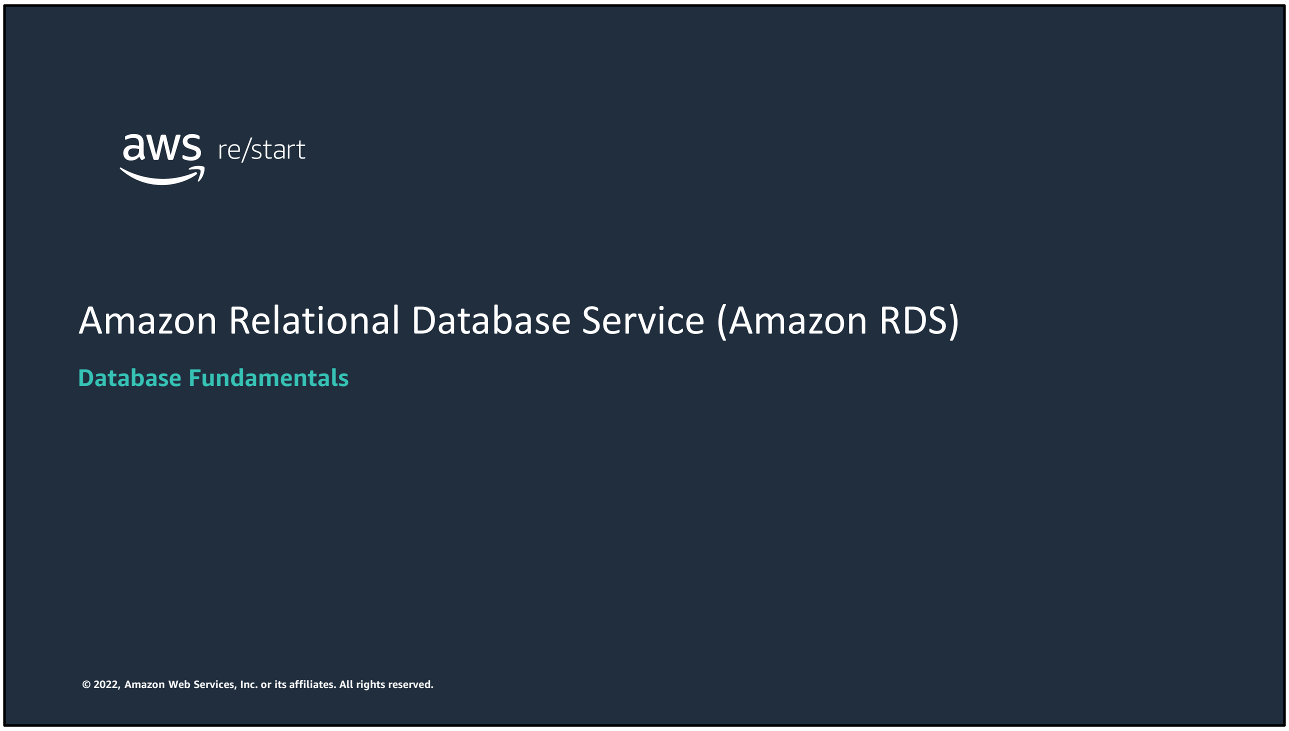


Amazon Relational Database Service (Amazon RDS)

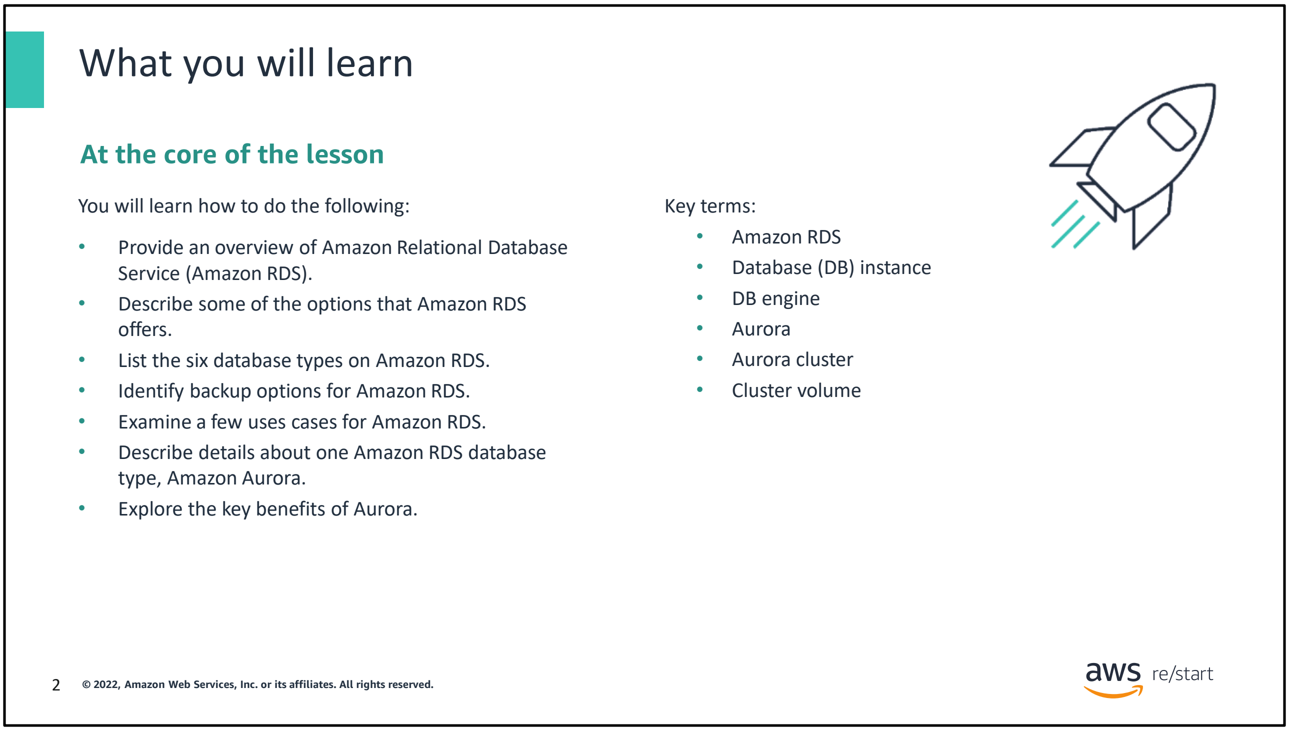
Database Fundamentals

Welcome to Amazon Relational Database Service (Amazon RDS).

In this module, you'll discover key concepts that are related to database solutions, including the following:

- Identifying the different database services in the cloud
- Discovering the differences between unmanaged and managed database solutions

The goal of this module is to help you understand the database resources that are available to power your solution. You will also review the different service features that are available so that you can understand how different choices affect solution availability.



What you will learn

At the core of the lesson

You will learn how to do the following:

- Provide an overview of Amazon Relational Database Service (Amazon RDS).
- Describe some of the options that Amazon RDS offers.
- List the six database types on Amazon RDS.
- Identify backup options for Amazon RDS.
- Examine a few uses cases for Amazon RDS.
- Describe details about one Amazon RDS database type, Amazon Aurora.
- Explore the key benefits of Aurora.

Key terms:

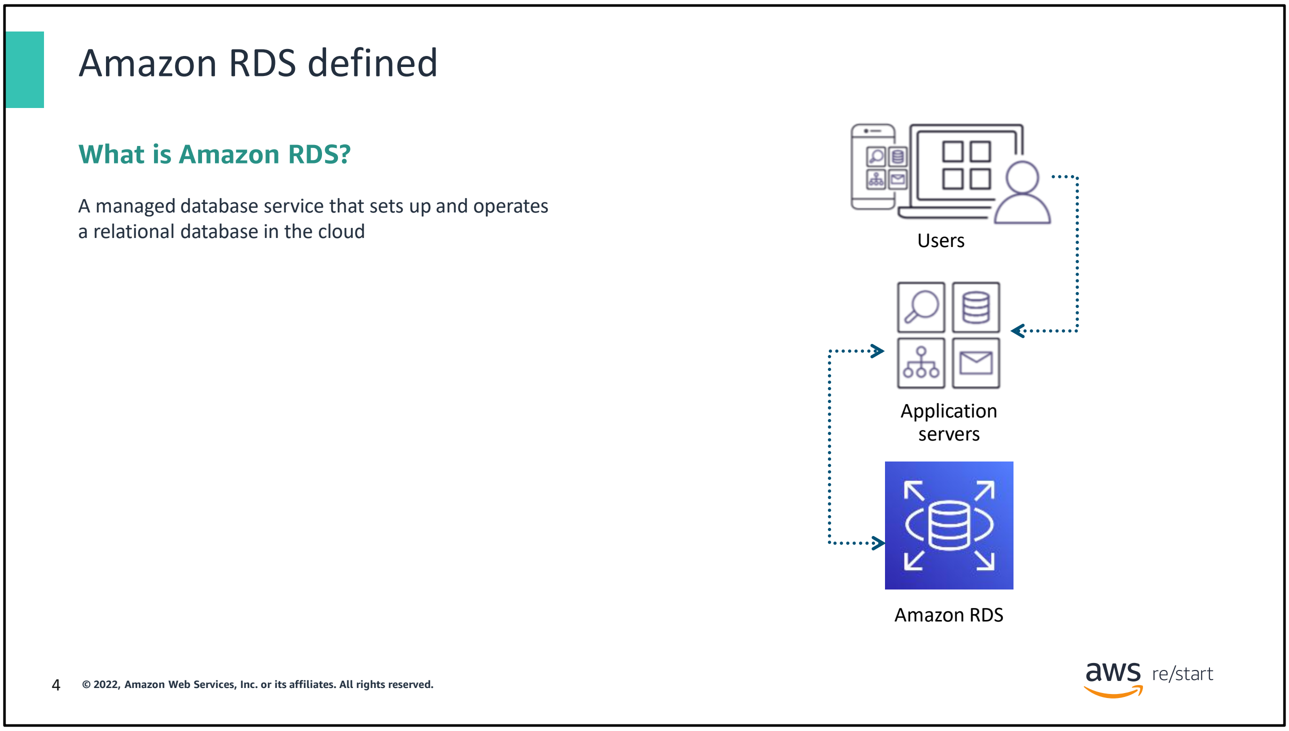
- Amazon RDS
- Database (DB) instance
- DB engine
- Aurora
- Aurora cluster
- Cluster volume

In this module, you will learn how to do the following:

- Provide an overview of Amazon RDS.
- Describe some of the options that Amazon RDS offers.
- List the six database types on Amazon RDS.
- Identify backup options for Amazon RDS.
- Examine a few uses cases for Amazon RDS.
- Describe details about one Amazon RDS database type, Amazon Aurora.
- Explore the key benefits of Aurora.

Amazon RDS

To begin with, you will discuss Amazon RDS.



Amazon RDS defined

What is Amazon RDS?

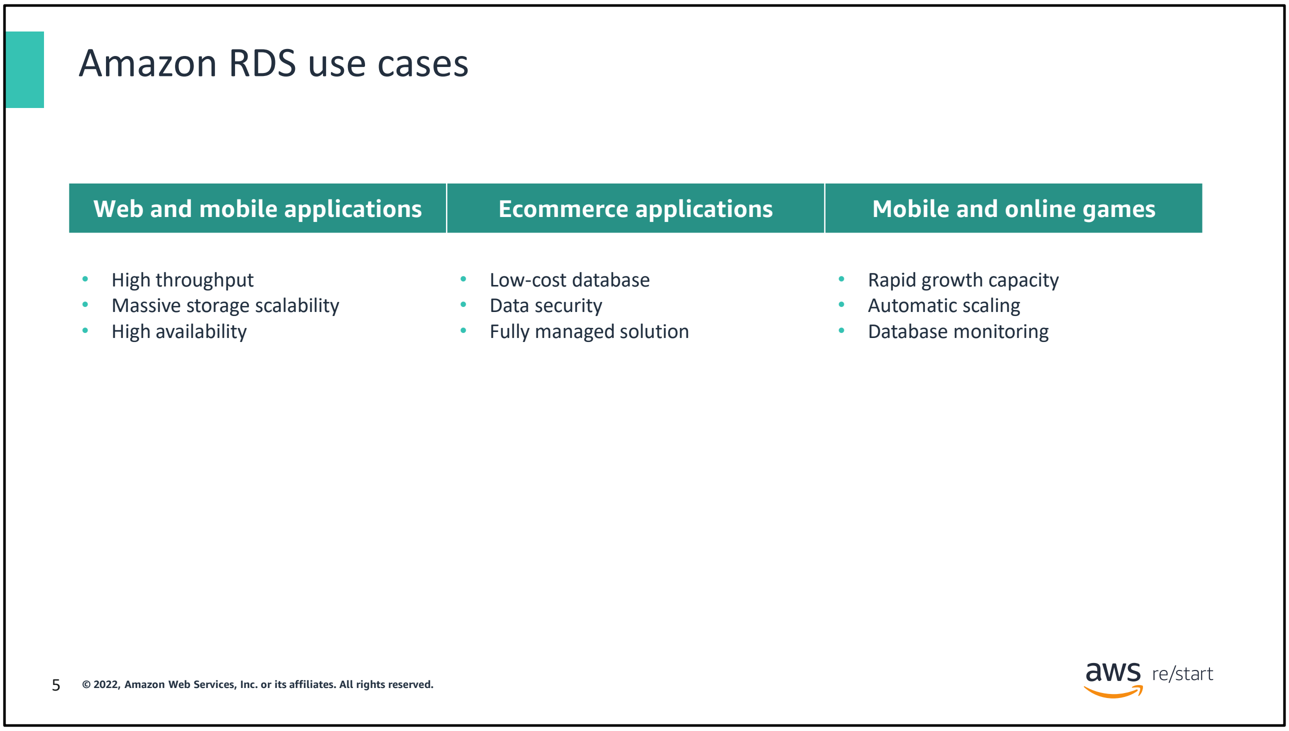
A managed database service that sets up and operates a relational database in the cloud

Users
Application servers
Amazon RDS

Amazon RDS is a managed database service that sets up and operates a relational database in the cloud.

Running an unmanaged, standalone relational database can be time-consuming and have limited scope. To address these challenges, AWS provides a service that sets up, operates, and scales the relational database without any ongoing administration. Amazon RDS provides cost-efficient and resizable capacity while automating time-consuming administrative tasks.

Amazon RDS frees you to focus on your applications so that you can give them the performance, high availability, security, and compatibility that they need. With Amazon RDS, your primary focus is your data and optimizing your application.



Amazon RDS use cases

Web and mobile applications

- High throughput
- Massive storage scalability
- High availability

Ecommerce applications

- Low-cost database
- Data security
- Fully managed solution

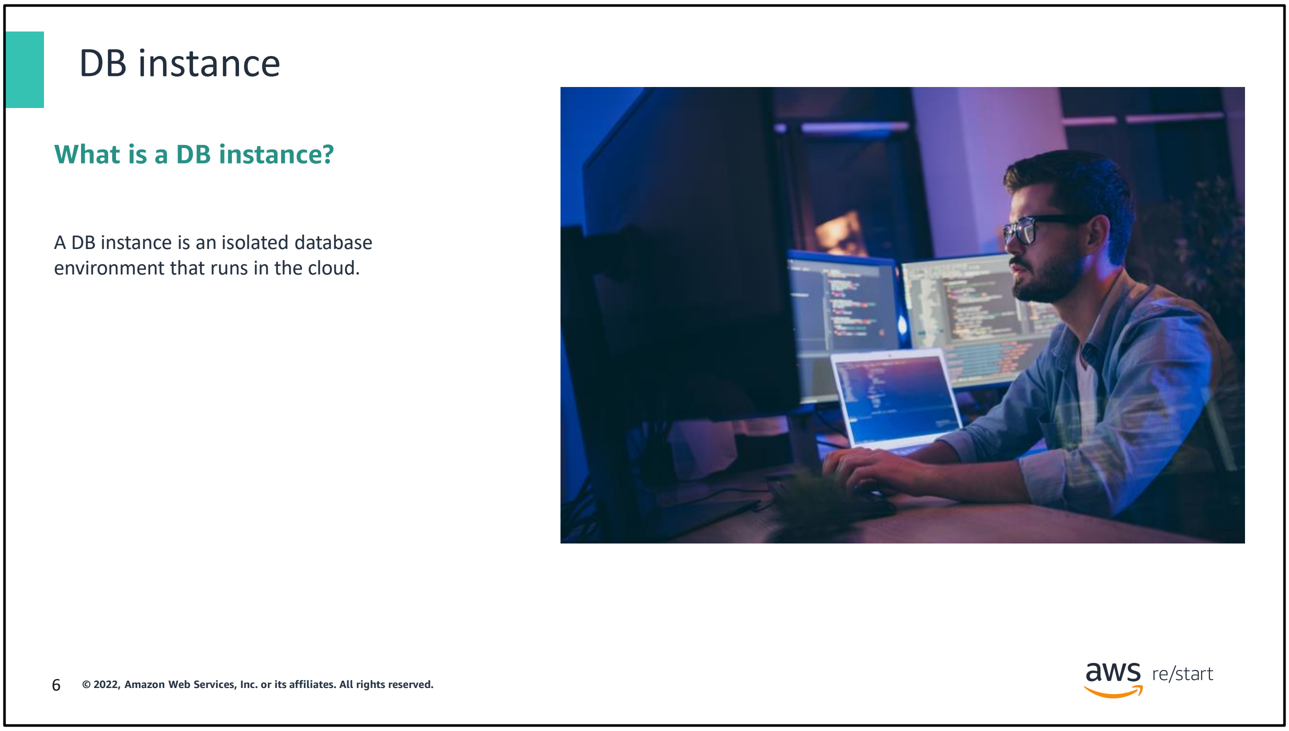
Mobile and online games

- Rapid growth capacity
- Automatic scaling
- Database monitoring

Amazon RDS works well for web and mobile applications that need a database with high throughput, extensive storage scalability, and high availability. Because Amazon RDS does not have any licensing constraints, it fits the variable usage pattern of these applications.

For small and large ecommerce businesses, Amazon RDS provides a flexible, secured, and low-cost database solution for online sales and retailing.

Mobile and online games require a database platform with high throughput and availability. Amazon RDS manages the database infrastructure so that game developers do not need to worry about provisioning, scaling, or monitoring database servers.



DB instance

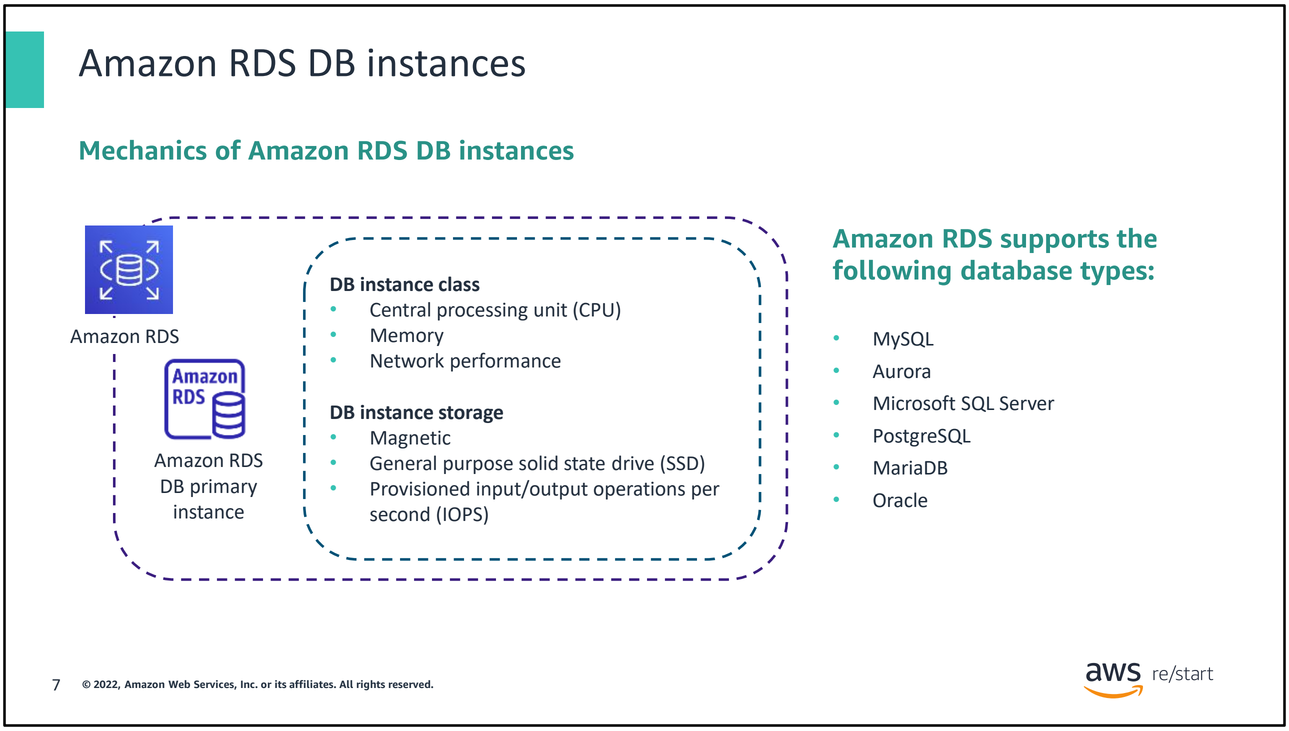
What is a DB instance?

A DB instance is an isolated database environment that runs in the cloud.

What is a DB instance ?

A DB instance is an isolated database environment that runs in the cloud.

It is the basic building block of Amazon RDS.



Amazon RDS DB instances

Mechanics of Amazon RDS DB instances

Amazon RDS

Amazon RDS DB primary instance

DB instance class

- Central processing unit (CPU)
- Memory
- Network performance

DB instance storage

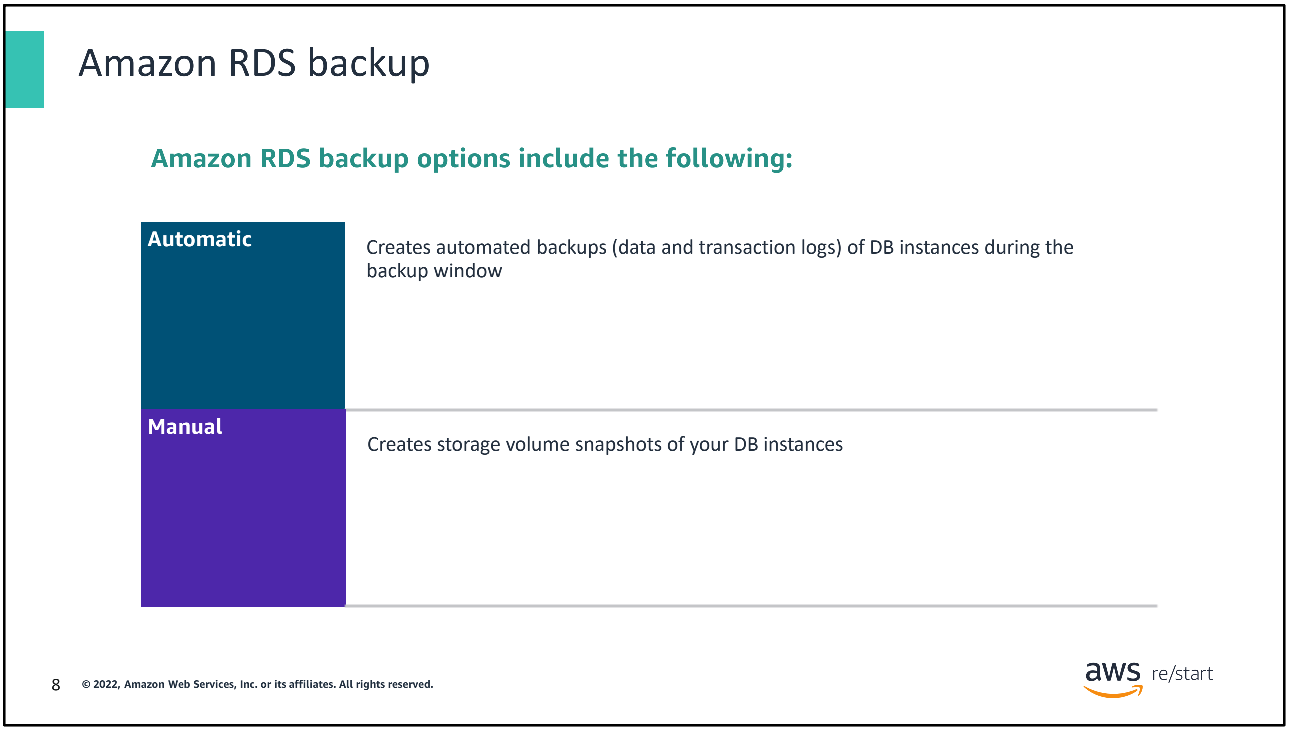
- Magnetic
- General purpose solid state drive (SSD)
- Provisioned input/output operations per second (IOPS)

Amazon RDS supports the following database types:

- MySQL
- Aurora
- Microsoft SQL Server
- PostgreSQL
- MariaDB
- Oracle

The basic building block of Amazon RDS is the DB instance. A DB instance is an isolated database environment. It can contain multiple user-created databases, and you can access it by using the same tools and applications that you use with a standalone database instance. The resources in a database instance are determined from its database instance class, and the type of storage is determined from the type of disks.

Database instances and storage differ in performance characteristics and price so that you can customize your performance and costs to the needs of your database. When you choose to create a DB instance, you must first specify which database engine to run. Amazon RDS currently supports the following databases: MySQL, Aurora, Microsoft SQL Server, PostgreSQL, MariaDB, and Oracle.



Amazon RDS backup

Amazon RDS backup options include the following:

Automatic	Creates automated backups (data and transaction logs) of DB instances during the backup window
Manual	Creates storage volume snapshots of your DB instances

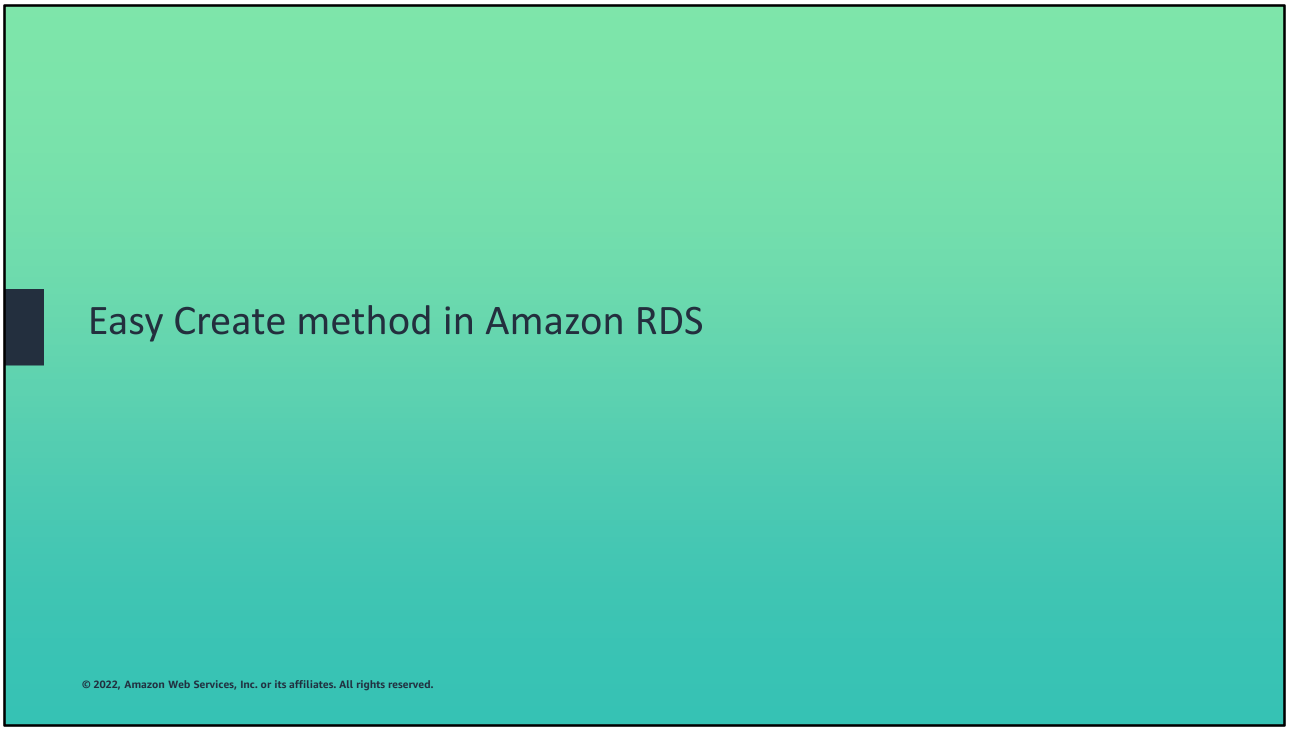
aws re/start

Because Amazon RDS is a fully managed service, one task that it automatically performs is the periodic backup of a DB instance. The entire instance is backed up to a storage volume snapshot during a specified backup window. It is retained according to a specified backup retention period. The first snapshot of a DB instance contains the full data. Subsequent snapshots are incremental and contain only the data that changed since the most recent snapshot.

Optionally, you can back up a database instance manually by creating a snapshot.

For more information about working with backups and managing Amazon RDS backups and snapshots, see “Backing up and restoring an Amazon RDS DB instance” at https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/CHAP_CommonTasks.BackupRestore.html.

For more information about Amazon RDS, see Amazon RDS FAQs at https://aws.amazon.com/rds/faqs/.

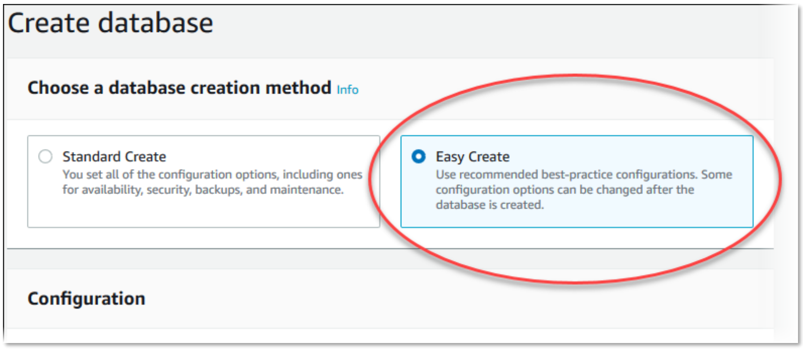


Easy Create method in Amazon RDS

This section covers a sample configuration that uses the Easy Create method to create a database in Amazon RDS. This section also examines some of the differences between the Standard method and Easy Create method.

Creating a DB instance in Amazon RDS

- The easiest way to create a DB instance is to use the AWS Management Console.
- After you create the DB instance, you can use standard MySQL utilities—such as MySQL Workbench—to connect to a database on the DB instance.



Create database

Choose a database creation method Info

○ Standard Create
You set all of the configuration options, including ones for availability, security, backups, and maintenance.

● Easy Create
Use recommended best-practice configurations. Some configuration options can be changed after the database is created.

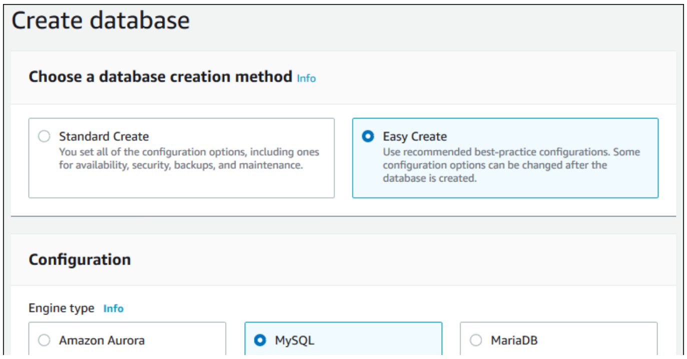
Configuration

You can create a DB instance in Amazon RDS in two ways. The management console offers an Easy Create method and a Standard method. If you choose to use the Easy Create method, the configuration options are based on Amazon best practices. You can specify a DB engine and instance size and can choose an identifier. You also have an opportunity to select Standard to create your instance. You determine or choose your configurations.

How to create a DB instance

MySQL example

1. In the Configuration section, choose **MySQL**.
2. Complete the remaining steps that are needed to answer the required configuration questions, and then choose **Create**.
3. Depending on the DB instance class and the amount of storage, it can take up to 20 minutes before the new instance is available.



Create database

Choose a database creation method Info

○ Standard Create
You set all of the configuration options, including ones for availability, security, backups, and maintenance.

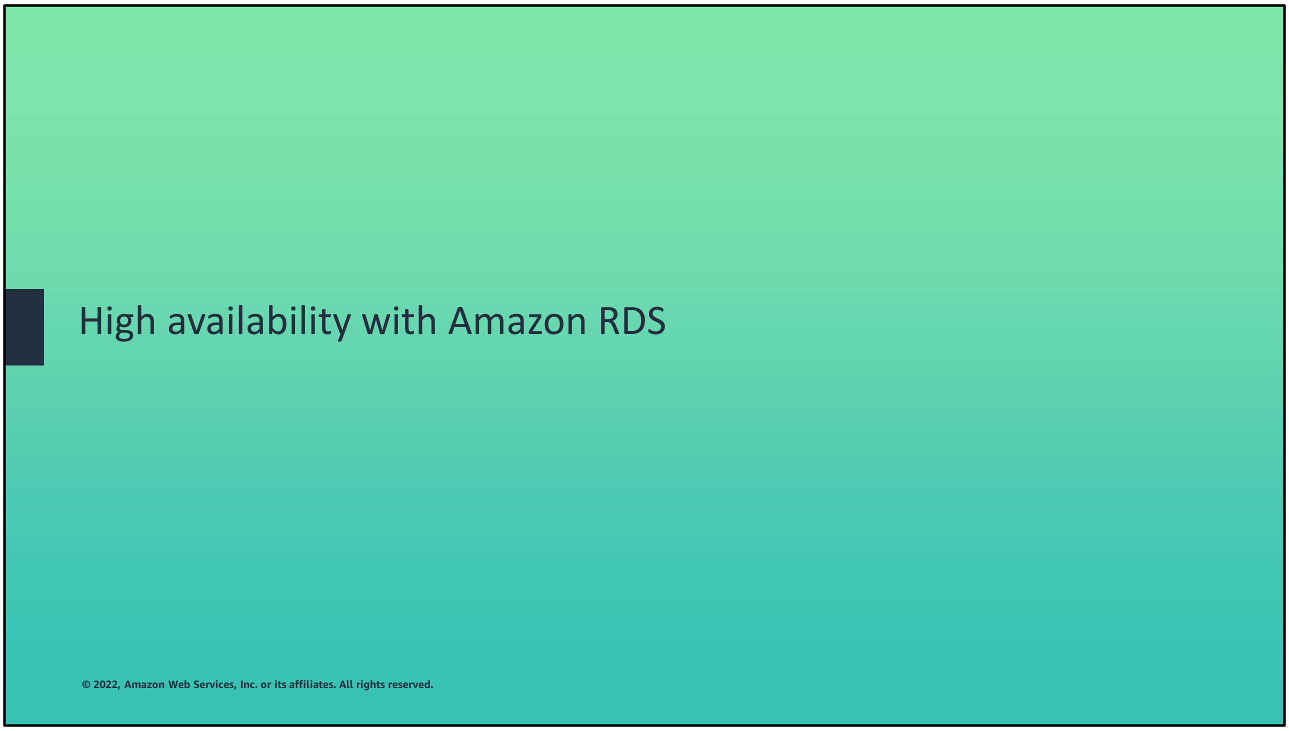
● Easy Create
Use recommended best-practice configurations. Some configuration options can be changed after the database is created.

Configuration

Engine type Info

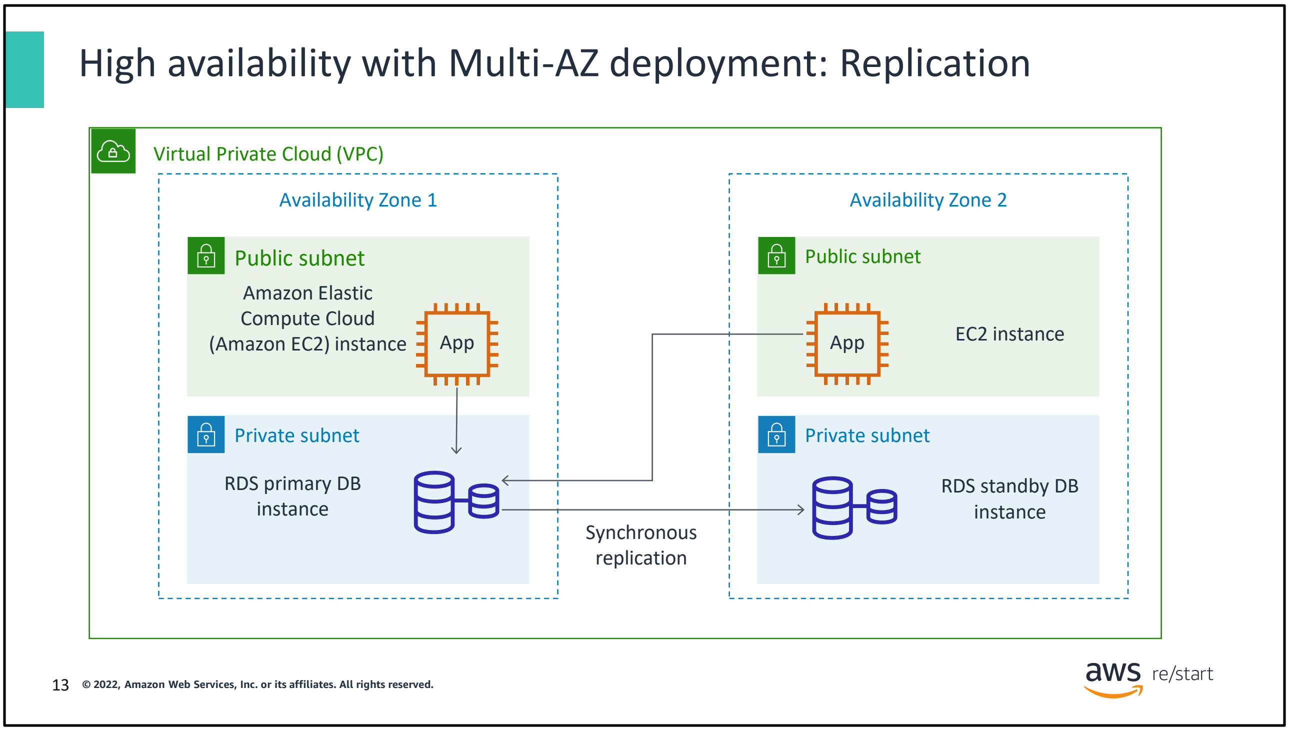
○ Amazon Aurora ● MySQL ○ MariaDB

This slide describes the main steps to create a DB instance using the Easy Create method.



High availability with Amazon RDS

This section covers high availability with Amazon RDS.



High availability with Multi-AZ deployment: Replication

Virtual Private Cloud (VPC)

Availability Zone 1

Public subnet

Amazon Elastic Compute Cloud (Amazon EC2) instance

App

Private subnet

RDS primary DB instance

Availability Zone 2

Public subnet

App

EC2 instance

Private subnet

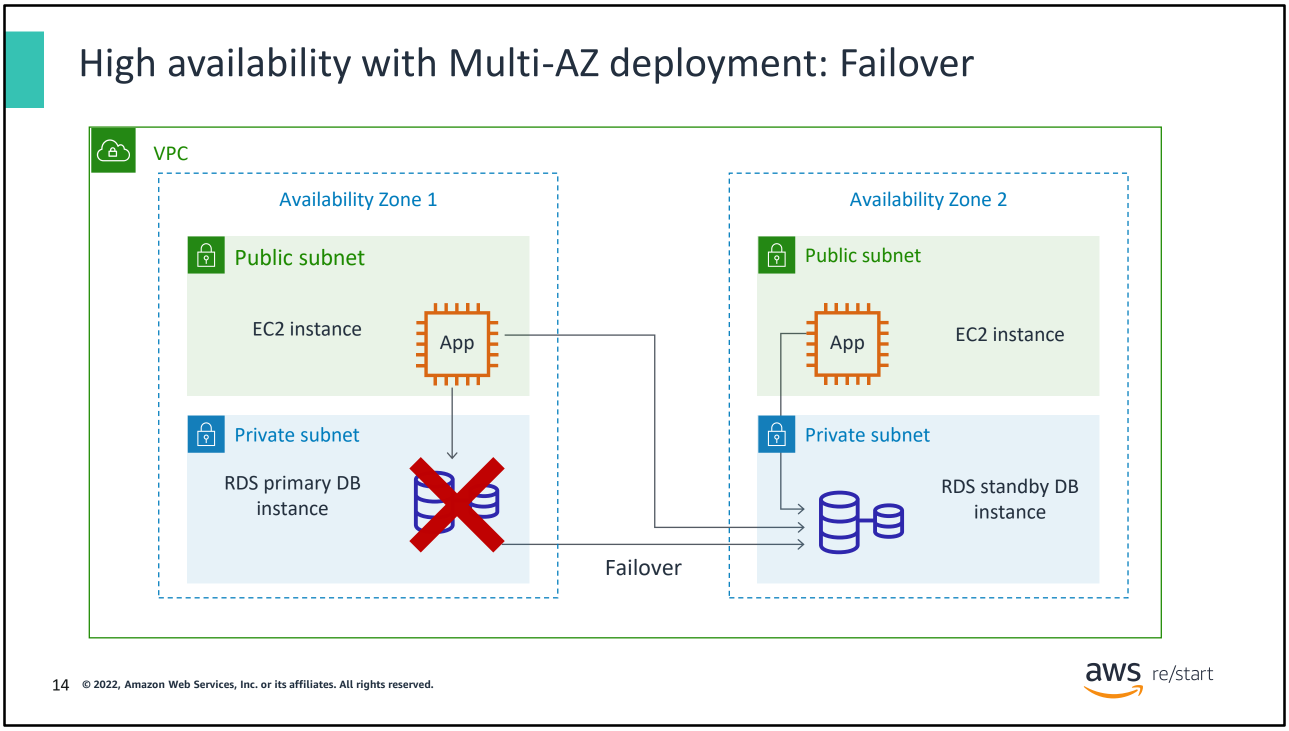
RDS standby DB instance

Synchronous replication

aws re/start

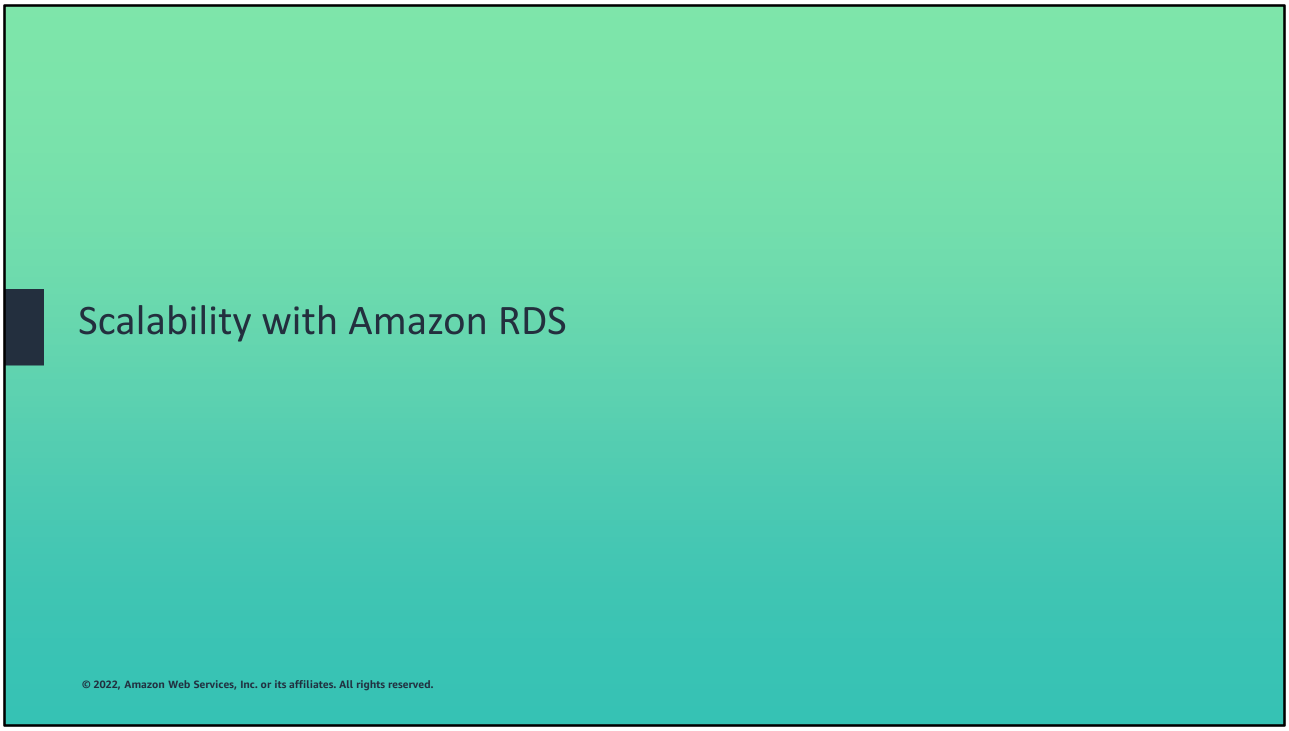
One of the main features of Amazon RDS is that you can configure a database instance with high availability by using a Multi-AZ deployment. This configuration automatically generates a standby copy of the database instance in another Availability Zone in the same virtual private cloud (VPC). After you make the initial full copy, transactions are synchronously replicated to the standby copy. Running a database in multiple Availability Zones can enhance availability during planned system maintenance. It can help protect your database against database failure and disruptions to Availability Zones.

In the diagram, an RDS DB instance is configured with high availability. The primary instance runs in Availability Zone 1, and it serves requests from applications that run in Availability Zones 1 and 2. The standby replica runs in Availability Zone 2, and it provides data redundancy. It is synchronously replicated from the primary instance.



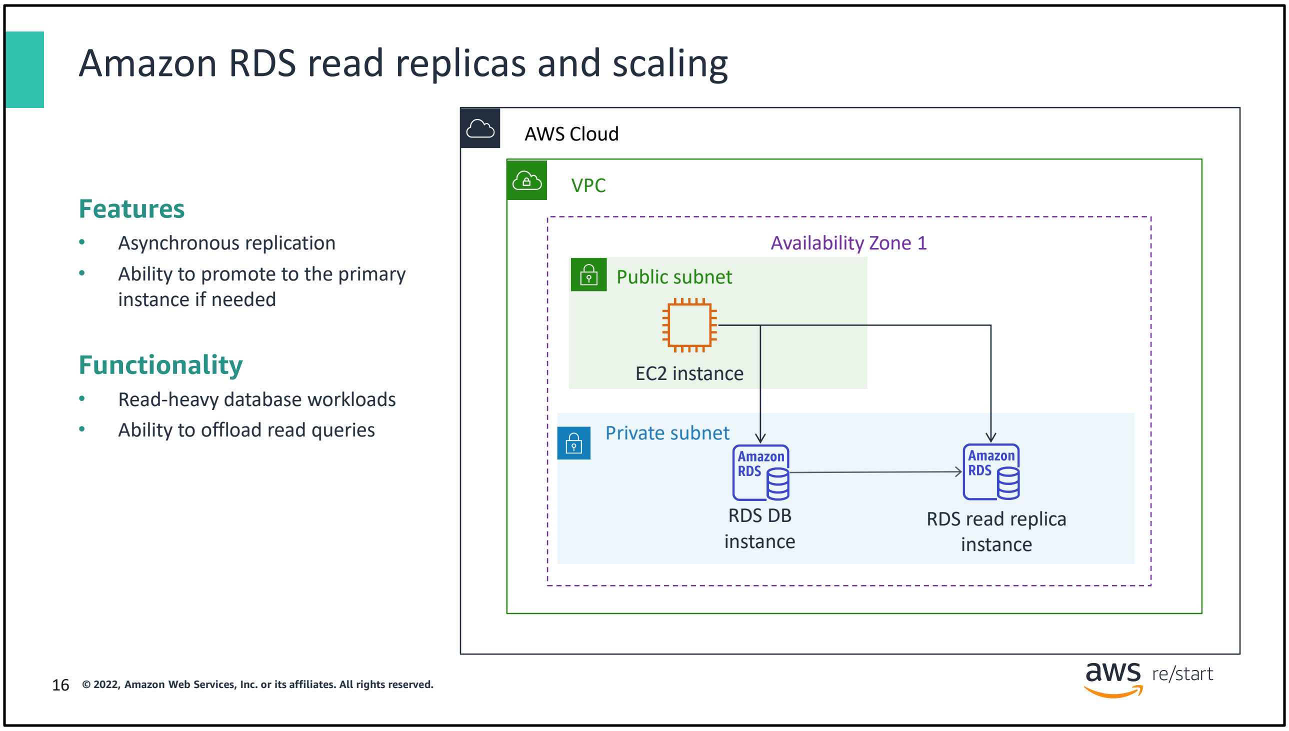
High availability with Multi-AZ deployment: Failover

If the primary database instance fails, Amazon RDS automatically brings the standby database instance online as the new primary instance. Requests from both applications are then directed to the new primary instance. The requesting applications use the Amazon RDS Domain Name System (DNS) endpoint to reference the database by name. As a result, the failover happens without needing to change the application code. Also, notice that no data loss occurs because of the synchronous replication.



Scalability with Amazon RDS

This section covers scalability with Amazon RDS.



Amazon RDS read replicas and scaling

Features

- Asynchronous replication
- Ability to promote to the primary instance if needed

Functionality

- Read-heavy database workloads
- Ability to offload read queries

In addition to a high availability configuration, Amazon RDS offers other ways of providing scalability. For example, you can create read replicas for MySQL, MariaDB, PostgreSQL, and Aurora. Updates to the source DB instance are asynchronously copied to the read replica instance. You can reduce the load on your source DB instance by routing read queries from your applications to the read replica. You can also use read replicas to scale out beyond the capacity constraints of a single DB instance for read-heavy database workloads.

Read replicas can also be promoted to become the primary DB instance. However, because it uses asynchronous replication, this option requires manual action.

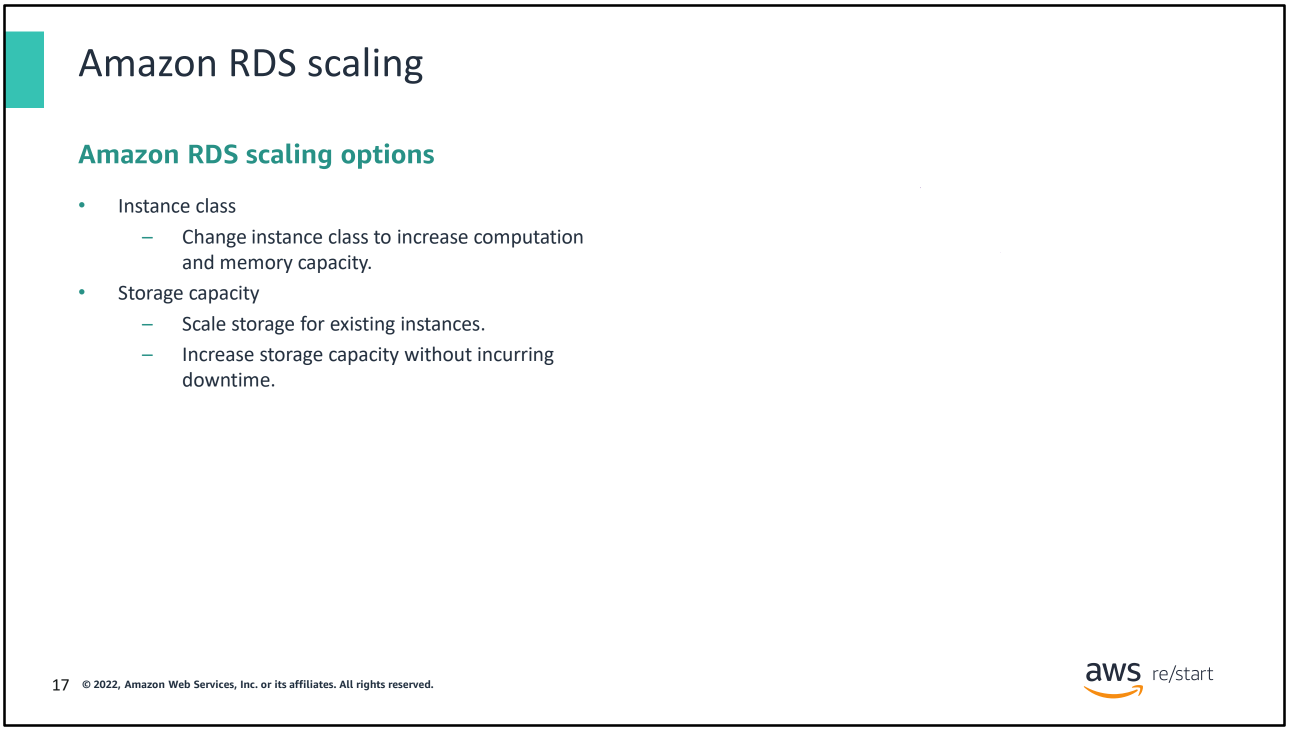
Read replicas can be created in a different Region from the primary DB instance. This feature can direct reads to a read replica in a geographic area that is closer to the user. Thus, it can help satisfy disaster recovery (DR) requirements or reduce latency.

For more information about scaling, refer to "Scaling Your Amazon RDS Instance Vertically and Horizontally" at https://aws.amazon.com/blogs/database/scaling-your-amazon-rds-instance-vertically-and-horizontally/.

For information about storage types, refer to "Working with storage for Amazon RDS DB instances" at
https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/USER_PIOPS.StorageTy

pes.html.

For information about read replicas, refer to Amazon RDS Read Replicas at https://aws.amazon.com/rds/details/read-replicas.



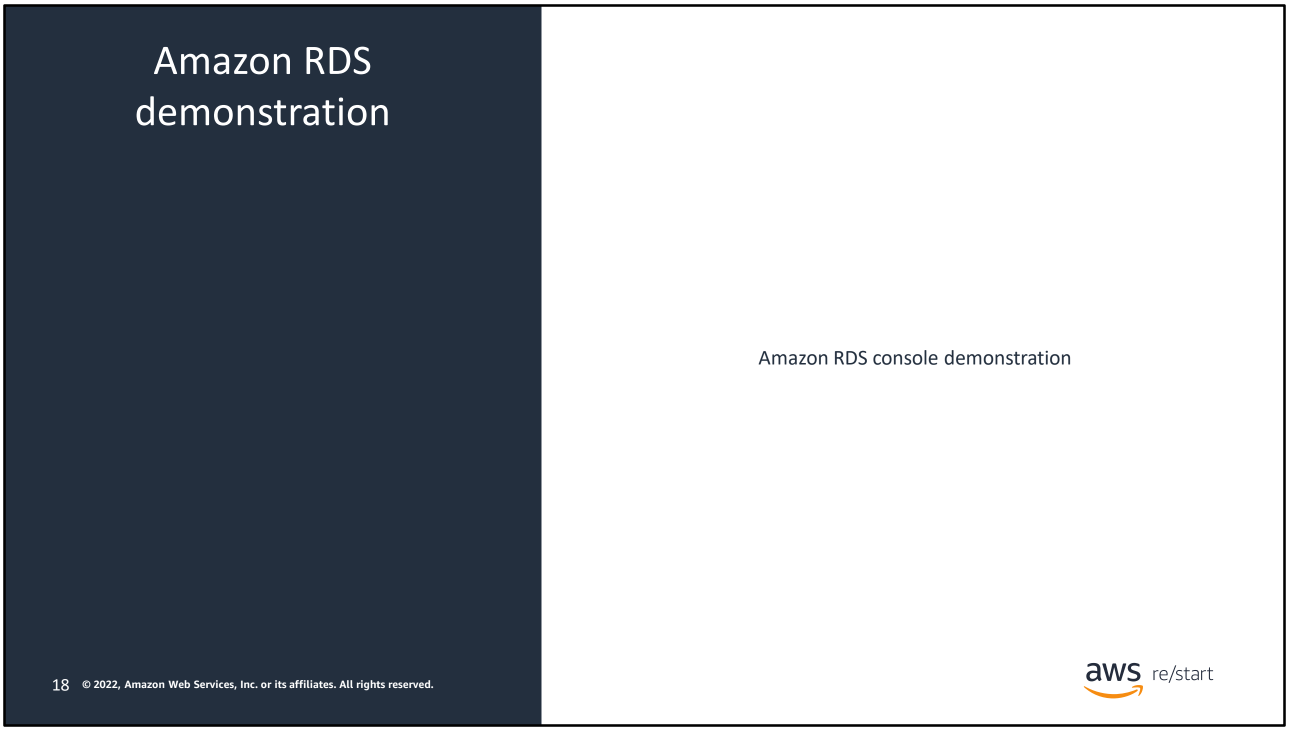
Amazon RDS scaling

Amazon RDS scaling options

- Instance class
  - Change instance class to increase computation and memory capacity.
- Storage capacity
  - Scale storage for existing instances.
  - Increase storage capacity without incurring downtime.

You can also increase the capacity of a database server by changing its instance class or storage capacity. By changing the instance class, you can increase the CPU and memory that are available to the instance. By modifying the allocated storage, you can increase storage capacity without incurring downtime.

Note that changing an instance class requires downtime.



Amazon RDS demonstration

Amazon RDS console demonstration

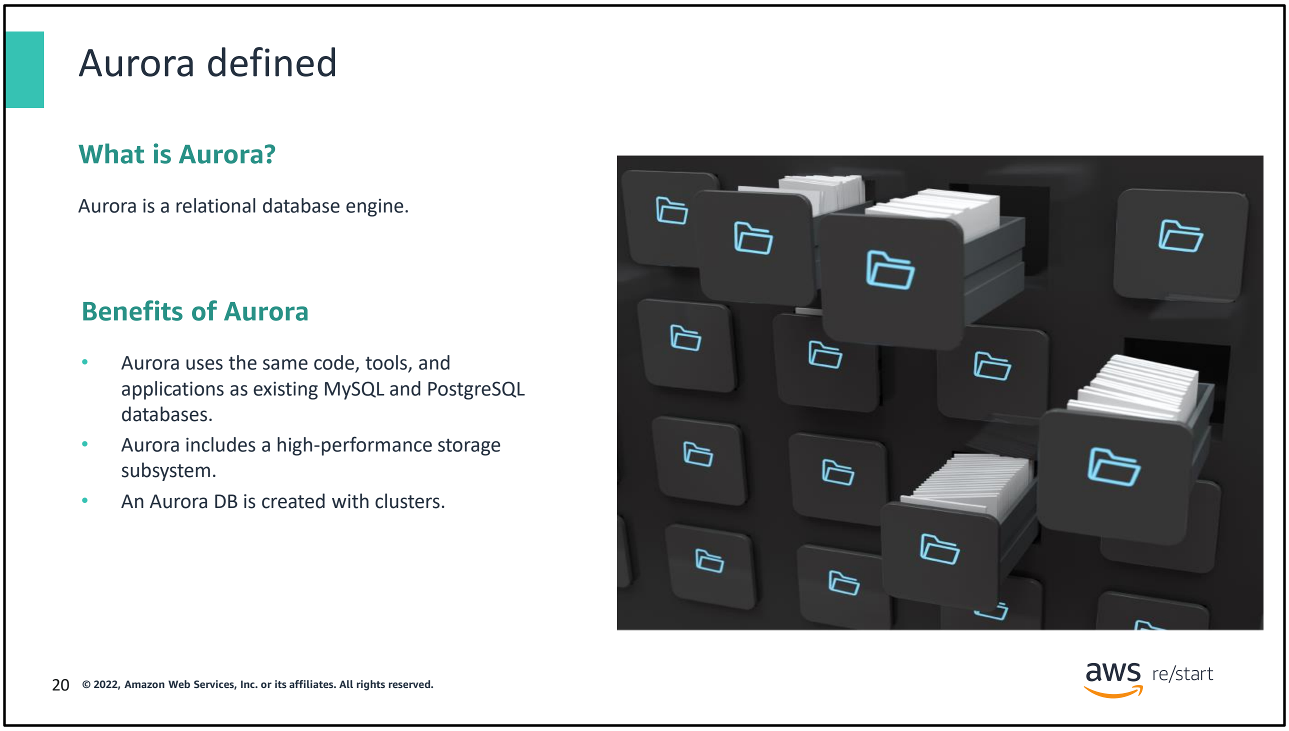
aws re/start

Review the Amazon RDS demonstration: Amazon Relational Database Service (RDS) Console Demo.

You can find this video demonstration in the learning management system.

Aurora

This section discusses Aurora.



Aurora defined

What is Aurora?

Aurora is a relational database engine.

Benefits of Aurora

- Aurora uses the same code, tools, and applications as existing MySQL and PostgreSQL databases.
- Aurora includes a high-performance storage subsystem.
- An Aurora DB is created with clusters.

What is Aurora?

- Aurora is part of the managed database service Amazon RDS. Aurora is a relational database engine.
- A database engine is a service for storing, processing, and securing data.

Some of the benefits of Aurora include the following:

- Aurora uses the same code, tools, and applications as existing MYSQL and PostgreSQL databases.
- Aurora includes a high-performance storage subsystem. Its database engine is customized to take advantage of that fast distributed storage.
- An Aurora DB has clusters that consist of one or more DB instances and a cluster volume that manages the data for those DB instances.

An Aurora DB cluster

Aurora DB cluster

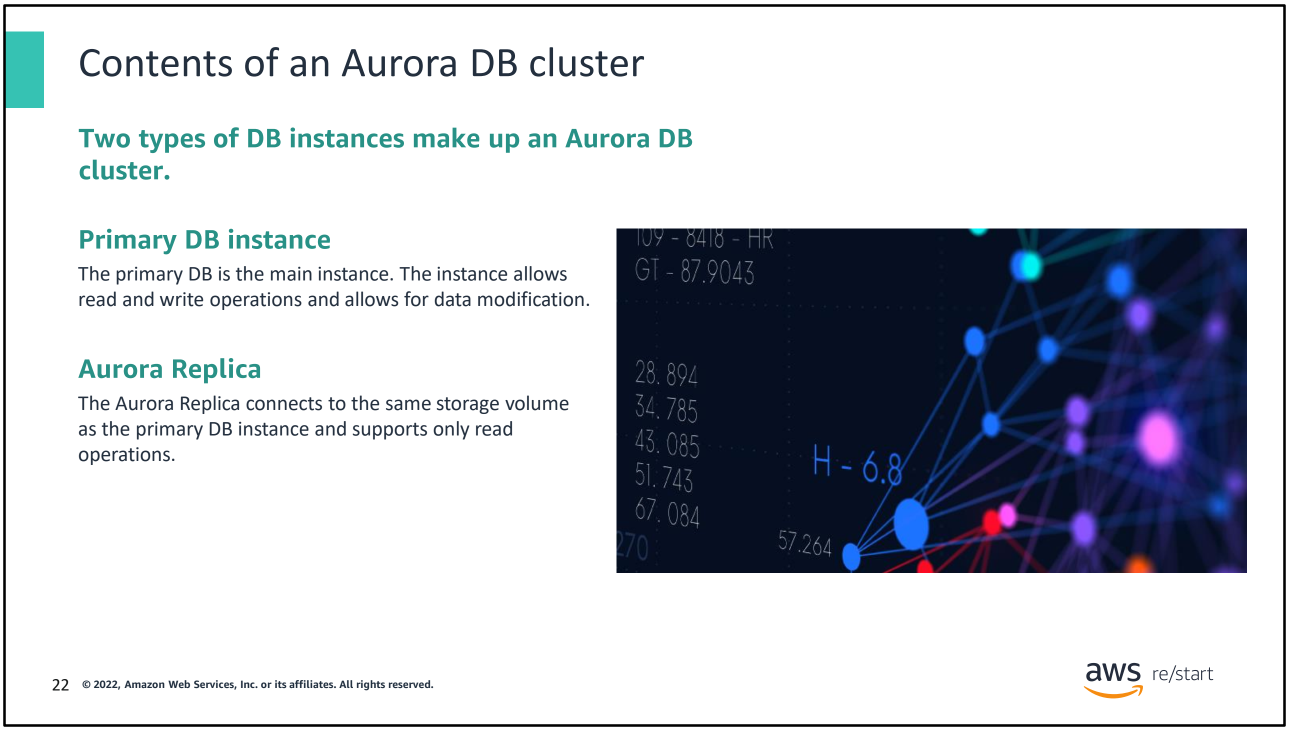
An Aurora DB cluster consists of one or more DB instances and a cluster volume that manages the data for those DB instances.

What is an Aurora DB cluster?

An Aurora DB cluster consists of one or more DB instances and a cluster volume that manages the data for those DB instances.

What is an Aurora cluster volume?

An Aurora cluster volume is a virtual database storage volume that spans multiple Availability Zones. Each Availability Zone has a copy of the DB cluster data.



Contents of an Aurora DB cluster

Two types of DB instances make up an Aurora DB cluster.

Primary DB instance

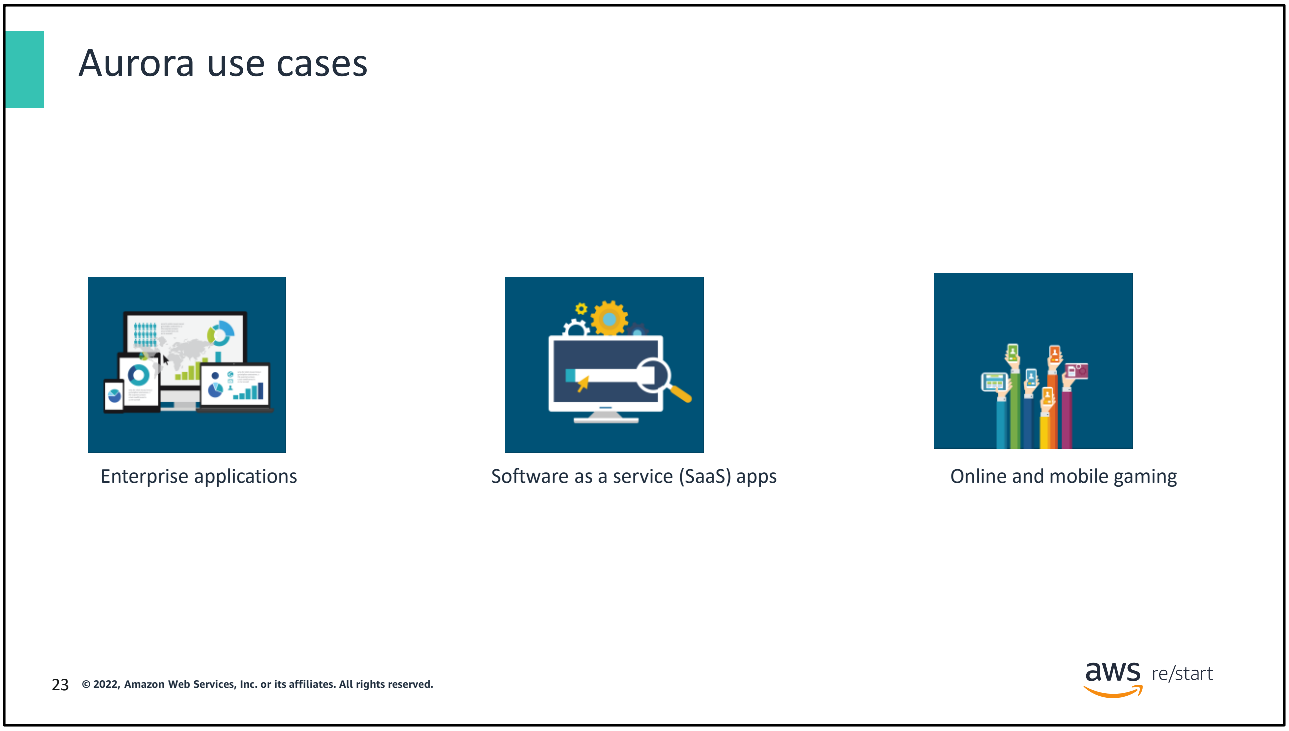
The primary DB is the main instance. The instance allows read and write operations and allows for data modification.

Aurora Replica

The Aurora Replica connects to the same storage volume as the primary DB instance and supports only read operations.

109 - 8418 - HR
GT - 87.9043
28. 894
34. 785
43. 085
51. 743
67. 084
270
H - 6.8
57.264

A primary DB instance and up to 15 Aurora replicas make up an Aurora DB cluster.



Aurora use cases

Enterprise applications

Software as a service (SaaS) apps

Online and mobile gaming

Enterprise applications

Compared to commercial databases, Aurora can help cut down your database costs by 90 percent or more while improving the database’s reliability and availability.

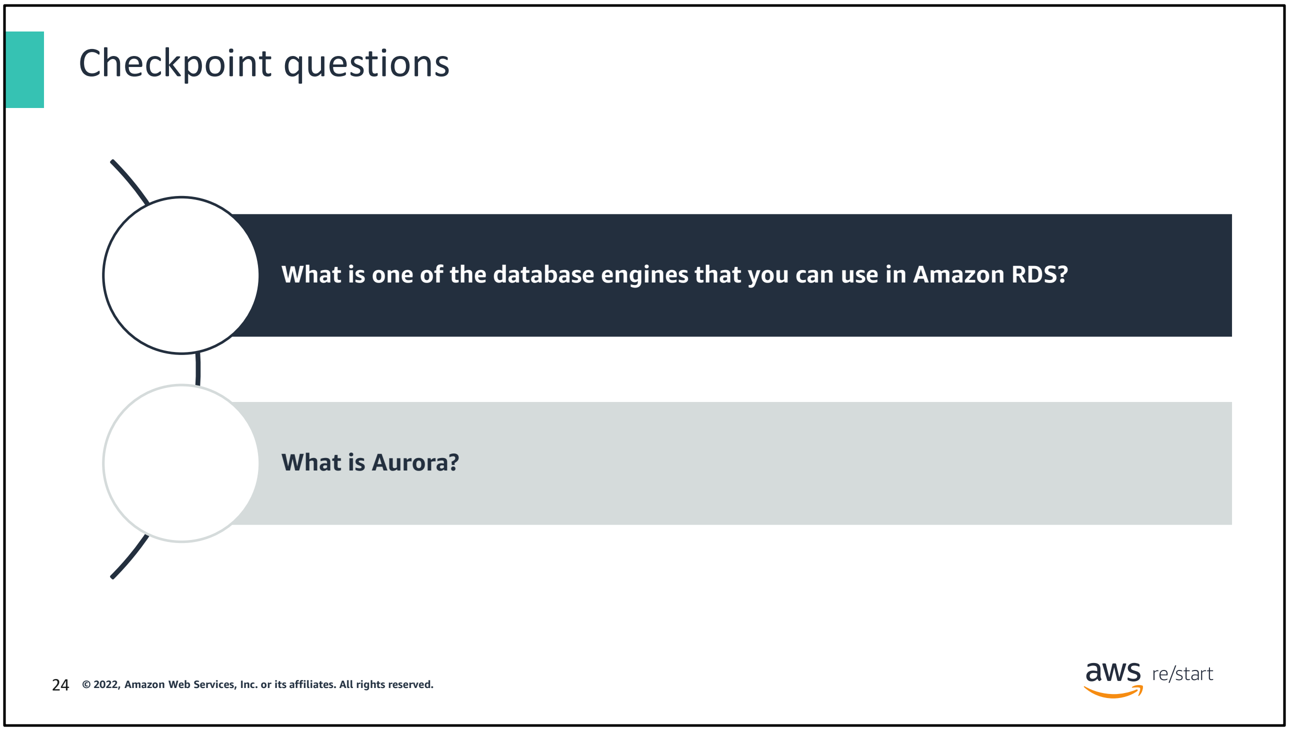
Software as a service (SaaS) applications

The Aurora managed database offering provides benefits to SaaS applications. SaaS companies can focus on building high-quality applications without worrying about the underlying database that powers the application.

Online and mobile gaming

Because web and mobile games are built to operate at very large scale, they require a database with high throughput and massive storage scalability.

Aurora provides enough room for future growth.



Checkpoint questions

What is one of the database engines that you can use in Amazon RDS?

What is Aurora?

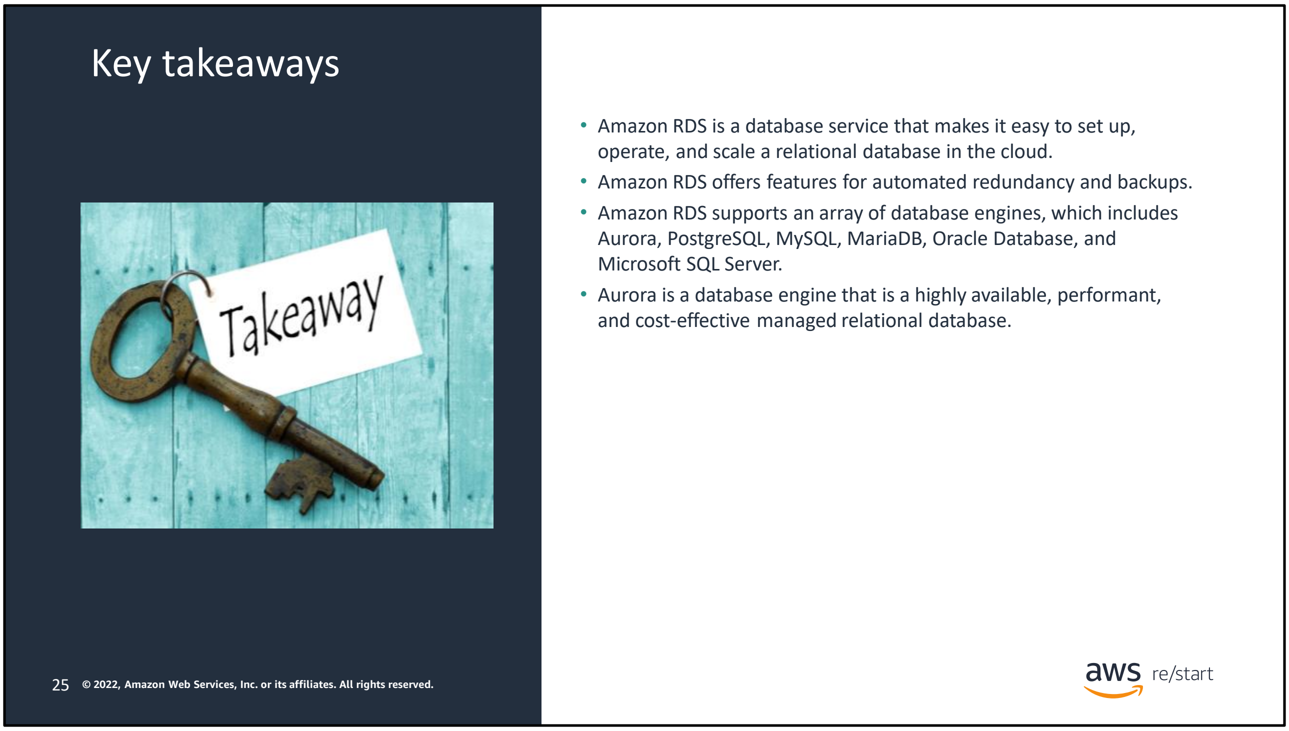
1. What is one of the database engines that you can use in Amazon RDS?

   Amazon RDS supports the following databases types:

   - MySQL
   - Aurora
   - Microsoft SQL Server
   - PostgreSQL
   - MariaDB
   - Oracle Database

2. What is Aurora?

   Aurora is a fully managed relational database engine that is compatible with MySQL and PostgreSQL.



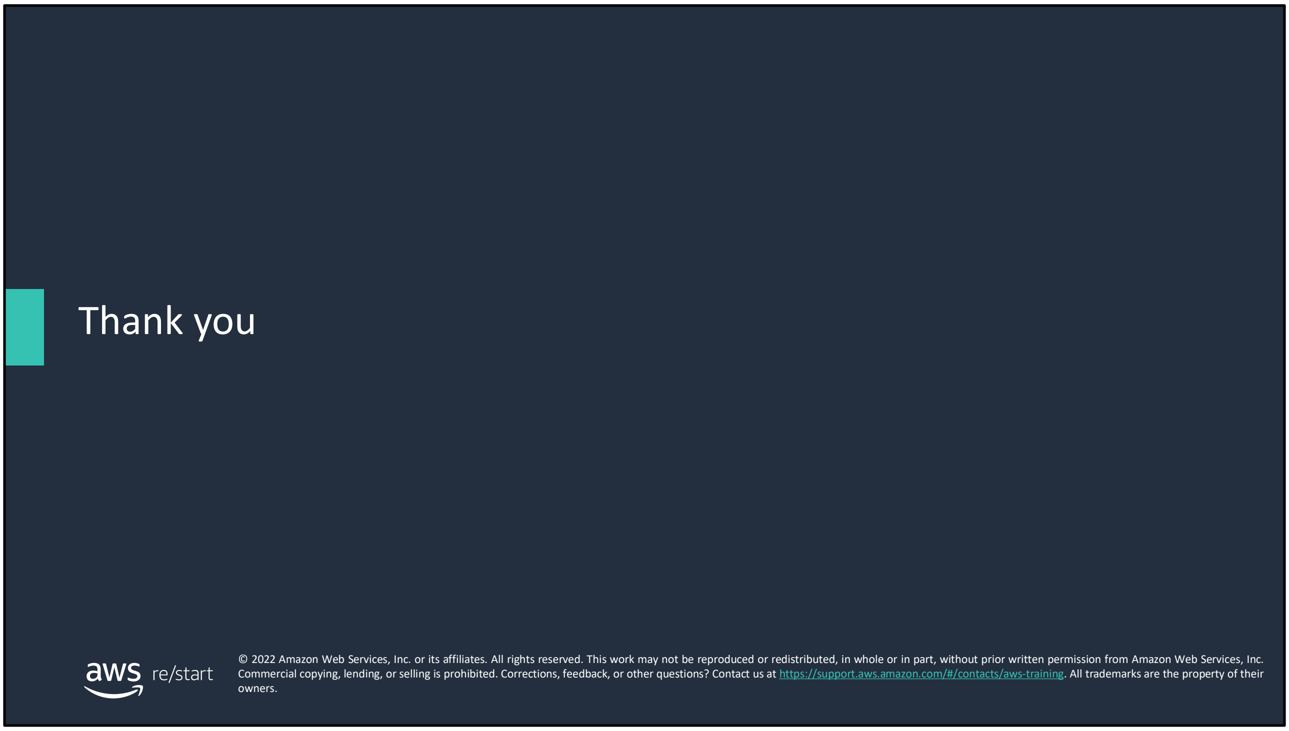
Key takeaways

- Amazon RDS is a database service that makes it easy to set up, operate, and scale a relational database in the cloud.
- Amazon RDS offers features for automated redundancy and backups.
- Amazon RDS supports an array of database engines, which includes Aurora, PostgreSQL, MySQL, MariaDB, Oracle Database, and Microsoft SQL Server.
- Aurora is a database engine that is a highly available, performant, and cost-effective managed relational database.

aws re/start

Key takeaways from this module include the following:

- Amazon RDS is a database service that makes it easy to set up, operate, and scale a relational database in the cloud.
- Amazon RDS offers features for automated redundancy and backups.
- Amazon RDS supports an array of database engines, which includes Aurora, PostgreSQL, MySQL, MariaDB, Oracle Database, and Microsoft SQL Server.
- Aurora is a database engine that is a highly available, performant, and cost-effective managed relational database.



Thank you

aws re/start

Thank you for completing this module.