

Read-only Routing vs Read-only Database in SQL Server Always On

In SQL Server Always On Availability Groups (AG), the concepts of **Read-only Routing** and **Read-only Database** are crucial for managing read-write and read-only workloads efficiently across the replicas in an AG. Here's a detailed explanation of both:

1. Read-only Database in SQL Server Always On AG

A **read-only database** refers to a replica in the Always On Availability Group (AG) that is specifically configured to only allow read-only access to the data. Typically, you would use secondary replicas for read-only operations, such as reporting, backup, or analytics, so that the primary replica is not overwhelmed with these operations.

Key Points:

- **Availability Group Configuration:** In an AG, you can have both primary and secondary replicas. By default, the primary replica is read-write, and the secondary replicas are read-only (unless they are configured to allow read-write).
- **Read-Only Mode:** Secondary replicas in an AG can be set to "read-only" mode, which means no write operations can be performed on them. This ensures that they are only used for querying data and not modifying it.
- **Usage:**
 - **Reporting:** Offload reporting workloads to the secondary replicas, reducing the load on the primary replica.
 - **Backups:** Take backups (e.g., full, differential, log) from secondary replicas to avoid impacting the primary replica's performance.
 - **Read-Only Operations:** Perform read-heavy operations such as analytical queries on secondary replicas.

Configuration:

You can configure a secondary replica to be read-only by using the **ALTER DATABASE** command with the **SET READ_ONLY** option, or it can be done automatically based on the AG settings when the replica is in synchronous or asynchronous mode.

```
ALTER DATABASE [YourDatabase] SET READ_ONLY;
```

2. Read-only Routing in SQL Server Always On AG

Read-only routing is a feature that allows the SQL Server to automatically redirect read-only queries from the application to a designated secondary replica in an Always On Availability Group (AG). This ensures that read-only workloads do not impact the primary replica, as they are routed to the secondary replicas that are specifically configured to handle read-only operations.

Key Points:

- **Routing Logic:** When a read-only query is submitted to the primary replica, the Always On AG routing mechanism automatically routes the query to a secondary replica that is configured to allow read-only queries.
- **Application Impact:** The application does not need to be aware of which replica is the primary or secondary. The read-only routing feature ensures that the correct replica is used for read-only workloads.
- **Configuration:** Read-only routing must be configured using the **Availability Group Listener** and the **Read-Only Routing List**. You need to define which replicas should handle read-only traffic.

Steps to Configure Read-only Routing:

1. **Create or Modify the AG Listener:** You need to set up an AG listener that can direct traffic to the appropriate replica.
2. **Configure Read-Only Routing:** Define which secondary replicas are available for read-only routing by modifying the AG settings.

Here's a sample of how to configure read-only routing in SQL Server:

-- Set read-only routing for the availability group

ALTER AVAILABILITY GROUP [YourAGName]

MODIFY REPLICA ON 'SecondaryReplicaName'

WITH (READ_ONLY_ROUTING_URL = 'TCP://SecondaryReplica:Port');

3. **Configure the Application:** The application needs to be aware of the AG listener and needs to specify that it should send read-only queries to the listener with the appropriate routing configuration.

For instance, in the connection string, you'd specify `ApplicationIntent=ReadOnly` to indicate that read-only queries should be routed to the secondary replica:

Server=AGListenerName;Database=YourDatabase;ApplicationIntent=ReadOnly;

Key Considerations:

- **Routing Order:** You can define the order in which the secondary replicas are chosen for routing the read-only queries. This is done by specifying a list of replicas in the availability group configuration.
- **Failover Impact:** When a failover occurs, the read-only routing configuration automatically adjusts so that read-only queries are directed to the new secondary replica.
-

Differences Between Read-only Database and Read-only Routing

Feature	Read-only Database	Read-only Routing
Definition	A replica that is set to read-only mode.	A routing mechanism that directs read-only queries to secondary replicas.
Primary Use	To offload read-only workloads (e.g., reporting) from the primary replica.	To automatically route read-only traffic to secondary replicas.
Configuration	Set the database on secondary replicas to read-only mode using <code>SET READ_ONLY</code> .	Configure read-only routing via Availability Group Listener and routing list.
Implementation	Database on secondary replica is set as read-only.	Connection string is set to direct read-only queries.
Impact on Failover	If the primary replica fails, the secondary replica may be promoted to primary, and the read-only databases will change accordingly.	After failover, read-only routing will point to the new secondary replica, ensuring minimal disruption.
Flexibility	No automatic redirection; applications need to manage connections manually.	Applications are automatically directed to the correct replica for read-only queries.

Summary:

Both **Read-only Databases** and **Read-only Routing** serve the purpose of offloading read-only workloads from the primary replica, but they are used differently:

- **Read-only Database** is a state of the replica where it is explicitly set to be read-only, typically for workloads like reporting, analytics, or backup operations.
- **Read-only Routing** is a dynamic feature that automatically directs read-only queries to the appropriate secondary replica, without requiring the application to manage which replica to use.

By using both features in combination, you can optimize your SQL Server Always On Availability Group to efficiently handle both read and write workloads across multiple replicas, ensuring high availability and performance.