

Fundamental Concepts

1. What is AWS RDS?

- **Answer:** AWS RDS (Relational Database Service) is a managed web service that makes it easier to set up, operate, and scale a relational database in the AWS Cloud. It provides cost-efficient and resizable capacity while automating time-consuming administration tasks such as hardware provisioning, database setup, patching, and backups.

2. Why would you use RDS instead of installing and managing your own database on an EC2 instance?

- **Answer:** The primary reason is that RDS is a **managed service**. This means AWS handles many of the administrative burdens, allowing developers to focus on their applications rather than database management. Key benefits include:
 - **Automated Patching & Updates:** AWS handles OS and database engine patching during defined maintenance windows.
 - **Automated Backups & Point-in-Time Recovery:** RDS automatically backs up your database and transaction logs, allowing you to restore to any point within the retention period.
 - **High Availability (Multi-AZ):** Easily configure synchronous replication to a standby instance in a different Availability Zone for failover.
 - **Scalability:** Easily scale compute resources (CPU/RAM) up or down (vertical scaling) or increase read capacity using Read Replicas (horizontal scaling for reads). Storage can also be scaled.
 - **Security:** Simplifies security configuration with features like Security Groups, encryption at rest and in transit.
 - **Monitoring:** Integrated with CloudWatch for monitoring database metrics.

3. Which relational database engines does RDS support? (Name a few)

- **Answer:** RDS supports several popular database engines, including:
 - Amazon Aurora (MySQL and PostgreSQL compatible editions)
 - MySQL
 - MariaDB
 - PostgreSQL
 - Oracle
 - Microsoft SQL Server

4. What is a DB Instance in RDS?

- **Answer:** A DB Instance is the basic building block of RDS. It's an isolated database environment running in the cloud. A DB instance contains the database engine you choose and can be accessed using standard database tools and applications. It has associated compute (CPU, memory) and storage resources that you define.

High Availability and Scalability

1. What is Multi-AZ deployment in RDS? What is its main purpose?

- **Answer:** Multi-AZ (Multi-Availability Zone) deployment is an RDS feature for **high availability and durability**. When enabled, RDS automatically provisions and maintains a synchronous standby replica of your primary database instance in a *different* Availability Zone within the same region.
- **Purpose:** Its main purpose is **failover support**. If the primary DB instance fails (due to hardware failure, AZ disruption, etc.), RDS automatically fails over to the standby replica, typically within 1-2 minutes, without manual intervention. The application endpoint remains the same. It also helps during maintenance, as patching can occur on the standby first. This is *not* primarily for scaling read traffic.

2. What is an RDS Read Replica? What is its main purpose?

- **Answer:** An RDS Read Replica is a read-only copy of your primary database instance. RDS uses the database engine's native asynchronous replication feature to create and maintain these replicas. You can create multiple Read Replicas from a primary DB instance.
- **Purpose:** Its main purpose is **read scaling**. Applications can direct read-heavy traffic to the Read Replicas, reducing the load on the primary DB instance which handles write operations. This improves overall application performance for read-intensive workloads. Read Replicas can also be promoted to become standalone primary instances if needed, although this is a manual process.

3. What is the key difference between Multi-AZ and Read Replicas?

- **Answer:**
 - **Purpose:** Multi-AZ is for **high availability (HA)** and **failover**. Read Replicas are for **read scalability**.
 - **Replication:** Multi-AZ uses **synchronous** replication (ensuring data consistency before confirming a write). Read Replicas use **asynchronous** replication (potential for minor replication lag).
 - **Standby Access:** The Multi-AZ standby instance **cannot** be accessed directly for read/write traffic; it only becomes active during a failover. Read Replicas **can** be accessed directly for read queries.
 - **Location:** A Multi-AZ standby is always in a different AZ within the *same* region. Read Replicas can be in the same AZ, a different AZ, or even a different *region* (cross-region Read Replicas).
 - **Failover:** Multi-AZ failover is **automatic**. Promoting a Read Replica to primary is a **manual** process.

4. How can you scale an RDS database?

- **Answer:** RDS offers several scaling options:
 - **Vertical Scaling (Scale Up/Down):** You can modify the DB instance class to increase or decrease CPU and memory resources. This usually involves a brief downtime while the instance is resized.

- **Storage Scaling:** You can increase the allocated storage space, often without downtime (depending on the engine and circumstances). Some engines also allow changing storage type (e.g., from gp2 to io1 for better IOPS).
- **Horizontal Read Scaling (Scale Out):** You can add Read Replicas to handle increasing read traffic, distributing the read load away from the primary instance.

Backups and Maintenance

1. How does RDS handle database backups?

- **Answer:** RDS provides two methods for backing up DB instances:
 - **Automated Backups:** Enabled by default. RDS automatically creates daily snapshots of your DB instance and captures transaction logs (which enable point-in-time recovery). You define a backup retention period (typically 1-35 days). Restoring creates a *new* DB instance from the snapshot or point in time.
 - **Manual Snapshots (DB Snapshots):** You can take manual snapshots of your DB instance at any time. These snapshots are kept until you explicitly delete them, even if you delete the original DB instance. Useful for creating baseline copies before major changes or for long-term archival beyond the automated backup retention period.

2. What is Point-in-Time Recovery (PITR)?

- **Answer:** Point-in-Time Recovery allows you to restore your database to any specific second within your configured backup retention period (up to the last 5 minutes typically). RDS uses the combination of the automated daily snapshots and the continuously backed-up transaction logs to achieve this. When you perform a PITR, RDS creates a new DB instance with the data restored to the exact moment you specified.

3. What is an RDS Maintenance Window?

- **Answer:** The maintenance window is a period of time (e.g., 30 minutes weekly) that you define, during which AWS can apply pending modifications like DB engine patches, OS updates, or instance scaling operations. AWS tries to perform necessary maintenance during this window to minimize impact on your application. If a critical security patch is needed, AWS might apply it outside the window. Multi-AZ deployments help minimize downtime during maintenance, as updates are often applied to the standby first before failing over.

Security

1. How do you control network access to an RDS DB instance?

- **Answer:** Network access is primarily controlled using **Security Groups**. RDS instances are launched within a VPC (Virtual Private Cloud). You associate one or more VPC Security Groups with your RDS DB instance. These security groups act as a virtual firewall, defining rules that specify which sources (e.g., specific EC2 instances, IP addresses) are allowed to connect to the database endpoint on specific ports (e.g., 3306 for MySQL, 5432 for PostgreSQL). By default, RDS instances are not accessible from the public internet.

2. How can you secure data within RDS?

- **Answer:** RDS offers several security layers:
 - **Network Isolation:** Using VPCs and Security Groups to control network traffic.
 - **IAM Integration:** Using IAM policies to control who can manage RDS resources (create, delete, modify instances) via the AWS API/console. (Note: IAM is generally *not* used for direct database user authentication, though RDS for PostgreSQL and MySQL now offer IAM database authentication as an option).
 - **Encryption at Rest:** You can enable encryption for the underlying storage of your DB instance and its automated backups, replicas, and snapshots using AWS KMS (Key Management Service). This needs to be enabled when the instance is created.
 - **Encryption in Transit:** You can enforce SSL/TLS connections between your application and your DB instance to encrypt data moving over the network. RDS provides the necessary certificates.
 - **Database Native Authentication:** Standard database username/password authentication managed within the database engine itself.

Scenario Questions

1. **Your web application's database is running on RDS. Users are complaining about slow load times, especially during peak hours when many users are reading data. Write operations are performing okay. What RDS feature could potentially help alleviate this issue?**
 - **Answer:** The most likely feature to help here is an **RDS Read Replica**. Since the issue is related to high *read* traffic impacting performance, offloading these read queries to one or more Read Replicas would reduce the load on the primary DB instance, allowing it to handle writes more efficiently and improving overall read performance for the application.
2. **You need to ensure your RDS database can survive a failure of an entire Availability Zone with minimal downtime and no data loss. Which RDS feature should you configure?**
 - **Answer:** You should configure a **Multi-AZ deployment**. This feature creates a synchronous standby replica in a different AZ. If the primary AZ fails, RDS automatically fails over to the standby replica in the other AZ, ensuring high availability and minimizing data loss (due to synchronous replication).

Tips for Answering:

- **Highlight "Managed":** Emphasize the benefits of RDS being a managed service (less operational overhead).
- **Know the Key Features:** Be solid on Multi-AZ, Read Replicas, Automated Backups, and Security Groups – these are core concepts.
- **Distinguish Features:** Clearly explain the difference between features serving similar but distinct purposes (e.g., Multi-AZ vs. Read Replicas).
- **Use Correct Terminology:** Use terms like DB Instance, Endpoint, Multi-AZ, Read Replica, Security Group, Snapshot correctly.

- **Focus on Purpose:** Explain *why* a feature exists (e.g., HA for Multi-AZ, read scaling for Replicas).

Good luck with your interview! A good understanding of RDS demonstrates knowledge of managed services and database fundamentals in the cloud.