Best practices for converting a Single-AZ Amazon RDS instance to a Multi-AZ instance

by Hanish Garg | on 14 JUN 2021 | in [Amazon RDS](https://aws.amazon.com/blogs/database/category/database/amazon-rds/) | [Permalink](https://aws.amazon.com/blogs/database/best-practices-for-converting-a-single-az-amazon-rds-instance-to-a-multi-az-instance/) | [Comments](https://commenting.awsblogs.com/embed.html?disqus_shortname=aws-database-blog&disqus_identifier=15255&disqus_title=Best+practices+for+converting+a+Single-AZ+Amazon+RDS+instance+to+a+Multi-AZ+instance&disqus_url=https://aws.amazon.com/blogs/database/best-practices-for-converting-a-single-az-amazon-rds-instance-to-a-multi-az-instance/) | [Share](https://aws.amazon.com/blogs/database/best-practices-for-converting-a-single-az-amazon-rds-instance-to-a-multi-az-instance/)

Many database users spend a considerable amount of time architecting their database for high availability. High availability is critical for business continuity. With the AWS Cloud, you can create highly available databases using [Amazon Relational Database Service](https://aws.amazon.com/rds/ha/) (Amazon RDS).

Amazon RDS is a fully managed database service that makes it easy to set up, operate, and scale relational databases in the cloud. You can set up Amazon RDS in a Single-AZ database (DB) instance or a Multi-AZ DB instance for high availability requirements.

In this post, I discuss best practices for converting a Single-AZ DB instance to a Multi-AZ DB instance for Amazon RDS for [MySQL](https://aws.amazon.com/rds/mysql/), [MariaDB](https://aws.amazon.com/rds/mariadb/), [PostgreSQL](https://aws.amazon.com/rds/postgresql/), and [Oracle](https://aws.amazon.com/rds/oracle/) databases, and how to reduce expected performance latencies from the conversion. [Amazon RDS for SQL Server](https://aws.amazon.com/rds/sqlserver/) and [Amazon Aurora](https://aws.amazon.com/rds/aurora) use a different technology stack to provide Multi-AZ capabilities.

**Amazon RDS architecture**

Each Amazon RDS instance runs on a DB instance backed by an [Amazon Elastic Block Store](https://aws.amazon.com/ebs/) (Amazon EBS) volume for storage. Amazon RDS takes daily snapshots of the database, which are stored durably in [Amazon Simple Storage Service](https://aws.amazon.com/s3/) (Amazon S3). In a Single-AZ setup, one RDS DB instance and one or more EBS storage volumes are deployed in one Availability Zone across data centers. In a Multi-AZ configuration, RDS DB instances and EBS storage volumes are deployed across multiple Availability Zones.

When you enable Multi-AZ, Amazon RDS maintains a redundant and consistent standby copy of your data using synchronous storage replication. Amazon RDS detects and automatically recovers from the most common failure scenarios for Multi-AZ deployments so that you can resume database operations as quickly as possible without administrative intervention. **Amazon RDS automatically performs a failover in the event of loss of availability in the primary Availability Zone, loss of network connectivity, compute unit failure, and storage failure**, as described in [Amazon RDS Multi-AZ Deployments](https://aws.amazon.com/rds/features/multi-az/). Having separate Availability Zones greatly reduces the likelihood that both copies are concurrently affected by most types of disturbances and therefore provides a highly available database architecture.

**Convert Single-AZ to Multi-AZ**

You can use the [AWS Management Console](https://console.aws.amazon.com/rds), [AWS Command Line Interface](https://docs.aws.amazon.com/cli/latest/reference/rds/create-db-instance.html) (AWS CLI), or [Amazon RDS API](https://docs.aws.amazon.com/AmazonRDS/latest/APIReference/API_CreateDBInstance.html) to create new Multi-AZ deployments or modify existing Single-AZ instances to become Multi-AZ deployments. To create a new Multi-AZ deployment using the console, choose the [Create a standby instance](https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/USER_CreateDBInstance.html) option for Multi-AZ deployment when launching a DB instance. To convert an existing Single-AZ DB instance to a Multi-AZ deployment, [modify your DB instance](https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/Overview.DBInstance.Modifying.html) via the console, AWS CLI, or Amazon RDS API. These changes are applied during the next maintenance window unless you choose to apply them immediately.

Afbeelding met tekst

Automatisch gegenereerde beschrijving

**Why latency sometimes elevates during conversion**

When converting a DB instance from Single-AZ to Multi-AZ, Amazon RDS creates a snapshot of the database volumes and restores these to new volumes in a different Availability Zone. Although the newly restored volumes are available almost immediately, they don’t reach their specified performance until the underlying storage blocks are copied from the snapshot.

Therefore, during the conversion from Single-AZ to Multi-AZ, you can experience elevated latency and performance impacts. This impact is a function of volume type, your workload, instance, and volume size, and can be significant and may impact large write-intensive DB instances during peak hour of operations. For more information, see [Modifying a DB instance to be a Multi-AZ deployment](https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/Concepts.MultiAZ.html#Concepts.MultiAZ.Migrating).

**How to minimize latency**

To minimize these latencies and improve performance, you can use a [read replica](https://aws.amazon.com/rds/features/read-replicas/) approach for Amazon RDS for MySQL, MariaDB, PostgreSQL, and Oracle Enterprise Edition databases ahead of time to prepare your environment. With this approach, underlying storage blocks are copied before they are accessed.

Amazon RDS Read Replicas make it easy to scale out beyond the capacity constraints of a single DB instance for read-heavy database workloads. You can create one or more replicas of a given source DB instance. In the following steps, I show how you can create a Multi-AZ read replica, synchronize it with your Single-AZ DB instance, and then promote it as your primary DB instance to minimize latencies during conversion.

1. On the Amazon RDS console, select your database.
2. On the Actions menu, choose Create read replica.
3. Use your production DB instance as the source.  
   Afbeelding met tekst

   Automatisch gegenereerde beschrijving
4. Select Create a standby instance.  
   Afbeelding met tekst

   Automatisch gegenereerde beschrijving  
   For full instructions, see [Creating a read replica](https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/USER_ReadRepl.html#USER_ReadRepl.Create).
5. When the new read replica is synchronized, you can promote it to be the primary DB instance, and then redirect your application to the new primary.  
   Plan your maintenance window for the cutover for a time after the new read replica is performing to expectations, by observing its [Amazon CloudWatch](http://aws.amazon.com/cloudwatch) write latency metric.Afbeelding met tekst

   Automatisch gegenereerde beschrijving
6. When promoting the newly created read replica, you can rename both the source DB instance and the new read replica to maintain a consistent database endpoint.
7. Enable automatic backups as needed.  
   Afbeelding met tekst

   Automatisch gegenereerde beschrijving

**Conclusion**

This post showed you the best practices of converting a Single-AZ RDS DB instance to a Multi-AZ RDS DB instance for MySQL, MariaDB, PostgreSQL, and Oracle databases without compromising on performance. With the read replica approach, your upstream applications don’t experience elevated latencies and performance degradations during the data hydration period.

Amazon RDS Multi-AZ configurations improve the availability and durability of customer data. For more information, see [Amazon RDS High Availability](https://aws.amazon.com/rds/ha/) and [High availability (Multi-AZ) for Amazon RDS](https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/Concepts.MultiAZ.html).

If you have any questions or comments about this post, please share them in the comments.

**About the author**

Afbeelding met persoon, person, kostuum, poseren

Automatisch gegenereerde beschrijvingHanish Garg is a Senior Solutions Architect for the US State, Local Government, and Education team at Amazon Web Services. He’s passionate about helping customers achieve their business objectives with database and serverless technologies.