**AWS RDS SERVICE**

**AWS Aurora DB**

Amazon Aurora is a fully managed relational database engine means this DB is created outside of EC2 instance and we do not need ec2 instance to use this developed by Amazon Web Services (AWS). It is designed to be compatible with MySQL and PostgreSQL, this is most reliable and availability as it keeps 6 replicas in 3 availability zone (AZ) providing high performance, scalability, and availability at a lower cost compared to traditional commercial databases.

**1. MySQL and PostgreSQL Compatibility:**

Aurora is compatible with MySQL and PostgreSQL, allowing applications already using these databases to be migrated without significant changes.

**Fully Managed**: Aurora is a managed service, so AWS handles tasks like database provisioning, backups, software patching, and hardware scaling

**2. High Performance:**

* Aurora offers up to five times better performance than standard MySQL databases and three times better than standard PostgreSQL databases.
* It achieves this with a distributed, fault-tolerant, self-healing storage system that automatically replicates data across multiple AWS Availability Zones.

**3. Scalability:**

* Aurora allows you to start with a smaller instance and automatically scale up to meet increased demand without downtime.
* It supports auto-scaling of read replicas to handle heavy read workloads and automatically distributes query traffic among multiple replicas.

**4. High Availability:**

* Aurora is designed for high availability, automatically replicating six copies of your data across three Availability Zones.
* It offers automatic failover support, meaning if the primary instance fails, a replica can take over quickly, ensuring minimal downtime.

**5. Automatic Backups and Point-in-Time Recovery:**

* Aurora continuously backs up your data to Amazon S3, providing automatic backups and point-in-time recovery to restore to any moment within the backup retention period.
* Snapshots and backups do not impact database performance.

**6. Security:**

* Aurora offers encryption at rest and in transit using AWS Key Management Service (KMS).
* You can configure Virtual Private Cloud (VPC) for network isolation, and Aurora integrates with AWS Identity and Access Management (IAM) for access control.

**7. Global Database:**

* Aurora allows you to set up a global database that spans multiple AWS regions, providing low-latency global reads and disaster recovery across regions.

**8. Cost-Efficient:**

* Aurora provides a pay-as-you-go pricing model, charging based on the resources you actually use (compute, storage, and I/O).
* It is cost-effective compared to traditional enterprise databases, with features designed to save costs through storage optimization and efficient resource management.

**Use Cases:**

* **Enterprise Applications**: Applications that need high availability, scalability, and performance, such as e-commerce platforms.
* **Web and Mobile Apps**: Applications that need fast, reliable database responses and scale as traffic grows.
* **Data Warehousing**: Aurora can also be used for data-intensive applications needing real-time analytics and querying.

**Amazon RDS and Aurora Documentation**

Amazon Relational Database Service (Amazon RDS) is a web service that makes it easier to set up, operate, and scale a relational database in the cloud. It provides cost-efficient, resizable capacity for an industry-standard relational database and manages common database administration tasks. Amazon Aurora is a fully managed relational database engine that's built for the cloud and compatible with MySQL and PostgreSQL. Amazon Aurora is part of Amazon RDS.

**What is Amazon Relational Database Service (Amazon RDS)?**

Amazon Relational Database Service (Amazon RDS) is a web service that makes it easier to set up, operate, and scale a relational database in the AWS Cloud. It provides cost-efficient, resizable capacity for an industry-standard relational database and manages common database administration tasks.

**Advantages of Amazon RDS**

Amazon RDS is a managed database service. It's responsible for most management tasks. By eliminating tedious manual processes, Amazon RDS frees you to focus on your application and your users.

Amazon RDS provides the following principal advantages over database deployments that aren't fully managed:

* You can use database engines that you are already familiar with: IBM Db2, MariaDB, Microsoft SQL Server, MySQL, Oracle Database, and PostgreSQL.
* Amazon RDS manages backups, software patching, automatic failure detection, and recovery.
* You can turn on automated backups, or manually create your own backup snapshots. You can use these backups to restore a database. The Amazon RDS restore process works reliably and efficiently.
* You can get high availability with a primary DB instance and a synchronous secondary DB instance that you can fail over to when problems occur. You can also use read replicas to increase read scaling.
* In addition to the security in your database package, you can control access by using AWS Identity and Access Management (IAM) to define users and permissions. You can also help protect your databases by putting them in a virtual private cloud (VPC).

**what is software patching in database**

Software patching in a database refers to the process of applying updates or modifications to the database software to address various issues such as security vulnerabilities, bugs, performance improvements, or adding new features. Patches are released by the database vendor and can be applied in the following forms:

1. **Security Patches**: Fix vulnerabilities that could be exploited by malicious users to gain unauthorized access or perform malicious actions.
2. **Bug Fixes**: Address software defects or errors that are causing incorrect behavior, crashes, or poor performance.
3. **Performance Enhancements**: Improve the efficiency of the database, optimizing query execution or other system operations.
4. **Feature Additions**: Introduce new functionalities or features to the database system, enabling users to take advantage of newer capabilities.

**Types of Patches:**

* **Critical Patch Updates (CPUs)**: These are typically released quarterly by vendors (like Oracle) and contain important fixes.
* **Service Packs**: Often cumulative, containing multiple patches rolled into a single update.
* **Hotfixes**: Smaller, targeted patches addressing very specific issues.

**Patch Application Process:**

* **Backup**: Always take a backup of the database before applying patches, as this allows for rollback in case of failure.
* **Testing**: Apply the patch in a test environment first to verify compatibility and stability.
* **Downtime Consideration**: Some patches may require the database to be offline, so planning for downtime might be necessary.
* **Monitoring**: After applying the patch, monitor the system for any unforeseen issues.

Patching is essential for maintaining database security, stability, and performance over time

**Comparison of responsibilities with Amazon EC2 and on-premises deployments**

**On-premises deployment**

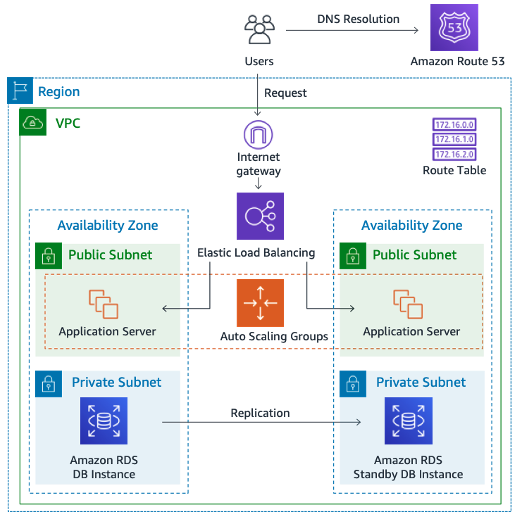
When you buy an on-premises server, you get CPU, memory, storage, and IOPS, all bundled together. You assume full responsibility for the server, operating system, and database software.

**Amazon EC2**

Amazon Elastic Compute Cloud (Amazon EC2) provides scalable computing capacity in the AWS Cloud. Unlike in an on-premises server, CPU, memory, storage, and IOPS are separated so that you can scale them independently. AWS manages the hardware layers, which eliminates some of the burden of managing an on-premises database server.

The disadvantage to running a database on Amazon EC2 is that you're more prone to user errors. For example, when you update the operating system or database software manually, you might accidentally cause application downtime. You might spend hours checking every change to identify and fix an issue.

**Amazon RDS DB instances Architecture**



**Elastic Load Balancing**

AWS routes user traffic through Elastic Load Balancing. A load balancer distributes workloads across multiple compute resources, such as virtual servers. In this sample use case, the Elastic Load Balancer forwards client requests to application servers.

**Application servers**

Application servers interact with RDS DB instances. An application server in AWS is typically hosted on EC2 instances, which provide scalable computing capacity. The application servers reside in public subnets with different Availability Zones (AZs) within the same Virtual Private Cloud (VPC). .

**RDS DB instances**

The EC2 application servers interact with RDS DB instances. The DB instances reside in private subnets within different Availability Zones (AZs) within the same Virtual Private Cloud (VPC). Because the subnets are private, no requests from the internet are permitted.

The primary DB instance replicates to another DB instance, called a *read replica*. Both DB instances are in private subnets within the VPC, which means that Internet users can't access them directly.

**DB engines**

A *DB engine* is the specific relational database software that runs on your DB instance. Amazon RDS supports the following database engines:

* IBM Db2

For more information, see [Amazon RDS for Db2](https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/CHAP_Db2.html).

* MariaDB

For more information, see [Amazon RDS for MariaDB](https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/CHAP_MariaDB.html).

* Microsoft SQL Server

For more information, see [Amazon RDS for Microsoft SQL Server](https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/CHAP_SQLServer.html).

* MySQL

For more information, see [Amazon RDS for MySQL](https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/CHAP_MySQL.html).

* Oracle Database

For more information, see [Amazon RDS for Oracle](https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/CHAP_Oracle.html).

* PostgreSQL

For more information, see [Amazon RDS for PostgreSQL](https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/CHAP_PostgreSQL.html).

[**https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/Welcome.html**](https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/Welcome.html)

**Aurora MySQL-Compatible Edition**

Aurora MySQL is Amazon’s enterprise-class MySQL-compatible database.

Aurora MySQL offers:

* Up to five times the throughput of MySQL Community Edition
* Up to 128 TB of autoscaling SSD storage
* Six-way replication across three Availability Zones
* Up to 15 read replicas with replica lag under 10-ms
* Automatic monitoring with failover

**Aurora PostgreSQL-Compatible Edition**

Aurora PostgreSQL is Amazon’s enterprise-class PostgreSQL-compatible database.

Aurora PostgreSQL offers:

* Up to three times the throughput of PostgreSQL
* Up to 128 TB of autoscaling SSD storage
* Six-way replication across three Availability Zones
* Up to 15 read replicas with replica lag under 10-ms
* Automatic monitoring with failover

**RDS MySQL**

MySQL is the most popular open source database in the world. MySQL on RDS offers the rich features of the MySQL community edition with the flexibility to easily scale compute resources or storage capacity for your database.

* Supports database size up to 64 TiB.
* Supports General Purpose, Memory Optimized, and Burstable Performance instance classes.
* Supports automated backup and point-in-time recovery.
* Supports up to 15 Read Replicas per instance, within a single Region or 5 read replicas cross-region.

**Steps to create Database**

**Choosing engine versions**

When you choose Aurora PostgreSQL, the global database feature is enabled for specific engine versions.

**Multi-AZ DB Cluster**

A Multi-AZ DB cluster has three DB instances. Each DB instance is in a different Availability Zone. A Multi-AZ DB cluster has one primary DB instance and two readable standby DB instances. Using a Multi-AZ DB cluster provides high availability, increased capacity for read workloads, and lower latency.

**Encryption key**

Choose the AWS KMS key identifier to encrypt a secret that is automatically generated and managed in AWS Secrets Manager. If you specify **aws/secretsmanager (default)**, then the KMS key provided by Secrets Manager is used to encrypt the secret.

**Deployment options applicable for all except AURORA DB**

**Multi-AZ DB cluster**

Creates a DB cluster with three DB instances. Each DB instance is in a different Availability Zone. A Multi-AZ DB cluster has one primary DB instance and two readable standby DB instances. Using a Multi-AZ DB cluster provides high availability, increased capacity for read workloads, and lower latency.

**Multi-AZ DB instance**

Creates a primary DB instance with one standby DB instance in a different Availability Zone. Using a Multi-AZ DB instance provides high availability, but the standby DB instance doesn't support connections for read workloads.

**Single DB instance**

Creates a single DB instance with no standby instances.

**DB instance class**

Choose the DB instance type that allocates the computational, network, and memory capacity required by planned workloads of this DB instance.

**Standard**  
Standard instances provide a balance of compute, memory, and network resources. They are a good choice for many database workloads.

**Memory optimized classes**  
Memory optimized instances accelerate performance for workloads that process large data sets in memory.

**Instance Store**  
If you use an m5d or r5d instance, you can store temporary tablespaces and Oracle Database Smart Flash Cache in the instance store. By using the Database Smart Flash Cache, you improve performance and reduce DB load. You also decrease storage and snapshot costs.

**Storage type**

**General purpose SSD (gp2)**

This storage is suitable for a broad range of database workloads. Provides the ability to burst to 3,000 IOPS. Baseline performance for these volumes is determined by the volume's size.

**General purpose SSD (gp3)**

A cost-effective option that supports a broad range of database workloads. Provides at least 3,000 IOPS and 125 MiBps baseline performance. IOPS and throughput can be scaled independently of the allocated storage whenever additional storage performance is needed for an added monthly cost.

**Provisioned IOPS (io1)**

This storage is suitable for I/O-intensive database workloads. Provides flexibility to provision I/O ranging from 1,000 to 256,000 IOPS.

**Provisioned IOPS (io2) - *new***

Amazon RDS io2 Block Express volumes, our next-generation Provisioned IOPs storage, provides high performance, high throughput, and consistent sub-millisecond latency for your production database workloads.

**Magnetic**

This storage is supported for backward compatibility. We recommend that you use General Purpose SSD or Provisioned IOPS SSD for any new storage needs.

**Connectivity**

Choose the connectivity settings for your database. These settings determine how client applications or utilities for your DB engine access the database after it is created.

The following are common scenarios for accessing a database in a VPC:

* **A database in a VPC accessed by an Amazon EC2 instance in the same VPC** – A common use of a database in a VPC is to share data with an application server that is running in an EC2 instance in the same VPC. The EC2 instance might run a web server with an application that interacts with the database.
* **A database in a VPC accessed by an EC2 instance in a different VPC** – When your database is in a different VPC from the EC2 instance that you're using to access it, you can use VPC peering to access the database.
* **A database in a VPC accessed by a client application through the internet** – To access a database in a VPC from a client application through the internet, you configure a VPC with a single public subnet, and an internet gateway to enable communication over the internet.
* **A database in a VPC accessed by a private network** – If your database isn't publicly accessible, you can use an AWS Site-to-Site VPN connection or an AWS Direct Connect connection to access it from a private network.

Database Authentication Options

**Password authentication**

Manage your database user credentials through your DB engine's native password authentication features. To learn more, see the documentation for your DB engine.

**Password and IAM database authentication**  
Manage your database user credentials through your DB engine's native password authentication features and IAM users and roles. IAM helps an administrator securely control access to AWS resources. IAM administrators control who can be authenticated and authorized for RDS resources.  
  
Learn More

* [RDS user guide](https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/UsingWithRDS.IAMDBAuth.html)
* [Aurora user guide](https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/UsingWithRDS.IAMDBAuth.html)

**Password and Kerberos authentication**  
Manage your database user credentials through your DB engine's native password authentication features and an AWS Managed Microsoft AD created with AWS Directory Service. This way, you can centrally store and manage user credentials, instead of individually for each DB instance.

**Aurora DB**

**DB instance class**

Choose the DB instance type that allocates the computational, network, and memory capacity required by planned workloads of this DB instance.

**Serverless**  
Serverless provides on-demand, auto-scaling configuration for Amazon Aurora. It automatically starts up, shuts down, and scales capacity up or down based on your application's needs.

**Memory optimized classes**  
Memory optimized instances accelerate performance for workloads that process large data sets in memory.

**Burstable classes**  
Burstable performance instances provide a baseline level of CPU performance with the ability to burst above the baseline.

Here are some common interview questions and answers related to AWS RDS and Aurora DB:

**1. What is Amazon RDS?**

**Answer**:  
Amazon RDS (Relational Database Service) is a managed service that simplifies the setup, operation, and scaling of relational databases in the cloud. It supports several database engines such as MySQL, PostgreSQL, MariaDB, Oracle, and Microsoft SQL Server. RDS automates tasks like backups, software patching, and hardware provisioning, allowing users to focus on their applications.

**2. What is Amazon Aurora? How is it different from RDS?**

**Answer**:  
Amazon Aurora is a fully managed relational database engine designed for high performance and availability, compatible with MySQL and PostgreSQL. It offers up to 5x the performance of standard MySQL and 3x for PostgreSQL, while being cost-effective. Aurora also provides features like automatic storage scaling, replication across multiple AZs, and continuous backups to Amazon S3.  
Differences:

* **Performance**: Aurora is more optimized and delivers higher throughput.
* **Storage Scaling**: Aurora automatically scales storage up to 128 TB without downtime.
* **Availability**: Aurora replicates data across multiple availability zones (AZs).

**3. How does Amazon RDS handle backups?**

**Answer**:  
RDS offers two types of backups:

* **Automated Backups**: This feature enables point-in-time recovery for your DB instance. RDS automatically performs a full daily backup and stores transaction logs. You can specify a retention period, and RDS allows recovery within that period.
* **Manual Snapshots**: Users can take manual snapshots, which are retained until explicitly deleted.

**4. How does Amazon Aurora achieve high availability and durability?**

**Answer**:  
Aurora replicates data six ways across three availability zones (AZs). It continuously backs up data to Amazon S3 and can perform instant failover to another AZ in case of hardware or software failure. Aurora's storage is fault-tolerant and self-healing, automatically recovering from disk failures.

**5. What is the difference between RDS Multi-AZ and Aurora Replicas?**

**Answer**:

* **RDS Multi-AZ**: Provides high availability by automatically replicating your primary database instance to a standby instance in a different AZ. It offers automatic failover but not load balancing.
* **Aurora Replicas**: Aurora supports up to 15 read replicas, which are designed for load balancing and high availability. It can promote one of the replicas to the primary instance if the original fails.

**6. What are DB Parameter Groups in RDS?**

**Answer**:  
DB Parameter Groups in RDS act as configuration groups that control database engine settings. These parameters are applied to all instances in a specific RDS instance class, allowing customization of various database settings, such as buffer cache size, connection timeout, etc.

**7. What are the benefits of using Amazon Aurora over MySQL/PostgreSQL on RDS?**

**Answer**:

* **Performance**: Aurora is up to 5x faster than MySQL and 3x faster than PostgreSQL.
* **Scaling**: Aurora automatically scales storage as needed.
* **High Availability**: Aurora replicates data across three AZs with instant failover.
* **Backup and Recovery**: Continuous backups to S3 with point-in-time recovery.
* **Read Replicas**: Aurora supports up to 15 replicas for MySQL and 5 for PostgreSQL, allowing for read scalability.

**8. How do you monitor the performance of RDS/Aurora?**

**Answer**:  
You can monitor performance using several tools:

* **Amazon CloudWatch**: Provides metrics such as CPU, memory, disk I/O, and network throughput.
* **Enhanced Monitoring**: Offers real-time metrics at the instance level, including operating system processes.
* **Performance Insights**: Helps analyze database performance by providing details on queries and database load.

**9. What is the difference between Amazon RDS Read Replicas and Multi-AZ deployments?**

**Answer**:

* **Read Replicas**: Used primarily for read scalability. They replicate data from the source DB instance and allow you to distribute read traffic, but do not provide automatic failover.
* **Multi-AZ Deployments**: Designed for high availability and disaster recovery. The data is synchronously replicated to a standby instance in a different AZ, and automatic failover occurs in case of failure.

**10. What is RDS IAM Authentication?**

**Answer**:  
RDS IAM Authentication allows you to authenticate to your RDS database instance using AWS Identity and Access Management (IAM) instead of a database password. It is useful for managing database access and eliminating the need to store database credentials in your application code.

**11. Can you explain Amazon Aurora Global Databases?**

**Answer**:  
Aurora Global Database allows a single Aurora database to span multiple AWS regions, providing fast local reads in the region and disaster recovery across regions. Data is replicated with low latency across regions, and in the event of regional degradation, you can promote another region’s database to be the master.

**12. What is the significance of database failover in AWS RDS?**

**Answer**:  
In Multi-AZ RDS deployments, failover occurs automatically when the primary database instance becomes unavailable due to a hardware failure, OS crash, or other issues. RDS promotes the standby instance to become the new primary, minimizing downtime and ensuring high availability.

**13. How do you encrypt data in Amazon RDS and Aurora?**

**Answer**:  
Both Amazon RDS and Aurora support encryption at rest using AWS Key Management Service (KMS). When enabled, RDS/Aurora encrypts the underlying storage, automated backups, snapshots, and replicas. Data in transit is encrypted using SSL/TLS.