Operational Management => High Availability & Resilience

In RDS, Multi-AZ deployments versus Read Replicas - when to use each?
How can RDS handle failover and disaster recovery?

❓ **1. Explain Multi-AZ deployments and their benefits in RDS**

✅ **What is Multi-AZ?**

- In **Multi-AZ**, AWS maintains **one primary DB** + a **synchronous standby replica** in a different Availability Zone.
- It's not for scaling reads — it's for **high availability & failover**.

✅ **Benefits:**

- **Automatic Failover**: If the primary goes down, RDS flips DNS to standby → app reconnects with minimal downtime.
- **No data loss**: Because replication is synchronous.
- **Automatic Backups** happen on standby → less load on primary.

👉 Example: In MySQL Multi-AZ, AWS manages primary in us-east-1a, standby in us-east-1b. If 1a fails, DB in 1b becomes active.

❓ **2. What are RDS Read Replicas and how do you implement them?**

✅ **What is a Read Replica?**

- A **read-only copy** of the DB, updated using **asynchronous replication**.
- Purpose: Offload **read-heavy workloads**, scale horizontally.

✅ **Use Cases:**

- Analytics, reporting queries.
- Scaling reads for high-traffic apps.
- Cross-region replicas for **global read access**.

✅ **How to Implement:**

1. Go to RDS → Select your DB.
2. Click **Create Read Replica**.
3. Choose region, instance type, and storage.
4. AWS creates a new DB instance (replica endpoint).

5. Point read queries to replica endpoint.

👉 Note: Read replicas can be promoted to standalone DBs (useful in disaster recovery).

❓ 3. Multi-AZ Deployments vs Read Replicas

Feature	Multi-AZ	Read Replica
Replication	Synchronous	Asynchronous
Purpose	High availability, failover	Scalability (offload read traffic)
Writes	Only primary accepts writes	Replicas = read-only
Failover	Automatic (DNS switch)	Not automatic (unless promoted)
Cross-region	No (same region only)	Yes (cross-region replicas allowed)
Use Case	Reliability & DR	Performance & global distribution

👉 Simple analogy:

- **Multi-AZ** = **insurance policy** (your DB won't go down).
- **Read Replica** = **extra cash counters** (scale for customer load).

⚙️ Synchronous replication — why and how

Synchronous replication means every transaction must be written to both:

1. The **primary** DB in AZ-A
2. The **standby** DB in AZ-B

   …before the transaction is confirmed as "committed".

✅ **Example:**

Suppose you run this SQL:

INSERT INTO orders VALUES (101, 'Laptop', 70000);

In Multi-AZ:

1. The data is written to the **primary DB** in us-east-1a.
2. Before confirming success, RDS ensures the **same transaction** is successfully written to the **standby** in us-east-1b.
3. Only then does it send an "OK" back to the application.

👉 If AZ-A fails, RDS automatically fails over to AZ-B **without data loss**, because both had the same data at all times.

That's why it's **synchronous** — it ensures **zero data loss (RPO = 0)** but **slightly higher latency** because it waits for acknowledgment from both sides.

🌐 2️⃣ What Read Replica means

Read Replica is a **performance and scalability setup**, not for failover.

- You create one or more replicas in **same or different AZs/Regions**.
- They are used for **read queries** (not writes).
- The **replication** is **asynchronous**.

⚙️ Asynchronous replication — why and how

Asynchronous replication means the primary DB does **not wait** for the replica to confirm data writing.

✅ **Example:**

Continuing our earlier example:

INSERT INTO orders VALUES (102, 'Phone', 25000);

1. The transaction is written and committed immediately on the **primary DB**.
2. The **replica** (in another AZ or region) receives this change later — maybe a few seconds later — depending on replication lag.

👉 This means if the **primary fails**, your **replica might not have the last few transactions** (possible data loss).
That's why it's **asynchronous** — it gives **better performance and scalability** (no wait time), but **not zero data loss**.

How RDS handles Failover and Disaster Recovery

✅ **1. Failover (Automatic High Availability)**

- **Used for short-term failures** — like if your main database instance or its Availability Zone goes down.
- In **Multi-AZ setup**, RDS automatically:
  - Detects that your main database (primary) has failed.
  - Switches to the **standby database** in another Availability Zone.
  - Updates the **DNS name** so your app automatically connects to the new one.
- This process usually finishes in **30 to 120 seconds** (1–2 minutes).
- You don't have to do anything — it happens automatically.

Example:
If your primary DB in us-east-1a crashes, RDS quickly switches to the standby DB in us-east-1b, and your app keeps running after a short pause.

✅ **2. Disaster Recovery (DR)**

- **Used for big failures** — like an entire AWS region going down.
- To prepare for this, you can use these options:

1. **Cross-Region Read Replica**
   - Create a **read-only copy** of your database in another region (e.g., us-west-2).
   - If the main region (e.g., us-east-1) fails, you can **promote** the replica to become a **new main database**.
   - This helps you get your service back quickly in another region.
2. **Snapshot Copy Across Regions**
   - You can **take backups (snapshots)** of your database and copy them to another region.
   - If your main region fails, you can **restore** the DB from that snapshot in a new region.
   - This is slower than replicas but cheaper.
3. **Aurora Global Database** (for Aurora users)
   - Aurora can **replicate data across regions in under a second**.
   - If one region fails, another region can take over **almost instantly**.

💡 **Example**
Let's say your main RDS database is in **us-east-1** (Virginia).

- If the AZ fails → **Multi-AZ failover** happens automatically within 1–2 minutes.
- If the **whole region fails** → You can **promote your cross-region replica** in **us-west-2** (Oregon) to make it your new main database.

🧠 **In short:**

Situation	Feature	What Happens
Single AZ fails	**Multi-AZ Failover**	Auto switch to standby (1–2 min)
Whole region fails	**Disaster Recovery**	Use replica or snapshot in another region

---

🎯 **Short Interview Version (2–3 lines each)**

- **Multi-AZ**: Provides HA with synchronous replication and automatic failover — good for reliability.
- **Read Replicas**: Asynchronous copies for scaling reads — good for performance & analytics.
- **When to use**: Multi-AZ = uptime guarantee, Read Replica = performance scaling.
- **Failover/DR**: AWS automates Multi-AZ failover; for DR across regions, use cross-region replicas or snapshot copies.

👉 **Level:**

- **Beginner** → "Multi-AZ = HA, Read Replicas = scaling."
- **Mid-level** → Explains sync vs async + use cases.
- **Senior** → Talks **cross-region DR strategy + Aurora global DBs**.

What is the **main purpose** of a Multi-AZ deployment in Amazon RDS?
A. To improve read query performance
B. To provide high availability and automatic failover
C. To scale writes horizontally
D. To reduce storage costs
✅ **Correct Answer: B.**
💡 **Explanation: Multi-AZ** provides **high availability** by maintaining a **synchronous standby** in another Availability Zone that takes over automatically during failure.

What is the **main purpose** of a Read Replica in RDS?
A. High availability
B. Load balancing for write operations
C. Scaling read workloads and offloading analytics queries
D. Backup retention
✅ **Correct Answer: C.**
💡 **Explanation: Read Replicas** are **asynchronous copies** of the primary DB used to **offload read-heavy workloads** and **improve performance**.

🧩 **MCQ 3 — (Mid-Level)**

Which statement correctly differentiates **Multi-AZ** and **Read Replicas**?

A. Both use synchronous replication for high availability

B. Multi-AZ uses asynchronous replication; Read Replicas use synchronous replication

C. Multi-AZ uses synchronous replication for failover; Read Replicas use asynchronous replication for scaling

D. Both are used primarily for analytics

✅ **Correct Answer: C.**

💡 **Explanation:**

- **Multi-AZ → synchronous** (for **HA & DR**)
- **Read Replica → asynchronous** (for **read scaling & analytics**)

Which of the following statements about **Multi-AZ deployments** is TRUE?

A. They increase read throughput

B. They provide automatic failover with no data loss

C. They allow independent queries on standby

D. They require manual promotion during failover

✅ **Correct Answer: B.**

💡 **Explanation:** Multi-AZ uses **synchronous replication** for **zero data loss** and AWS handles **automatic failover** transparently via DNS switch.

When you create an **RDS Read Replica**, what kind of replication does AWS use?

A. Synchronous replication

B. Semi-synchronous replication

C. Asynchronous replication

D. Two-phase commit replication

✅ **Correct Answer: C. Asynchronous replication**

💡 **Explanation:** Read Replicas use **asynchronous replication**, so there may be slight lag between primary and replica.

What happens when the **primary instance in a Multi-AZ deployment** fails?

A. AWS deletes the instance and requires manual restart

B. AWS automatically promotes the standby to primary and updates DNS

C. Database becomes read-only until manually restored

D. Application must change endpoint to reconnect

✅ **Correct Answer: B.**

💡 **Explanation:** AWS performs **automatic failover** to the **standby** and updates the **same DNS endpoint**, minimizing downtime.