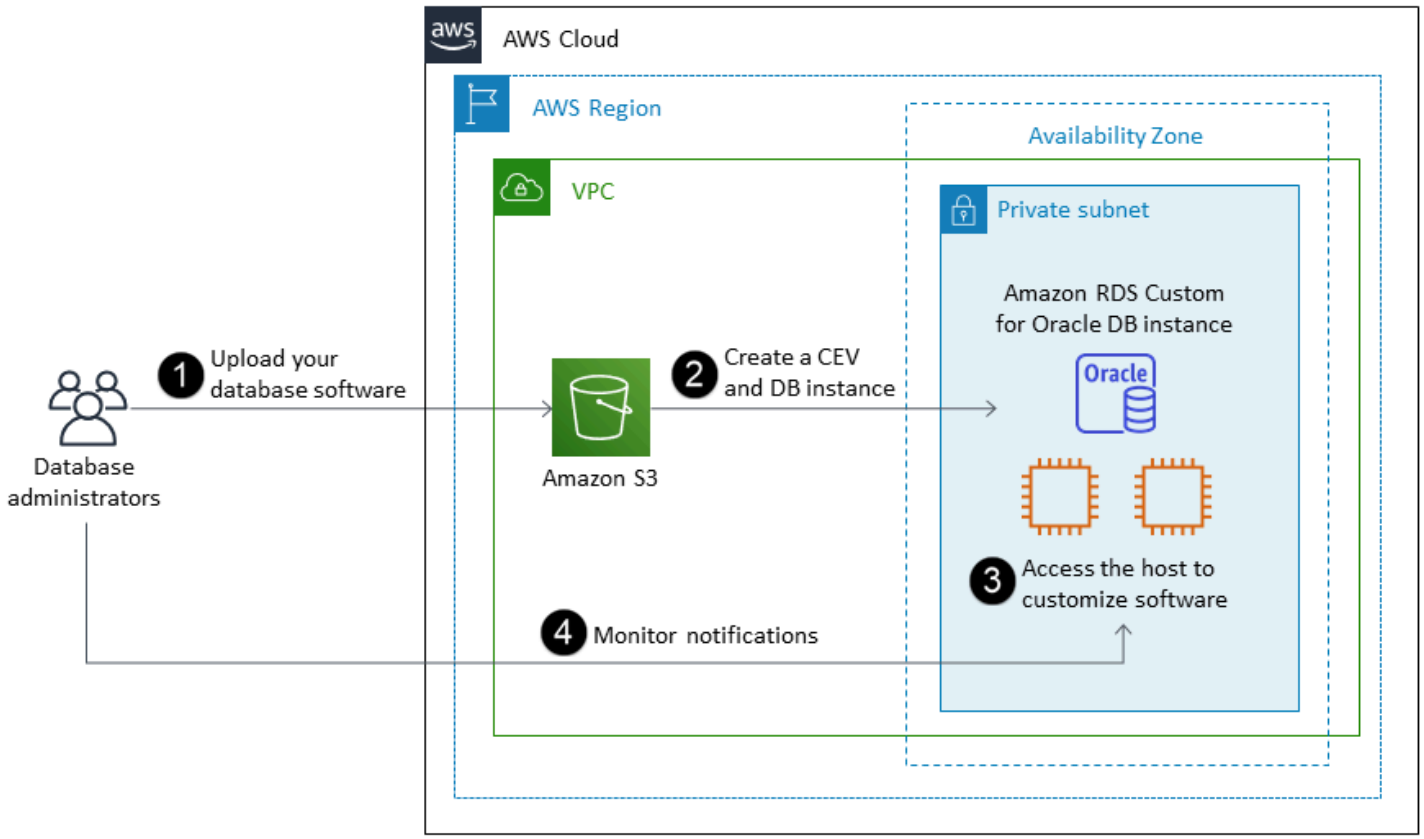
- You need access to SYS or SYSTEM user to meet vendor-based application deployment requirements.
- You need access to the underlying operating system to configure settings, install patches, and enable native features to meet the dependent application's requirements.
- You want to access and customize the database environment (by applying custom database patches or modifying operating system packages) to meet your database and application needs.

How it works

To use Amazon RDS Custom for Oracle, you follow these steps, which are illustrated in the following diagram:

1. Upload your database software to an Amazon Simple Storage Service (Amazon S3) bucket.
2. Create a custom engine version (CEV) and DB instance.
3. Connect your application to the DB instance endpoint and access the host to customize your software.
4. Monitor the notifications generated by Amazon RDS Custom automation.

For more information about these steps, see the Amazon RDS Custom documentation.



To provision your Amazon RDS Custom for Oracle DB instance, review the requirements in the Amazon RDS Custom for Oracle documentation. Amazon RDS Custom currently supports Oracle Database Enterprise Edition with the Oracle Database versions listed in the documentation. Amazon RDS Custom is available on the Oracle Linux operating system.

In Amazon RDS Custom for Oracle, you use your own media, patches, and Oracle licenses. When you create a custom engine version (CEV), Amazon RDS Custom installs the media that you provide. You have access to the underlying EC2 instance that hosts the DB engine. You can access the EC2 instance by using Secure Shell (SSH) or AWS Systems Manager and perform your customizations.

You can also install software to run custom applications and agents. Because you have privileged access to the host, you can modify file systems to support legacy applications. You can also apply custom database patches or modify operating system packages on your Amazon RDS Custom DB instances.

Amazon RDS Custom automatically provides monitoring, backups, and instance recovery, and ensures that your DB instance uses a supported AWS infrastructure, operating system, and database. If you want to customize your instance, you can pause Amazon RDS Custom automation

for up to 24 hours and then resume it when your customization work is complete. Pausing the automation prevents Amazon RDS automation from directly interfering with your customizations.

When you resume automation, the support perimeter determines whether your customization of the database or operating system environment interferes with, or breaks, Amazon RDS Custom automation. Amazon RDS Custom supports your customization of the host and database environment as long as your changes don't put the DB instance outside the support perimeter. The support perimeter checks are performed every 30 minutes by default, and also occur after events such as snapshot deletions or uninstalling the Amazon RDS Custom agent, which monitors the DB instance. The Amazon RDS Custom agent is a critical component for ensuring Amazon RDS Custom functionality. If you uninstall the agent, Amazon RDS Custom runs the support perimeter check after one minute and moves the DB instance outside the support perimeter.

Amazon RDS Custom for Oracle is supported in a limited selection of AWS Regions and with limited DB instance classes. It is available on the Oracle Linux operating system and currently supports Oracle Database Enterprise Edition with the Oracle Database versions listed in the documentation. For specifics, see RDS Custom for Oracle requirements and limitations in the AWS documentation.

For additional information, see the following resources:

- Amazon RDS Custom for Oracle – New Control Capabilities in Database Environment (AWS News blog)
- Using Amazon RDS for Oracle cross-Region automated backups to enhance your DR capabilities:
  - Managed disaster recovery with Amazon RDS for Oracle cross-Region automated backups – Part 1 (AWS Database blog)
  - Managed disaster recovery with Amazon RDS for Oracle cross-Region automated backups – Part 2 (AWS Database blog)
- Migrating from an on-premises or self-managed Oracle database to Amazon RDS Custom for Oracle by using native tools:
  - Physical migration of Oracle databases to Amazon RDS Custom using Data Guard (AWS Database blog)
  - Physical migration of Oracle databases to Amazon RDS Custom using RMAN duplication (AWS Database blog)
- Integrating an Amazon Elastic File System (Amazon EFS) shared file system with Amazon RDS for Oracle to share files between the database and application servers or as a staging location to keep backups and data loads: Integrate Amazon RDS Custom for Oracle with Amazon EFS (AWS Database blog)

Amazon EC2 for Oracle

Amazon EC2 supports a self-managed Oracle database—that is, it gives you full control over the setup of the infrastructure and the database environment. Running the database on Amazon EC2 is very similar to running the database on your own server. You have full control of the database and operating system-level access, so you can use your choice of tools to manage the operating system, database software, patches, data replication, backup, and restoration. This migration option requires you to set up, configure, manage, and tune all the components, including Amazon EC2 instances, storage volumes, scalability, networking, and security, based on AWS architecture best practices.

For more information about migrating from Oracle to Amazon EC2, see the rehost patterns on the AWS Prescriptive Guidance website.

When to choose Amazon EC2

Amazon EC2 is a good migration option for your Oracle database when:

- You need full control over the database and access to its underlying operating system.
- You want to control your backups, replication, and clustering.
- You want to use features and options that aren't currently supported by Amazon RDS. For details, see Oracle Database Feature Support in the Amazon RDS documentation.
- You need a specific Oracle Database version that isn't supported by Amazon RDS. For a list of supported version and editions, see Amazon RDS for Oracle in the Amazon RDS documentation.
- Your database size and performance needs exceed Amazon RDS offerings. For details, see Amazon RDS DB instance storage in the Amazon RDS documentation.
- You want to avoid automatic software patches that might not be compliant with your applications.
- You want to achieve higher IOPS and provision storage capacity than the current limits. For details, see Amazon RDS DB instance storage in the Amazon RDS documentation.

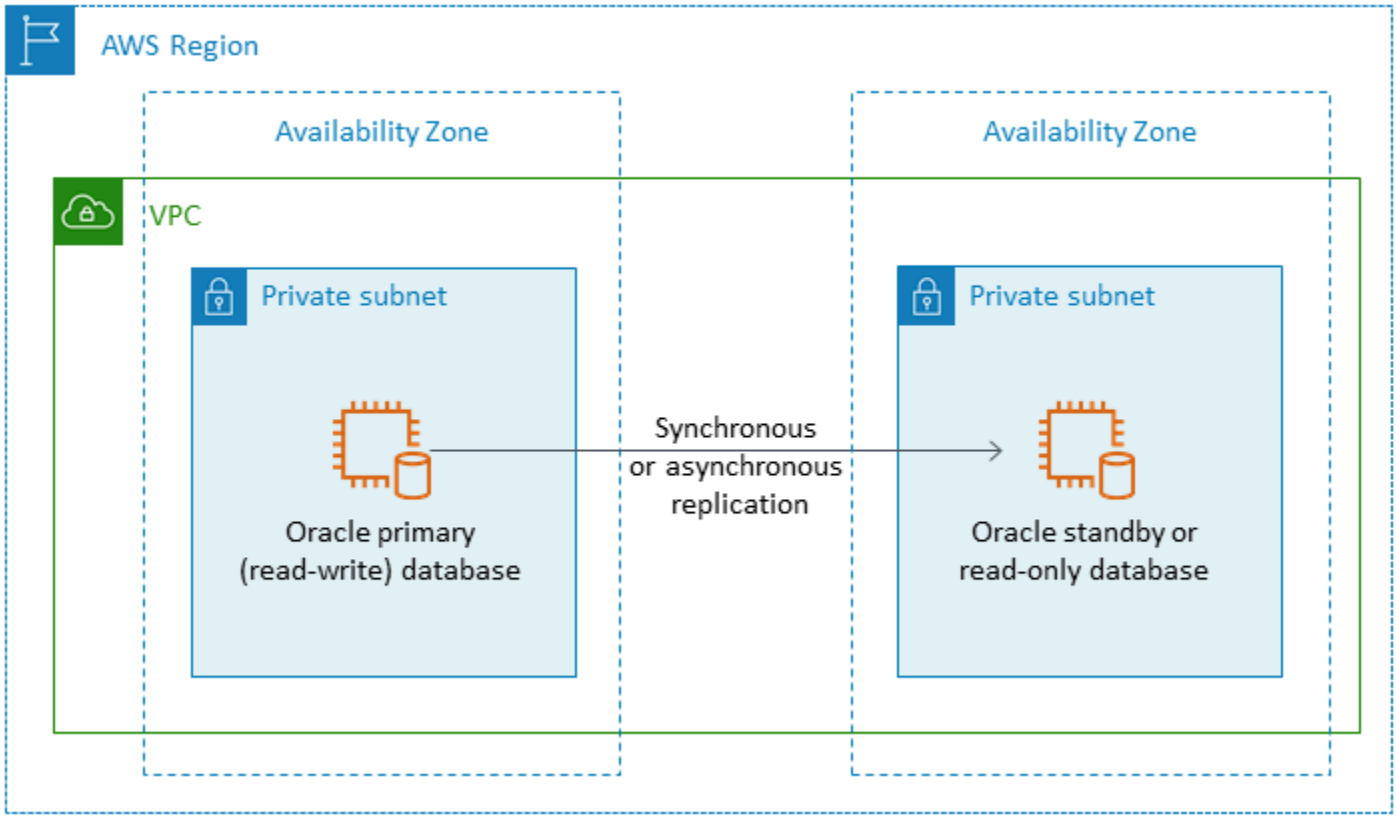
High availability

Oracle Database on Amazon EC2 can work with any Oracle-supported replication technology to achieve high availability and disaster recovery. Some of the common solutions are Oracle Data Guard, Oracle Active Data Guard, and Oracle GoldenGate.

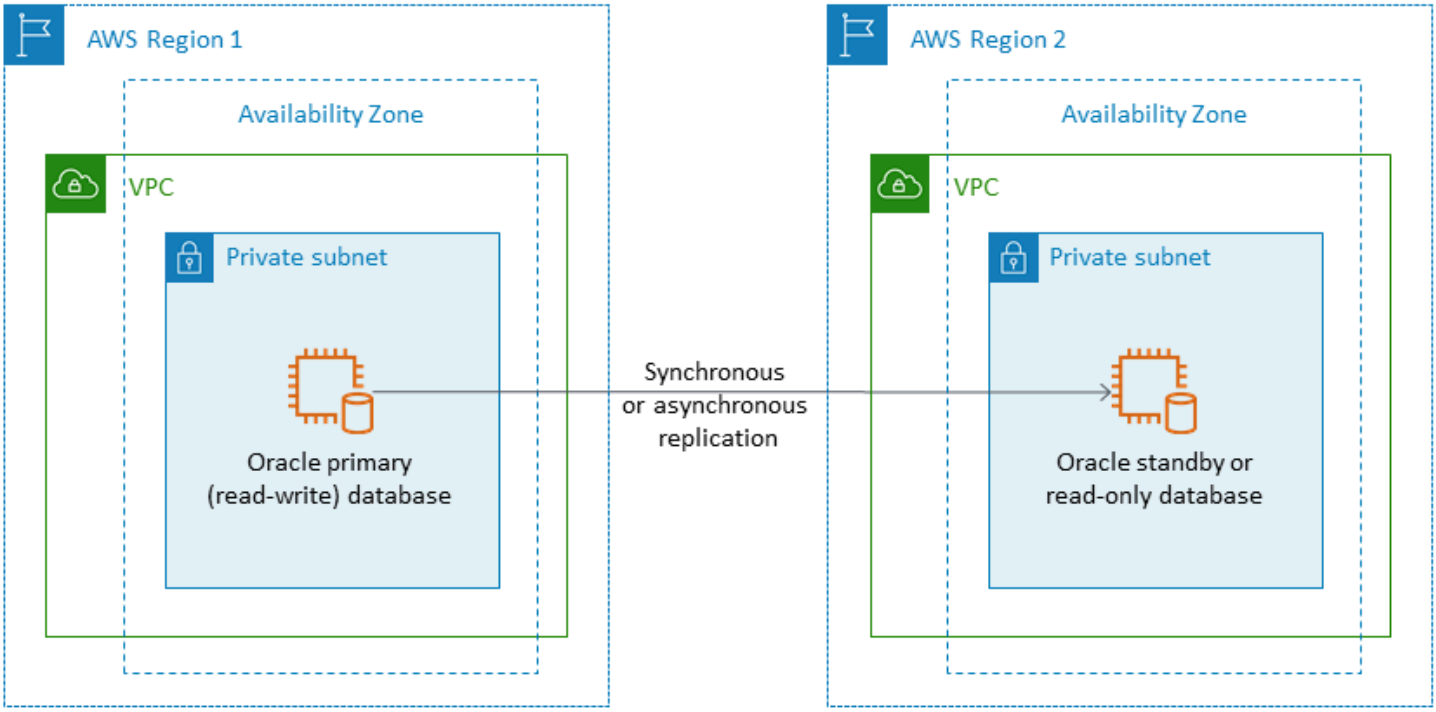
An Oracle database on Amazon EC2 uses Oracle Data Guard or Oracle Active Data Guard to achieve high availability, data protection, and disaster recovery.

- Oracle Data Guard provides a set of services for creating, maintaining, and managing standby databases, to help protect Oracle production databases against disasters and data corruption. Oracle Data Guard automatically maintains each standby database by transmitting redo changes from the primary database, and then applying the redo to the standby database. If the primary database goes down for any planned or unplanned outage, you can fail over to the standby database by converting it into a primary read-write database. Oracle Data Guard is included with Oracle Database Enterprise Edition (EE) only and doesn't require a separate license.
- Oracle Active Data Guard provides read-only access to a physical standby database for queries, sorting, reporting, and other read operations while it applies redo changes continuously from the primary database. Oracle Active Data Guard requires a separate license that must be additionally purchased with Oracle Database EE. Oracle Active Data Guard features include Real-Time Query, Automatic Block Repair, Far Sync, Standby Block Change Tracking, Active Data Guard Rolling Upgrade, Global Database Services, and Application Continuity.

The following diagram shows how you can use Oracle Database on Amazon EC2 in two Availability Zones within a single AWS Region. The primary database is a read-write database, and the standby database is configured with either Data Guard (physical standby with no read access) or Active Data Guard. All the redo data from the primary database is transferred and applied to the standby database asynchronously by default.



You can also use Oracle Data Guard or Oracle Active Data Guard to configure high availability and disaster recovery across multiple AWS Regions, using Oracle Database on Amazon EC2 for your primary database and standby database, as illustrated in the following diagram.



VMware Cloud on AWS for Oracle

> **Notice**
>
> As of April 30, 2024, VMware Cloud on AWS is no longer resold by AWS or its channel partners. The service will continue to be available through Broadcom. We encourage you to reach out to your AWS representative for details.

VMware Cloud on AWS is an integrated cloud offering jointly developed by AWS and VMware. When you migrate Oracle Database to VMware Cloud on AWS, you have full control of the database and operating system-level access, as with Amazon EC2. You can run advanced architectures like Oracle Real Application Cluster (RAC) and Oracle RAC extended clusters (across different Availability Zones) in VMware Cloud on AWS. You can choose from a number of migration methods and tools based on your needs and your existing system.

For online migrations, VMware technologies like VMware Hybrid Cloud Extension (VMware HCX) and HCX vMotion help you migrate VM workloads from on-premises VMware clusters to VMware Cloud on AWS. For offline migrations of Oracle workloads, you can use Oracle Recovery Manager (RMAN), AWS Snowball, AWS Storage Gateway, or VMware HCX.

When to choose VMware Cloud on AWS

VMware Cloud on AWS is a good option for your Oracle database when:

- Your Oracle databases are already running in an on-premises data center in vSphere virtualized environments.
- You need to run Oracle RAC in the cloud.
- You have a large number of databases and you need fast migration (for example, only a few hours) to the cloud without requiring any additional work from the migration team.

For more information, see the blog posts How to Migrate Oracle Workloads to VMware Cloud on AWS and Best Practices for Virtualizing Oracle RAC with VMware Cloud on AWS on the AWS Partner Network (APN) blog.

Tools for homogeneous database migrations

> **Notice**
>
> As of April 30, 2024, VMware Cloud on AWS is no longer resold by AWS or its channel partners. The service will continue to be available through Broadcom. We encourage you to reach out to your AWS representative for details.

A number of tools and technologies are available for data migration. You can perform the migration in a single phase or in multiple phases, based on your database size, consistency, the bandwidth of the network connection between your on-premises environment and AWS, and the allowed time for database migration. The following chart provides a list of tools and information to help you choose the option that best meets your needs.

Migration tool	Database size	Supports	Recommended for
Oracle SQL Developer (Database Copy feature)	Up to 200 MB	Amazon RDS Amazon EC2	Small databases with any number of objects.
Oracle SQL*Loader	Up to 10 GB	Amazon RDS Amazon EC2	Small to medium-size databases with a

			limited number of objects.
Oracle Export and Import Utilities	Up to 10 GB	Amazon RDS Amazon EC2	Small to medium-size databases with a large number of objects.
Oracle Data Pump	Up to 20 TB	Amazon RDS Amazon EC2	Preferred method for any database that's 10 GB – 20 TB in size.
AWS DMS	Any size	Amazon RDS Amazon EC2	Minimal downtime migration. Database size is limited by bandwidth. You can use AWS DMS with Oracle Data Pump for large database migrations.
Oracle GoldenGate	Any size	Amazon RDS Amazon EC2 VMware Cloud on AWS	Minimal downtime migration. Used with Oracle Data Pump for large database migrations.
Oracle Data Guard	Any size	Amazon RDS Custom Amazon EC2 VMware Cloud on AWS	Minimal downtime migration. Used with Oracle RMAN to replicate changes after initial data transfer.

Oracle RMAN	Any size	Amazon RDS Custom Amazon EC2 VMware Cloud on AWS	Databases over 2 TB, or if database backup is already in Amazon Simple Storage Service (Amazon S3).
AWS Application Migration Service	Any size	Amazon EC2	Fast replication with minimal downtime during cutover. For more information, see the Application Migration Service documentation.
VMware HCX	Any size	VMware Cloud on AWS	HCX vMotion provides online or offline migration of a single virtual machine (VM) at a time with no downtime.

The following subsections provide more information about each tool.

Oracle SQL Developer

Oracle SQL Developer is a free GUI tool from Oracle for data manipulation, administration, development, and management. This Java-based tool is available for Microsoft Windows, Linux, or macOS. You can use the Database Copy feature to migrate small databases to AWS, where the total size of your data is under 200 MB. The data transfer between source and target database is done directly over the network. To use this option, you will need a reliable network connection between the source and target database. In addition, keep in mind that this method does not encrypt data during transfer.

Oracle SQL Developer supports both Amazon RDS for Oracle and Oracle databases on Amazon EC2.

Oracle SQL*Loader

Oracle SQL*Loader is a bulk data load utility available from Oracle for loading data from external files into a database. SQL*Loader is included with the full Oracle Database client binaries. You can use SQL*Loader for small to medium-size databases under 10 GB that contain a limited number of objects. Because this is a schema-based method, it involves exporting specific schemas individually from the source database and loading them into the target database. If you have multiple schemas in a database, you have to repeat the process for each schema.

Oracle SQL*Loader supports both Amazon RDS for Oracle and Oracle databases on Amazon EC2.

Oracle Export and Import

Oracle Export and Import utilities help you migrate databases that are smaller than 10 GB and don't include binary float and double data types. The import process creates the schema objects, so you don't have to run a script to create them beforehand. This makes the process well suited for databases that have a large number of small tables.

You can use this tool for both Amazon RDS for Oracle and Oracle databases on Amazon EC2.

Oracle Data Pump

Oracle Data Pump is an enhanced version of Oracle Export and Import. This utility is used to export and import data and metadata from or to Oracle databases. You can run Data Pump export/import on an entire database, selective schemas, tablespaces, or database objects. Data Pump is the recommended tool for migrating data to AWS, for large databases that range from 10 GB to 20 TB in size. It allows a high degree of parallelism, flexible data extraction options, and scalable operations, which enable high-speed movement of data and metadata from source database to target database. Oracle Data Pump also supports encryption and compression when exporting your data to data dump files.

You can use this tool for both Amazon RDS for Oracle and Oracle databases on Amazon EC2. You can also use Oracle Data Pump with AWS DMS and Oracle GoldenGate to handle the initial data transfer for large databases.

For Amazon RDS for Oracle, after the data is exported into dump files using the Oracle Data Pump export utility, the Oracle Data Pump import utility requires the data files to be available in the database server instance to import them into the database. You can't access the file system in the Amazon RDS DB instance directly, so you will need to transfer the dump files to Amazon RDS using one of these options:

- Use a database link between the two databases. This process uses Oracle Data Pump and the Oracle DBMS_FILE_TRANSFER package. It creates a database link between the source (on-premises) Oracle database and the target Amazon RDS for Oracle database. This option requires higher bandwidth connectivity between source and target databases; we recommend that you use AWS Direct Connect. This option is recommended only for small databases. For more information, see Importing data with Oracle Data Pump and a database link in the Amazon RDS documentation.
- Use an Amazon S3 bucket. Amazon RDS for Oracle supports Amazon S3 integration. This option is recommended when you have large data dump files and your database size is in terabytes. You can then copy the data dump files from on premises to your S3 bucket by using AWS Direct Connect (if your data size is from 10 GB to 5 TB) or AWS Snowball (if your data size is more than 5 TB) depending on the required migration time for your database.

After the data pump file is uploaded to Amazon S3, you can download it to the DATA_PUMP_DIR directory on the target Amazon RDS for Oracle DB instance, and then import the data into the DB instance. For more information, see Importing data with Oracle Data Pump and an Amazon S3 bucket in the Amazon RDS documentation.

With Oracle Data Pump, you can migrate larger databases in phases, on a schema-by-schema basis. You can migrate to a different version of the Oracle Database software and also migrate to platforms that have different hardware and software configurations.

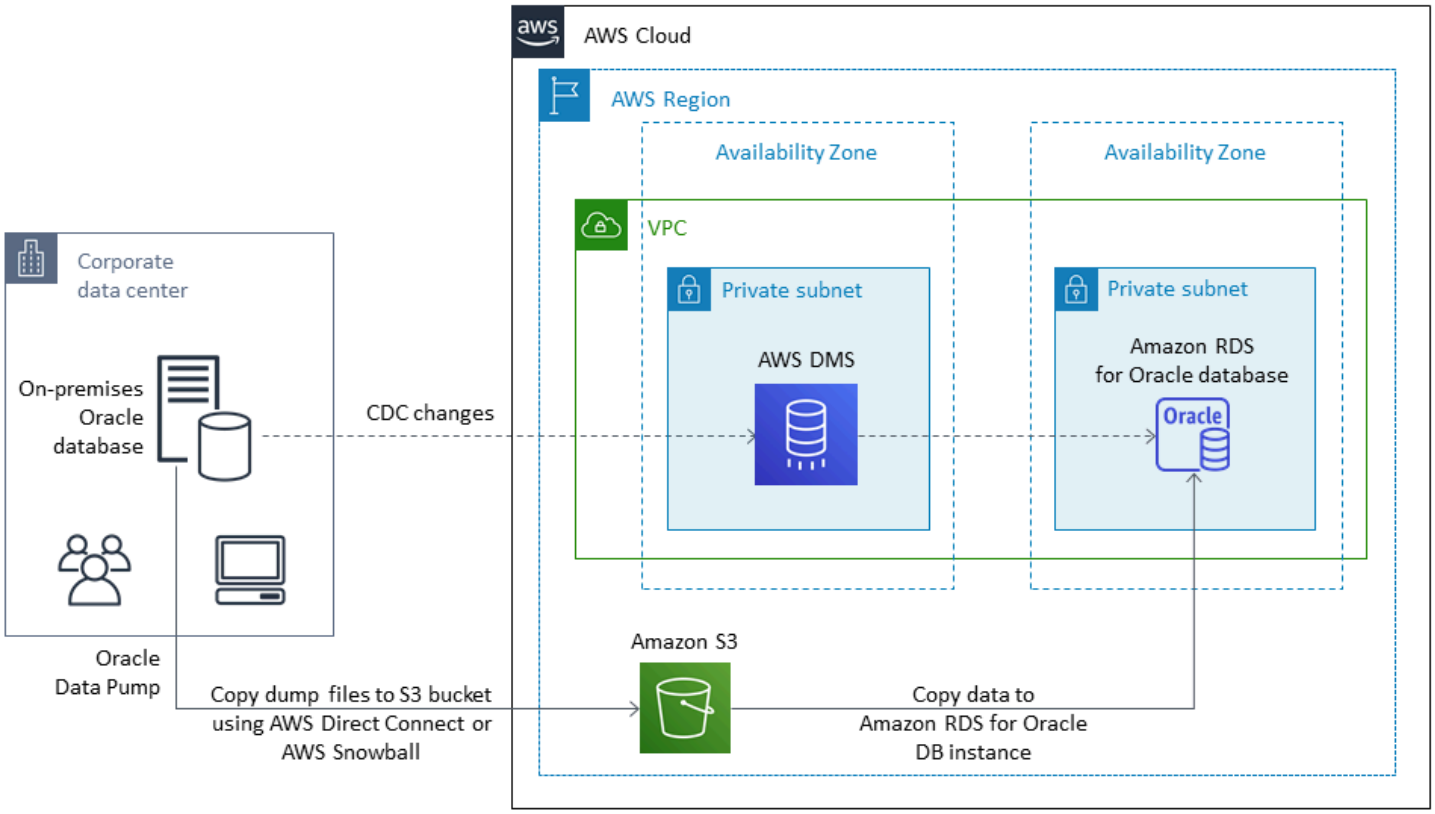
AWS DMS

AWS Database Migration Service (AWS DMS) is a managed service that helps you move data to and from AWS easily and securely. AWS DMS supports most commercial and open-source databases, and facilitates both homogeneous and heterogeneous migrations. AWS DMS offers both one-time full database copy and change data capture (CDC) technology to keep the source and target databases in sync and to minimize downtime during a migration.

AWS DMS can perform a full copy of your Oracle database schema for small (10-20 GB) to medium (100-200 GB) sized databases. For very large databases, you can migrate the data to Amazon RDS or Amazon EC2 by using Oracle Data Pump, and then use the AWS DMS CDC feature for ongoing replication with minimal downtime. When the data is synchronized, you can cut over to the target database.

The following diagram shows how you can use Oracle Data Pump and AWS DMS together to migrate an on-premises database to Amazon RDS for Oracle with minimal downtime. The Oracle

Data Pump export utility exports the schema to database dump files, and then transfers those files to Amazon S3 by using either AWS Direct Connect or AWS Snowball (depending on the size of the database, network bandwidth, and allowed migration time). After the dump files are loaded into Amazon S3, you can upload the files over to an Amazon RDS for Oracle DB instance. The Oracle Data Pump import utility then imports the data to Amazon RDS for Oracle, and AWS DMS CDC replicates all the changes from the source database to the target Amazon RDS for Oracle database.



For more information about using AWS DMS to migrate Oracle source databases, see Using an Oracle database as a source for AWS DMS in the AWS documentation.

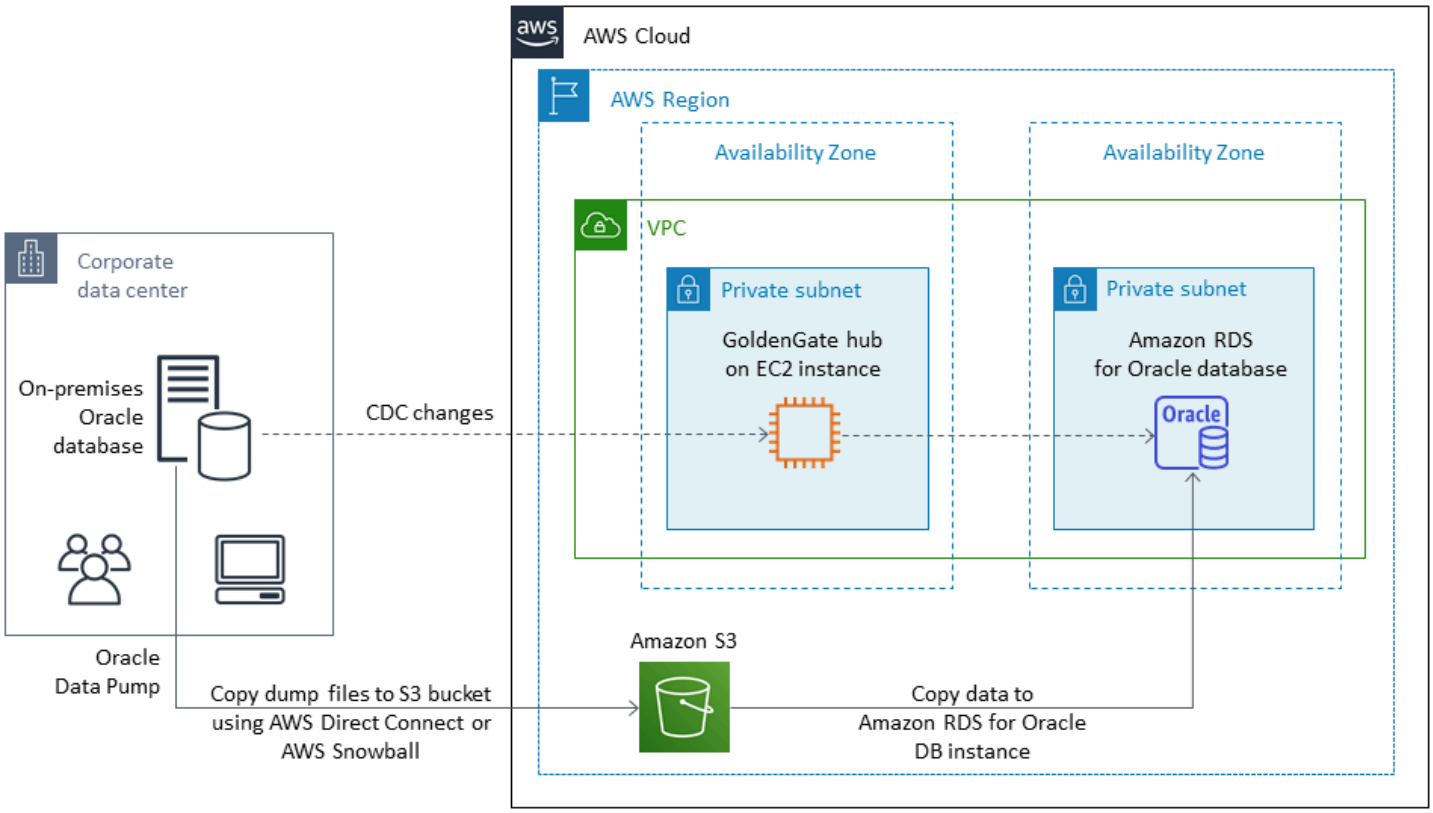
Oracle GoldenGate

Oracle GoldenGate is a tool for replicating data between a source database and one or more destination databases with minimal downtime. You can use it to build high availability architectures, and to perform real-time data integration, transactional change data capture, replication in heterogeneous environments, and continuous data replication.

You can run Oracle GoldenGate from your on-premises server in your source environment. However, we recommend that you install and run this tool from an EC2 instance, which serves as the GoldenGate hub, on AWS for better performance. You can have multiple GoldenGate hubs,

especially if you are migrating data from one source database to multiple destinations. You can use GoldenGate with Amazon RDS for Active-Active database replication, zero-downtime migration and upgrades, disaster recovery, data protection, and in-region and cross-region replication. For details, see Using Oracle GoldenGate with Amazon RDS in the AWS documentation.

The following diagram shows how to use Oracle Data Pump and Oracle GoldenGate together to migrate an on-premises Oracle database to Amazon RDS for Oracle.



Oracle GoldenGate requires a separate license from Oracle.

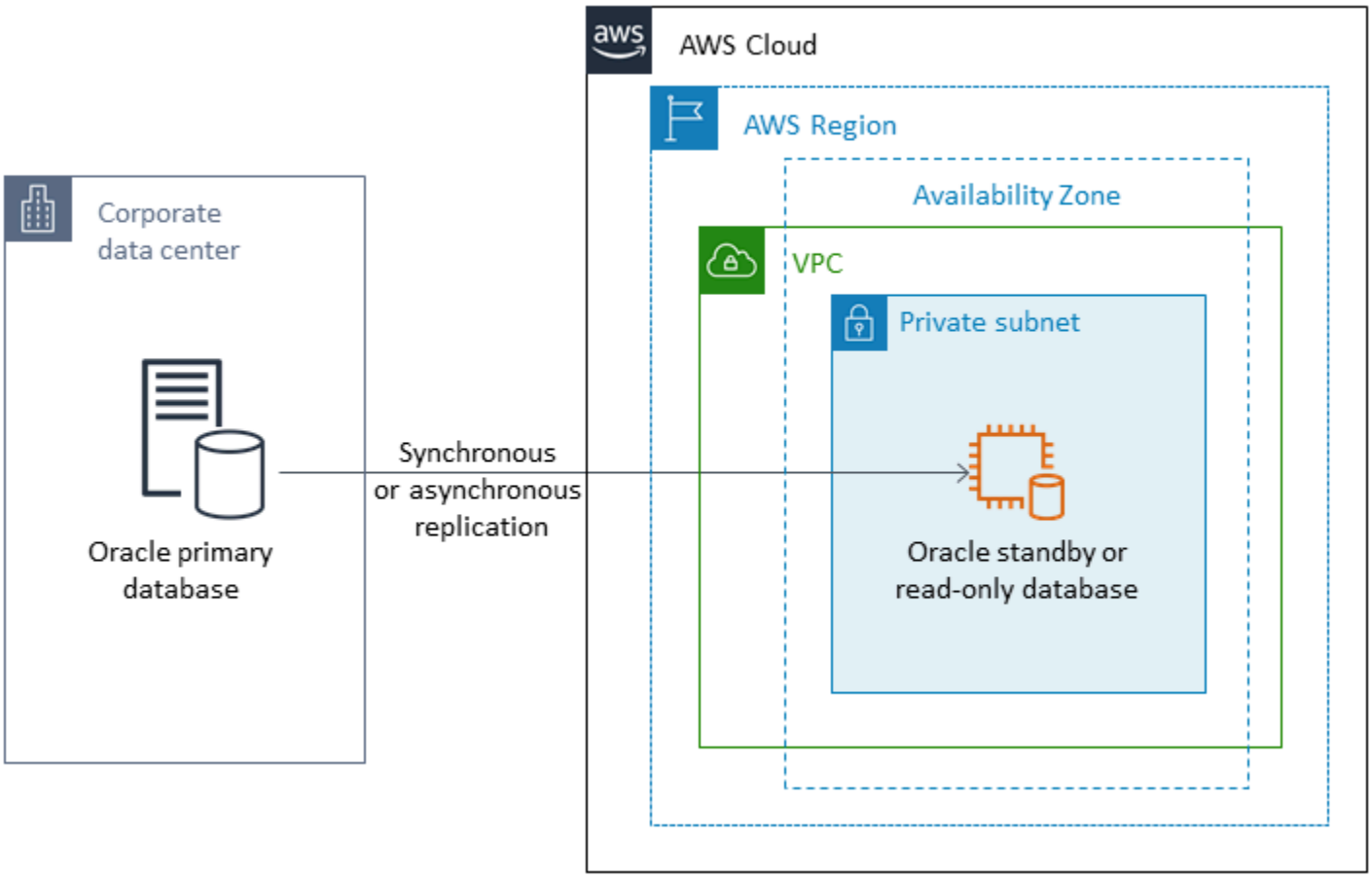
Oracle GoldenGate supports both Amazon RDS for Oracle and Oracle databases running on Amazon EC2 or VMware Cloud on AWS.

Oracle Data Guard

Oracle Data Guard provides a set of services for creating, maintaining, monitoring and managing Oracle standby databases. You can migrate your entire Oracle database from on premises to Amazon EC2 with minimal downtime by using Oracle Recovery Manager (RMAN) and Oracle Data Guard. With RMAN, you restore your primary database to the target standby database on Amazon EC2, using either backup/restore or the duplicate database method. You then configure the target

database as a physical standby database with Oracle Data Guard, allowing all the transaction/redo data changes from the primary on-premises database to the standby database.

When the primary on-premises Oracle database is in sync with the target standby database on the EC2 instance, you can switch over to the target database, which will convert it to a read-write database. You can then point your application connections to the new primary database. With this option, you can achieve minimum downtime and get an exact physical copy of your database on AWS. The migration is illustrated in the following diagram.



Oracle Data Guard supports Oracle databases running on Amazon EC2, Amazon RDS Custom, and VMware Cloud on AWS.

Oracle RMAN

Oracle Recovery Manager (RMAN) is a tool provided by Oracle for performing and managing Oracle database backups and restorations. You can use RMAN to back up your Oracle database from on premises or from your data center, and restore it to an Oracle database on an EC2 instance. Use this method if you are planning to move your entire database to a self-managed Oracle database on

an EC2 instance. The database can be of any size, and you can use parallelism, compression, and encryption in your backups.

You can place the Oracle RMAN backup of your on-premises Oracle database directly in an S3 bucket by using the Oracle Secure Backup (OSB) Cloud module, AWS Storage Gateway, or AWS DataSync. You can then use an AWS Identity and Access Management (IAM) role to give the S3 bucket access to your target Oracle database on an EC2 instance, and restore the database by using the RMAN backup files. You can take incremental backups from your on-premises Oracle database and apply them to the target Oracle database on the EC2 instance until the on-premises and target databases are in sync. You can then perform the switchover at a convenient time.

Oracle RMAN supports Amazon EC2, Amazon RDS Custom, and VMware Cloud on AWS migrations. It's the recommended approach when you can allow enough downtime for migrating your data to AWS.

VMware HCX

VMware Hybrid Cloud Extension (HCX) enables you to migrate your on-premises Oracle databases to AWS without having to retrofit your VMware infrastructure. It includes several migration methods that are detailed in the blog posts How to Migrate Oracle Workloads to VMware Cloud on AWS and Migrating Workloads to VMware Cloud on AWS with Hybrid Cloud Extension (HCX). One of these methods, HCX vMotion, provides a live migration of a single VM with no downtime and high availability.

HCX is available free of charge to VMware Cloud on AWS customers.

Licensing options

Oracle Database licensing on AWS is based on the size of the instance on which the database is installed. Many Oracle Database workloads need high memory, storage, and I/O bandwidth, but are not CPU-bound, so you can reduce the number of virtual CPUs (vCPUs) in your deployment without affecting performance.

AWS offers the following CPU options for optimizing your Amazon RDS and EC2 instances for specific workload or business needs:

- Number of CPU cores: You can customize the number of CPU cores for the instance.
- Threads per core: You can disable multithreading by specifying a single thread per CPU core.

For more information, see Optimizing CPU options in the Amazon EC2 documentation and Introducing Optimize CPUs for Amazon RDS for Oracle on the AWS website.

You can run Oracle Database on AWS under two different licensing models:

- License Included
- Bring Your Own License (BYOL)

License Included

In the License Included model, the Oracle Database software license is made available by AWS, so you don't have to purchase your own Oracle license separately. The License Included model pricing includes software, underlying hardware resources, and Amazon RDS management capabilities for Amazon RDS for Oracle. You pay for compute capacity by the hour your DB instance runs, with no long-term commitments. This frees you from the costs and complexities of planning, purchasing, and maintaining hardware.

For both Single-AZ and Multi-AZ deployments, pricing is per DB instance-hour consumed, from the time you launch a DB instance until you stop or delete the instance.

The License Included model supports Standard Edition Two (SE2). For pricing information, see Amazon RDS for Oracle pricing on the AWS website.

BYOL

The Bring Your Own License (BYOL) model is intended for customers who prefer to use their existing Oracle Database licenses or purchase new Oracle licenses. If you already own an Oracle database license, you can use the BYOL model to run your Oracle database on Amazon RDS. If you're migrating your Oracle database to Amazon EC2 or to VMware Cloud on AWS, you must use your own Oracle license.

> **Notice**
>
> As of April 30, 2024, VMware Cloud on AWS is no longer resold by AWS or its channel partners. The service will continue to be available through Broadcom. We encourage you to reach out to your AWS representative for details.

To run a DB instance under the BYOL model, you must have the appropriate Oracle Database license for the DB instance class and Oracle Database edition you want to run. You must also follow Oracle's policies for licensing Oracle Database software in the cloud computing environment.

If you use the BYOL model, you must have a license for both the primary DB instance and the standby DB instance in a Multi-AZ deployment. Amazon RDS supports Multi-AZ deployments for Oracle as a high-availability, failover solution. We recommend Multi-AZ for production workloads. For more information, see Configuring and managing a Multi-AZ deployment in the Amazon RDS documentation.

The BYOL model supports Oracle Database Enterprise Edition (EE) and Standard Edition Two (SE2).

For more information about licensing options for Amazon RDS for Oracle, see Oracle Licensing and the Amazon RDS for Oracle FAQs on the AWS website.

Heterogeneous database migration

Because of the innovations and improvements in open-source databases and cloud computing platforms like AWS, many organizations are moving from proprietary (online transaction processing or OLTP) database engines such as Oracle to open-source engines. Oracle databases are mission-critical systems for any organization, but being locked into a particular vendor is a risky and costly situation. Low operating cost and no licensing fees are compelling reasons to consider switching the underlying database technology to open-source or AWS Cloud-native databases.

Other reasons for migrating off Oracle are vendor lock-in periods, licensing audits, expensive licensing, and cost. Oracle's list pricing is based on a per-core model with additional costs for features such as partitioning and high availability. For this reason, many organizations choose to migrate their Oracle databases to either open-source databases (such as PostgreSQL, MySQL, or MariaDB) or AWS Cloud-native databases (such as Amazon Aurora or Amazon DynamoDB) when they migrate to AWS.

You can also migrate your Oracle data warehouse database to Amazon Redshift, which is a fast, fully managed cloud data warehouse. Amazon Redshift is integrated with your data lake, offers up to three times faster performance than any other data warehouse, and costs up to 75 percent less than any other cloud data warehouse. For more information, see Migrate from Oracle to Amazon Redshift on the AWS website.

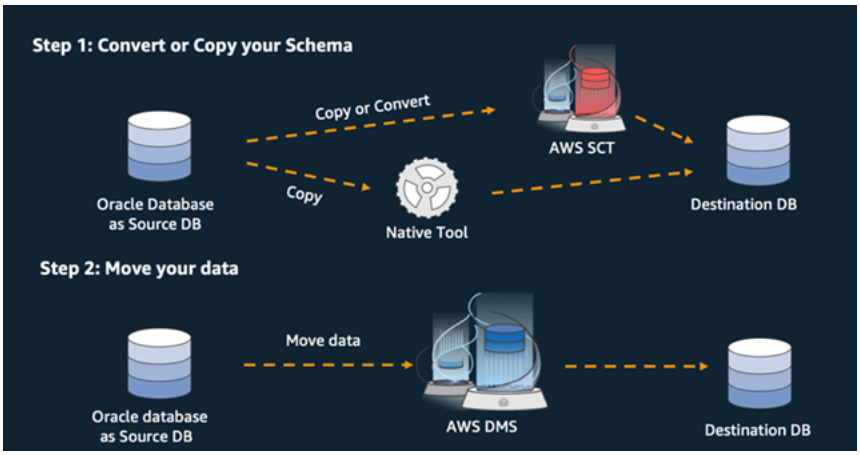
To migrate to an open-source or AWS-native database, choose the right database depending on the type of data you have, the access model, scalability, application practicalities, and complexity. For example, PostgreSQL databases have become very popular in recent years for their powerful functionality and high degree of compatibility with commercial databases, and they're the most common migration target for users who are refactoring their Oracle databases. But migrating from Oracle to PostgreSQL and to other open-source databases has often been difficult and time-consuming, and requires careful assessment, planning, and testing.

This process becomes easier with services like AWS DMS and AWS Schema Conversion Tool (AWS SCT), which help you migrate your commercial database to an open-source database on AWS with minimal downtime.

In heterogeneous database migrations, the source and target databases engines are different, as in Oracle to Amazon Aurora, or Oracle to PostgreSQL, MySQL, or MariaDB migrations. The schema structure, data types, and database code in the source and target databases can be quite different,

so the schema and code must be transformed before the data migration starts. For this reason, heterogeneous migration is a two-step process:

- Step 1. Convert the source schema and code to match that of the target database. You can use AWS SCT for this conversion.
- Step 2. Migrate data from the source database to the target database. You can use AWS DMS for this process.



AWS DMS handles all required data type conversions automatically during migration. The source database can be located in your own premises outside AWS, it can be a database that's running on an EC2 instance, or it can be an Amazon RDS database (see Sources for data migration in the AWS DMS documentation). The target can be a database in Amazon EC2, Amazon RDS, or Amazon Aurora.

For more information about refactoring your Oracle database on AWS, see the re-architect patterns on the AWS Prescriptive Guidance website.

Tools for heterogeneous database migrations

The following chart provides a list of tools that you can use to migrate from Oracle Database to another database engine.

Migration tool	Target database support	Used for
AWS SCT	Amazon RDS for MySQL	Schema conversion
	Amazon RDS for PostgreSQL	

	Amazon Aurora MySQL	
	Amazon Aurora PostgreSQL	
AWS DMS	Amazon RDS for MySQL	Data migration
	Amazon RDS for PostgreSQL	
	Amazon Aurora MySQL	
	Amazon Aurora PostgreSQL	

The following subsections provide more information about each tool.

AWS SCT

AWS Schema Conversion Tool (AWS SCT) converts your existing commercial database schemas to an open-source engine or to an AWS Cloud-native database. AWS SCT makes heterogeneous database migrations predictable by automatically converting 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 for manual conversion. AWS SCT can also scan your application source code for embedded SQL statements and convert them as part of a database schema conversion project.

AWS DMS

AWS Database Migration Service (AWS DMS) migrates your data rapidly and securely to AWS. During migration, the source database remains fully operational, minimizing application downtime. AWS DMS supports homogeneous migrations such as Oracle to Oracle as well as heterogeneous migrations between different database platforms, such as Oracle to an open-source database or to an AWS cloud-native database. AWS DMS manages the complexities of the migration process, including automatically replicating data changes that occur in the source database to the target database. After the database migration is complete, the target database remains synchronized with the source database for as long as you choose, and you can switch over to the target database at a convenient time.

Best practices for migrating to Amazon RDS for Oracle

Based on the assessment of your database and your project requirements, if your goal is to migrate to Amazon RDS for Oracle, follow the best practices in this section to provision your target database, perform the migration, and test, operate, and optimize your Amazon RDS for Oracle database.

> **Important**
>
> Make sure that you have a rollback plan before you migrate your database.

Provisioning your target database

After you finish assessing, planning, and preparing your database migration strategy, follow these best practices when provisioning your Amazon RDS for Oracle database:

- Right-size the Amazon RDS for Oracle DB instance based on your requirements for CPU, memory, IOPS, and storage type.
- Set the correct time zone and character set.
- Make sure to launch Amazon RDS in the correct virtual private cloud (VPC).
- Create the security groups with correct port and IP addresses.
- Provision your Amazon RDS database in a private subnet for security.
- If possible, provision the DB instance by using the latest Oracle Database version, which is currently 19c. Earlier versions are nearing end of support. For more information, see Amazon RDS support for Oracle Database 19c.
- If you want to use encryption, always enable it while you are provisioning the Amazon RDS database.
- Create a separate option group and parameter group for each Amazon RDS database.

Exporting data from your source database

There are many tools for migrating an Oracle database to an Amazon RDS for Oracle database. The most commonly used tool is Oracle Data Pump. Before you export your source Oracle database, check the following to facilitate the export process:

- Check the database size, to see if you can migrate it schema by schema, instead of migrating the full database. Migrating schemas individually is less error prone and more manageable than migrating them all at once.
- Export data in parallel mode, by using the Oracle Data Pump PARALLEL parameter, for better performance.
- Check if the tables have large objects (LOBs). If you have large tables with LOBs, we recommend that you export those tables separately.
- During the export process, avoid running long database transactions on your source database to avoid Oracle read inconsistency errors.
- If you are using replication tools such as AWS DMS, Oracle GoldenGate, or Quest SharePlex, make sure that you have enough space on your on-premises server to hold archive logs for 24-72 hours, depending on how long the migration takes.

Transferring data dump files to AWS

If you're using AWS Direct Connect, which provides high bandwidth connectivity between your on-premises environment and AWS, you can copy the Data Pump files by using either the Oracle DBMS_FILE_TRANSFER utility or the Amazon S3 integration feature. If you do not have high bandwidth through AWS Direct Connect, use AWS Snowball to transfer large database export dump files.

Importing data to your target database

- If you're migrating a very large database, we recommend that you provision a bigger Amazon RDS instance type initially, for the duration of the migration, for faster data loads. After the migration is complete, you can change the DB instance to the right-sized instance type.
- Increase the size of redo log files, undo tablespaces, and temporary tablespaces to improve performance during migration, if needed.
- Disable the Multi-AZ option during the import process, and enable it after migration is complete.
- Disable the generation of archive logs by setting the backup retention to zero to achieve faster data load.
- Prepare the target database by creating tablespaces, users, roles, profiles, and schemas in advance.
- If you have large tables with LOBs, import each LOB table separately.

Post-import steps

- Check the import log files for errors, and fix any errors after the import is complete.
- Check for invalid objects. If you find any, compile and fix them.
- Some procedures might not compile due to lack of permissions on SYS objects that are not allowed or supported in Amazon RDS. These procedures have to be rewritten.
- If you are using sequences, validate the sequence values against the source database to avoid sequence inconsistency.
- Make sure that the object count in your Amazon RDS database is the same as in the source database. Validate tables, indexes, procedures, triggers, functions, packages, constraints, and other objects.
- If your source database has database links to other databases, test the connectivity to confirm that the links still work.
- Gather dictionary-level and schema-level statistics for optimal performance.

Testing the migration

We recommend the following tests to validate your application against your new Amazon RDS for Oracle database:

- You might have to upgrade your Oracle client software or JDBC software based on the Amazon RDS for Oracle database version. If you've migrated to a newer version of Oracle Database, it might not support older versions of Oracle client software.
- Perform functional testing.
- Compare the performance of SQL queries in your source and target databases, and tune the queries as needed. Some queries might perform more slowly in the target database, so we recommend that you capture the baselines of the SQL queries in the source database.
- When the application team finishes testing and confirms that your Amazon RDS database is functioning properly, you can:
  - Right-size the Amazon RDS DB instance based on your assessment.
  - Enable backup retentions.
  - Enable archive logs.
  - Reset the size of redo log files.

- Enable the Multi-AZ option.
- Create Amazon CloudWatch alarms and set up Amazon Simple Notification Service (Amazon SNS) topics for alerts.

For additional validation during the proof-of-concept (POC) phase, we recommend the following supplemental tests:

- Run performance tests to ensure that they meet your business expectations.
- Test database failover, recovery, and restoration to make sure that you're meeting RPO and RTO requirements.
- List all critical jobs and reports, and run them on Amazon RDS to evaluate their performance against your service-level agreements (SLAs).

Operating and optimizing your Amazon RDS database

When your database is on AWS, make sure that you are following best practices in areas such as monitoring, alerting, backups, and high availability in the cloud. For example:

- Set up CloudWatch monitoring, and enable detailed monitoring.
- Use Amazon RDS Performance Insights and the Oracle Enterprise Manager (OEM) Management Agent to monitor your database.
- Set up alerts by using SNS topics.
- Set up automatic backups using AWS Backup. You can also use Oracle Data Pump backups or take manual snapshots.
- For high availability, set up the Amazon RDS Multi-AZ feature.
- If you need read-only databases, set up a Read Replica within the same or across AWS Regions according to your needs.

Oracle database migration patterns

Use the following links to see the AWS Prescriptive Guidance patterns for migrating an Oracle database to AWS:

- Rehost patterns (from Oracle to Amazon EC2)
- Replatform patterns (from Oracle to Amazon RDS for Oracle)
- Re-architect patterns (from Oracle to open-source and AWS Cloud-native databases)

If you're looking for patterns that cover the use of a specific tool, type in the tool name in the search box or choose it from a filter. For example, you can query for all Oracle migration patterns that use AWS DMS.

AWS Partners

Database migration can be a challenging project that requires expertise and tools. You can accelerate your migration and time to results through partnership. AWS Database Migration Service Delivery Partners have the required expertise to help customers migrate to the cloud easily and securely. These partners have the expertise for both homogeneous migrations, such as Oracle to Oracle, and heterogeneous migrations between different database platforms, such as Oracle to Amazon Aurora or Microsoft SQL Server to MySQL.

Based on your requirements and preferences, you can use the AWS Partner to handle the complete migration or to help with only some aspects of the migration. In addition, you can use tools and solutions provided by AWS Partners to help with the migration.

Additional resources

Blog posts

- Database Migration—What Do You Need to Know Before You Start?
- Migrating Oracle databases with near-zero downtime using AWS DMS
- How to Migrate Your Oracle Database to PostgreSQL
- How to Migrate Your Oracle Database to Amazon Aurora
- How to Migrate Oracle Workloads to VMware Cloud on AWS
- Best Practices for Virtualizing Oracle RAC with VMware Cloud on AWS

AWS documentation

- Amazon Aurora
- Amazon EC2
- Amazon RDS
- Amazon RDS Custom
- Amazon Redshift
- AWS DMS
- AWS SCT
- Using Oracle GoldenGate with Amazon RDS
- Oracle licensing

Additional information

- Oracle Data Pump
- Oracle Data Guard
- Oracle Export and Import
- Oracle GoldenGate
- Oracle RMAN
- Oracle SQL Developer
- Oracle SQL*Loader

- Licensing Oracle Software in the Cloud Computing Environment
- VMware HCX

Appendix: Oracle migration questionnaire

Use the questionnaire in this section as a starting point to gather information for the assessment and planning phases of your migration project. You can download this questionnaire in Microsoft Excel format and use it to record your information.

General information

1. What is the name of your Oracle database?
2. What is the version of your Oracle database?
3. What is the edition of the database: Standard or Enterprise?
4. What is the size of your database?
5. What is the database character set?
6. What is the time zone of the database?
7. What are the average and maximum I/O transactions per second (TPS)?
8. What is the IOPS (on average and maximum) for this database for read/write operations?
9. What is the redo log generation per hour (on average and maximum) per day?
10. How many schemas do you plan to migrate?
11. What is the size of each schema?
12. How many big tables (over 100 GB) do you have per schema?
13. Can you archive the tables that don't need to migrate?
14. What is the size of system global areas (SGAs) and program global areas (PGAs) or Automatic Memory Management (AMM) usage, in megabytes?
15. How many tables have LOBs? What is the maximum size of the LOBs?
16. Do all your tables with LOBs have primary keys?
17. Do you have database links that point to other databases?
18. What are the SLA requirements for your database?
19. What are the RTO and RPO requirements for your database?
20. How much database downtime can you allow for migration purposes?
21. Do you have any compliance, regulatory, or auditing requirements?

Infrastructure

1. What is the hostname of the database?
2. What is the operating system used for this database?
3. How many CPU cores does the server have?
4. What is the memory size on the server?
5. Are you using local storage?
6. Do you use network-attached storage (NAS) or storage area network (SAN) storage types?
7. Do you have a RAC database? If yes, how many nodes does it have?
8. Do you use partitioning features?
9. Do you use Oracle Spatial?
10. Do you have a multi-tenant database?

Database backups

1. How do you back up your database? How often?
2. What is your retention period for archive logs and backups?
3. Do you use backups to clone your database?
4. Where do you store your backup?

Database security

1. Do you use Oracle Database Vault?
2. Do you use data masking?
3. Do you use Secure Sockets Layer (SSL)?
4. Do you use Oracle Advanced Security features such as Transparent Data Encryption (TDE)?
5. Do you use Oracle Advanced Compression?

Database high availability and disaster recovery

1. What are your high availability requirements?

2. Do you use Oracle Data Guard? Where are your primary and standby database regions?
3. Do you use Oracle Active Data Guard?
4. Do you use a Domain Name System (DNS) alias for database connectivity?
5. Do you use replication tools such as Oracle GoldenGate, Quest SharePlex, or Oracle Streams?

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
Removed section	Removed information about AWS Workload Qualification Framework (AWS WQF).	July 20, 2023
Removed section	Removed information about CloudEndure Migration, which is being discontinued. AWS Application Migration Service is the primary migration service recommended for lift-and-shift migrations to the AWS Cloud.	September 23, 2022
Added section	Added information about migrating Oracle databases to Amazon RDS Custom.	June 30, 2022
Updated section	Updated the *CloudEndure Migration section* with the latest information about product availability.	May 10, 2022
Updated AWS WQF information	Updated the AWS WQF section with the latest support and availability information.	October 16, 2020
Added sections	Updated Oracle database migration strategies with additional information, added best practices for migrating	March 16, 2020

	to Amazon RDS, and added a questionnaire for migration assessment and planning.	
Initial publication	—	February 24, 2020

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 AWS CodeCommit. 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, AWS Panorama offers devices that add CV to on-premises camera networks, and Amazon SageMaker 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.

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.

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.

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.

G

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.

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.

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 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.

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

An approach in system engineering that takes privacy into account throughout the whole engineering 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.

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).

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.

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 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.

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.

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.