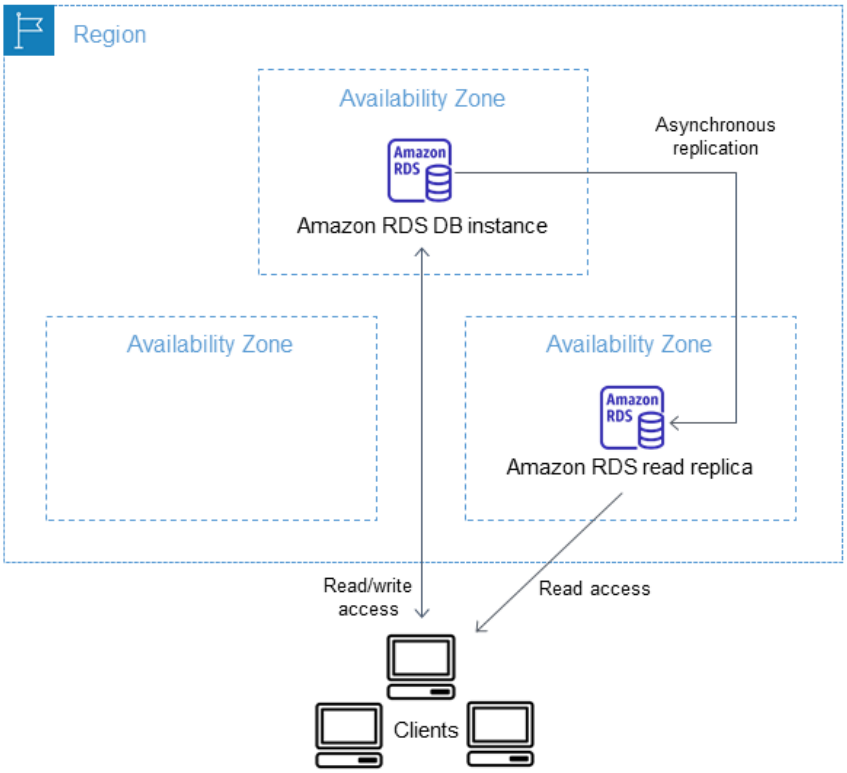
Working with DB instance read replicas

A *read replica* is a read-only copy of a DB instance. You can reduce the load on your primary DB instance by routing queries from your applications to the read replica. In this way, you can elastically scale out beyond the capacity constraints of a single DB instance for read-heavy database workloads.

To create a read replica from a source DB instance, Amazon RDS uses the built-in replication features of the DB engine

After you create a read replica from a source DB instance, the source becomes the primary DB instance. When you make updates to the primary DB instance, Amazon RDS copies them asynchronously to the read replica. The following diagram shows a source DB instance replicating to a read replica in a different Availability Zone (AZ). Client have read/write access to the primary DB instance and read-only access to the replica.



Use cases for read replicas

Deploying one or more read replicas for a given source DB instance might make sense in a variety of scenarios, including the following:

- Scaling beyond the compute or I/O capacity of a single DB instance for read-heavy database workloads. You can direct this excess read traffic to one or more read replicas.
- Serving read traffic while the source DB instance is unavailable. In some cases, your source DB instance might not be able to take I/O requests, for example due to I/O suspension for backups or scheduled maintenance. In these cases, you can direct read traffic to your read replicas. For this use case, keep in mind that the data on the read replica might be "stale" because the source DB instance is unavailable.
- Business reporting or data warehousing scenarios where you might want business reporting queries to run against a read replica, rather than your production DB instance.
- Implementing disaster recovery. You can promote a read replica to a standalone instance as a disaster recovery solution if the primary DB instance fails.

How read replicas work

When you create a read replica, you first specify an existing DB instance as the source. Then Amazon RDS takes a snapshot of the source instance and creates a read-only instance from the snapshot. Amazon RDS then uses the asynchronous replication method for the DB engine to update the read replica whenever there is a change to the primary DB instance.

The read replica operates as a DB instance that allows only read-only connections. An exception is the RDS for Oracle DB engine, which supports replica databases in mounted mode. A mounted replica doesn't accept user connections and so can't serve a read-only workload. The

primary use for mounted replicas is cross-Region disaster recovery. For more information, see Working with read replicas for Amazon RDS for Oracle.

Applications connect to a read replica just as they do to any DB instance. Amazon RDS replicates all databases from the source DB instance.

Promoting a read replica to be a standalone DB instance

You can promote a read replica into a standalone DB instance. When you promote a read replica, the DB instance is rebooted before it becomes available.

There are several reasons you might want to promote a read replica to a standalone DB instance:

- Performing DDL operations (MySQL and MariaDB only) – DDL operations, such as creating or rebuilding indexes, can take time and impose a significant performance penalty on your DB instance. You can perform these operations on a MySQL or MariaDB

read replica once the read replica is in sync with its primary DB instance. Then you can promote the read replica and direct your applications to use the promoted instance.

- Sharding – Sharding embodies the "share-nothing" architecture and essentially involves breaking a large database into several smaller databases. One common way to split a database is splitting tables that are not joined in the same query onto different hosts. Another method is duplicating a table across multiple hosts and then using a hashing algorithm to determine which host receives a given update. You can create read replicas corresponding to each of your shards (smaller databases) and promote them when you decide to convert them into standalone shards. You can then carve out the key space (if you are splitting rows) or distribution of tables for each of the shards depending on your requirements.
- Implementing failure recovery – You can use read replica promotion as a data recovery scheme if the primary DB instance fails. This approach complements synchronous replication, automatic failure detection, and failover.

  If you are aware of the ramifications and limitations of asynchronous replication and you still want to use read replica promotion for data recovery, you can. To do this, first create a read replica and then monitor the primary DB instance for failures. In the event of a failure, do the following:

  1. Promote the read replica.
  2. Direct database traffic to the promoted DB instance.
  3. Create a replacement read replica with the promoted DB instance as its source.

When you promote a read replica, the new DB instance that is created retains the option group and the parameter group of the former read replica. The promotion process can take several minutes or longer to complete, depending on the size of the read replica. After you promote the read replica to a new DB instance, it's just like any other DB instance. For example, you can create read replicas from the new DB instance and perform point-in-time restore operations. Because the promoted DB instance is no longer a read replica, you can't use it as a replication

target. If a source DB instance has several read replicas, promoting one of the read replicas to a DB instance has no effect on the other replicas.

Backup duration is a function of the number of changes to the database since the previous backup. If you plan to promote a read replica to a standalone instance, we recommend that you enable backups and complete at least one backup prior to promotion. In addition, you can't promote a read replica to a standalone instance when it has the `backing-up` status. If you have enabled backups on your read replica, configure the automated backup window so that daily backups don't interfere with read replica promotion.

The following steps show the general process for promoting a read replica to a DB instance:

1. Stop any transactions from being written to the primary DB instance, and then wait for all updates to be made to the read replica. Database updates occur on the read replica after they have occurred on the primary DB instance, and this replication lag can vary significantly. Use the `Replica Lag` metric to determine when all updates have been made to the read replica.
2. For MySQL and MariaDB only: If you need to make changes to the MySQL or MariaDB read replica, you must set the `read_only` parameter to `0` in the DB parameter group for the read replica. You can then perform all needed DDL operations, such as creating indexes, on the read replica. Actions taken on the read replica don't affect the performance of the primary DB instance.
3. Promote the read replica by using the Promote option on the Amazon RDS console, the AWS CLI command `promote-read-replica`, or the `PromoteReadReplica` Amazon RDS API operation.
4. **Note**
5. The promotion process takes a few minutes to complete. When you promote a read replica, replication is stopped and the read replica is rebooted. When the reboot is complete, the read replica is available as a new DB instance.

6. (Optional) Modify the new DB instance to be a Multi-AZ deployment. For more information, see Modifying an Amazon RDS DB instance and Configuring and managing a Multi-AZ deployment.

Console

To promote a read replica to a standalone DB instance

1. Sign in to the AWS Management Console and open the Amazon RDS console at https://console.aws.amazon.com/rds/.
2. In the Amazon RDS console, choose Databases.
   The Databases pane appears. Each read replica shows Replica in the Role column.
3. Choose the read replica that you want to promote.
4. For Actions, choose Promote.
5. On the Promote Read Replica page, enter the backup retention period and the backup window for the newly promoted DB instance.
6. When the settings are as you want them, choose Continue.
7. On the acknowledgment page, choose Promote Read Replica.

AWS CLI

To promote a read replica to a standalone DB instance, use the AWS CLI `promote-read-replica` command.

Example

For Linux, macOS, or Unix:

```
aws rds promote-read-replica \
```

```
    --db-instance-identifier myreadreplica
```

For Windows:

```
aws rds promote-read-replica ^
    --db-instance-identifier myreadreplica
```

RDS API

To promote a read replica to a standalone DB instance, call the Amazon RDS API `PromoteReadReplica` operation with the required parameter `DBInstanceIdentifier`.