

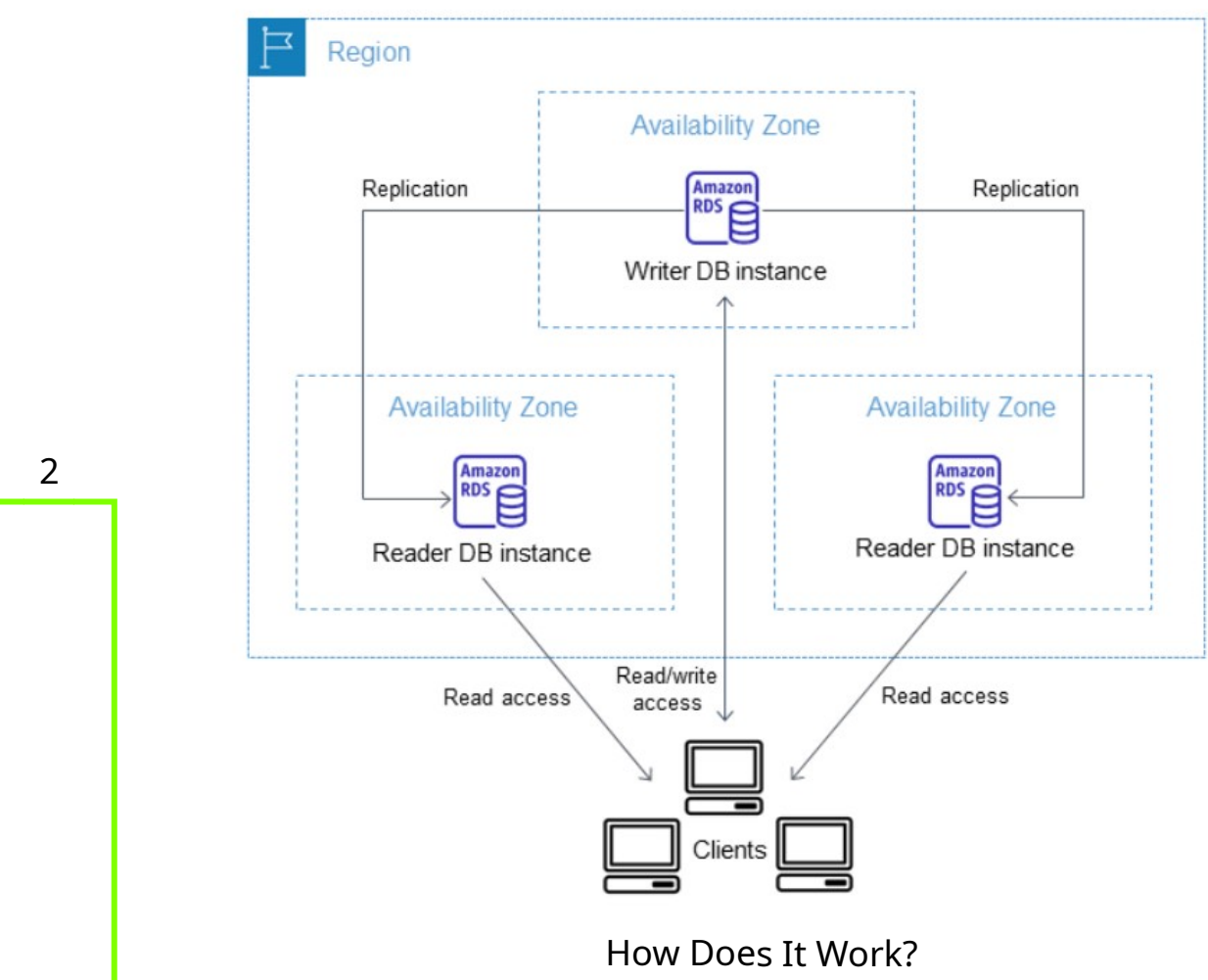
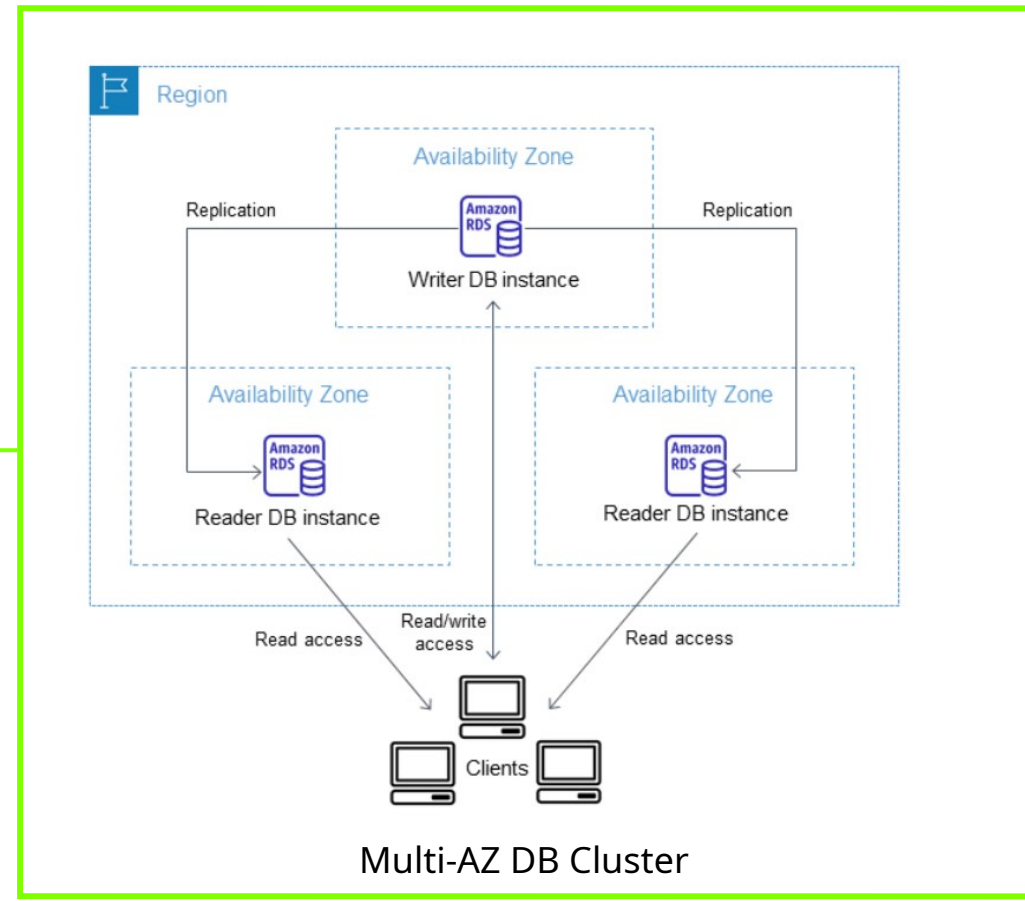


## 1 Single DB instance

- Introduction
  - 1 A Single DB Instance in AWS RDS refers to a standalone database instance running in a single Availability Zone without standby replicas or automatic failover.
  - 2 It's ideal for non-critical applications that don't require high availability.
  - 3 Lower cost compared to Multi-AZ deployments, as there are no standby instances or extra resources required.
- Drawbacks
  - 1 No Automatic Failover If the instance fails or the Availability Zone goes down, there is no automatic recovery, leading to downtime
  - 2 No High Availability Data is not replicated across Availability Zones, making it less suitable for mission-critical applications.
  - 3 Manual Backup Recovery In the case of a failure, you would need to restore from a backup manually, which takes time and effort.

## 2 Multi-AZ DB instance

- Introduction
  - 1 A Multi-AZ DB Instance in AWS RDS creates two database instances in two different Availability Zones (AZs) within the same region.
  - 2 This setup ensures high availability by keeping your application running smoothly if one instance or AZ experiences any issue.
- How Does It Work?
  - 1 AWS creates one main (primary) DB instance and a standby instance in a different AZ
  - 2 Data is always copied (replicated) from the primary to the standby instance in real-time
  - 3 If the primary instance faces any issue (like system failure or maintenance), AWS automatically switches (failover) to the standby instance without any manual effort.
  - 4 A new primary is available to serve your new workload in as quickly as 60 seconds
- Where Is It Used?
  - 1 Production Applications
  - 2 Critical Business Apps
  - 3 Uptime Guarantees
- Benefits of Multi-AZ Instances
  - 1 High Availability
  - 2 No Data Loss
  - 3 Uninterrupted Backups
  - 4 Automatic Failover
- Downsides of Multi-AZ Instances
  - 1 Higher Cost
  - 2 No Performance Gain
  - 3 Slight Latency
- Important Considerations
  - 1 Not for Scaling
  - 2 Region Limitation



- Deployment Structure A Multi-AZ DB Cluster in Amazon RDS includes
  - 1 One Writer DB Instance
    - 1 This is the main database where all the write operations (inserting, updating, deleting data) happen.
    - 2 It also handles read operations.
  - 2 Two Reader DB Instances
    - 1 These are replicas that only handle read operations.
    - 2 They are placed in different Availability Zones (AZs) to ensure high availability
  - 3 All three instances (one writer and two readers) are in different Availability Zones within the same region, which helps provide high availability and protection against failure.
- Replication Process
  - 1 When data is written to the writer instance, it is copied to the two reader instances.
  - 2 This copying process happens almost immediately, ensuring that all the instances have up-to-date data.
  - 3 Write operations are handled by the writer instance
  - 4 Read operations are shared among the writer and both reader instances, improving performance
- Failover Mechanism
  - 1 If the writer instance fails (due to an issue in one AZ), one of the reader instances is quickly promoted to be the new writer.
  - 2 This switch happens automatically, ensuring that the database stays available with very little downtime.
  - 3 Since the data is always being copied to the reader instances, there's no data loss when switching over.
- Handling Read Traffic
  - 1 Write operations Only the writer instance can handle these
  - 2 Read operations All three instances (writer and both readers) can handle these, helping to share the load and make the system faster.
- Lower Write Latency
  - 1 Compared to a Multi-AZ DB Instance deployment, Multi-AZ DB Clusters provide lower write latency.
  - 2 This is because the replication process between the writer and reader instances is more efficient.
  - 3 In a Multi-AZ DB Cluster, the writer instance does not need to wait for the readers to fully synchronize before moving on to the next task.
  - 4 This makes the system faster and more efficient for handling write operations, especially in high-traffic environments

## 3 Benefits of Multi-AZ Instances

- 1 High Availability
- 2 Increased Read Capacity
- 3 Lower Write Latency
- 4 Automatic Failover
- 5 Data Durability
- 6 Cost Efficiency for Read-Heavy Workloads

## 4 Downsides of a Multi-AZ DB Cluster

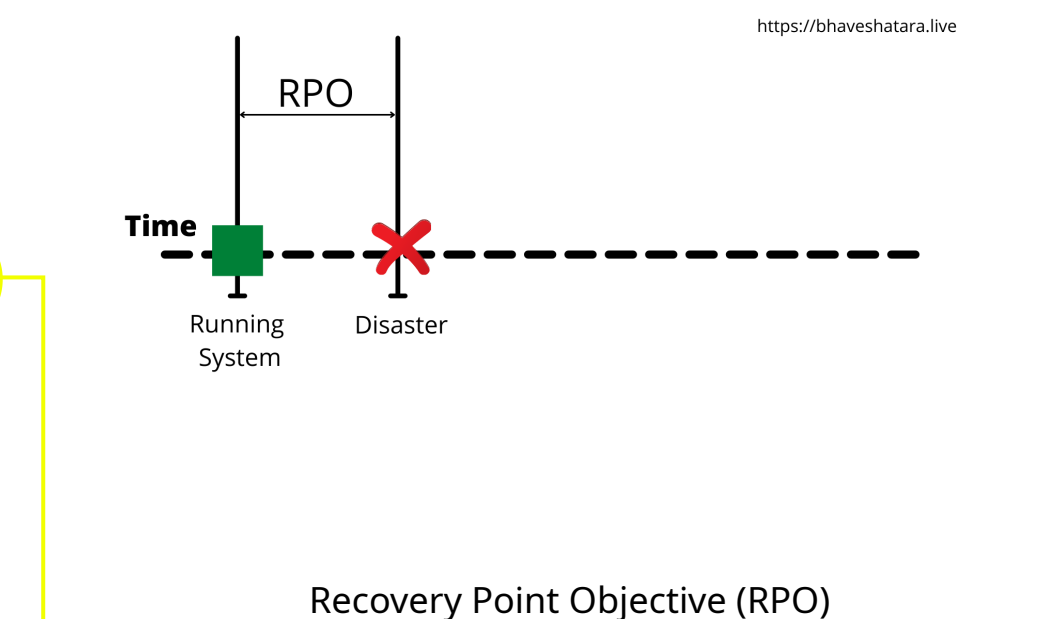
- 1 Higher Cost
- 2 Increased Complexity
- 3 Limited Availability of Reader Instances
- 4 No Cross-Region Disaster Recovery
- 5 Currently supported by the following database engines
  - 1 Amazon RDS for MySQL (starting with version 8.0.28 and higher)
  - 2 Amazon RDS for PostgreSQL (starting with version 13.4 and higher)

## 4 How to choose availability option

### 1 Introduction

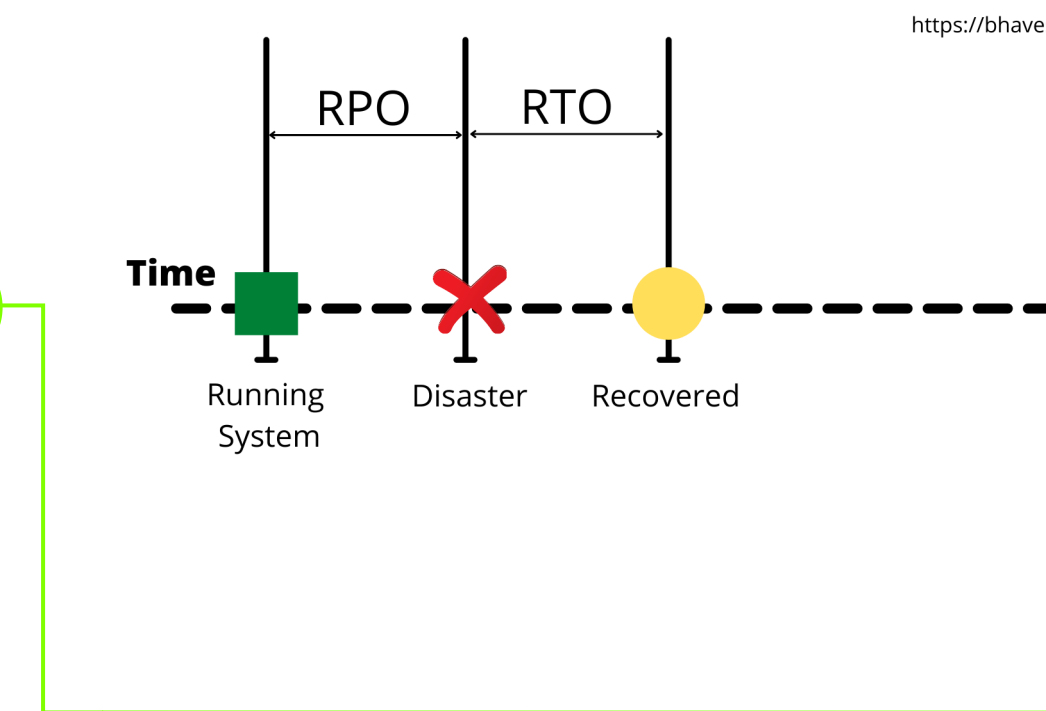
- 1 In AWS RDS, we have three main options for database high availability
  - 1 Single Host
  - 2 Multi-AZ
  - 3 Multi-AZ Cluster
- 2 Deciding which option is best for your application requires careful planning because there's a big difference in pricing.
- 3 To make the right choice, it's essential to understand two key factors
  - 1 Recovery Point Objective (RPO)
  - 2 Recovery Time Objective (RTO)
- 4 Which helps us understand how quickly our database can be restored and how much data loss is acceptable in the event of an outage

### 2 Recovery Point Objective (RPO)



- 1 Maximum amount of data you can afford to lose if something goes wrong, like a system crash.
- 2 The corporate backup strategy is based on your RPO because RPO defines how much data you can afford to lose
  - 1 If you took a backup at 10:00 AM and the database crashes at 10:30 AM
  - 2 You would lose 30 minutes of data because the last backup was at 10:00 AM
- 3 Example
  - 1 In this case, if your RPO is 1 hour, you're still within the tolerance for data loss, so you're fulfilling your RPO.
  - 2 However, if your RPO were 15 minutes, you wouldn't be meeting it, as you would have lost more data than allowed.
  - 3 In this case, you would need to adjust your backup strategy to take backups more frequently, such as every 15 minutes, to meet the 15-minute RPO.

### 3 Recovery Time Objective (RTO)



- 1 RTO (Recovery Time Objective) is the maximum time your system can be down after a problem
- 2 It tells you how quickly you need to restore the system before it starts affecting your business
  - 1 If your database crashes at 10:30 AM and your RTO is 1 hour, that means you need to have your database restored and running by 11:30 AM
- 3 Example
  - 2 If you manage to restore it within that hour, you are meeting your RTO.
  - 3 But if the recovery takes longer than 1 hour, you're not meeting the RTO, and you may need to adjust your recovery strategy to ensure faster recovery times in the future.

### 4 RPO And RTO Availability Options

- |                    |                             |          |                            |
|--------------------|-----------------------------|----------|----------------------------|
| 1 Single Host      | 1 RPO (Data Loss Tolerance) | High     | Rely on backups            |
|                    | 2 RTO (Recovery Time)       | High     |                            |
| 2 Multi-AZ         | 1 RPO (Data Loss Tolerance) | Low      | Use replication            |
|                    | 2 RTO (Recovery Time)       | Low      | Typically takes 60 seconds |
| 3 Multi-AZ Cluster | 1 RPO (Data Loss Tolerance) | Low      | Use replication            |
|                    | 2 RTO (Recovery Time)       | Very Low | Typically under 35 seconds |