https://www.sqldbachamps.com

Praveen Madupu - +91 98661 30093
Sr SQL Server DBA, Dubai
praveensqldba12@gmail.com

Performing a **Disaster Recovery Drill** for a **SQL Server Always On Availability Group (AG) with 3 replicas** is essential to validate the failover mechanism and ensure that all replicas are functioning correctly.

In a 3-replica setup, you typically have one **primary replica** and two **secondary replicas** (either synchronous or asynchronous).

This step-by-step guide details how to simulate a disaster recovery drill for Always On Availability Groups, testing the failover, and ensuring that the system is disaster-ready.

1. Overview of the Availability Group Setup

In a typical Always On AG with 3 replicas:

- **Primary Replica**: The read-write replica that handles all data modifications.
- **Secondary Replicas**: Can be either synchronous (with automatic failover) or asynchronous (manual failover) and can be used for read-only workloads, reporting, or backups.

The availability mode affects how data is synchronized between the replicas:

- **Synchronous Commit**: Ensures no data loss during a failover by waiting for acknowledgment from the secondary replica before committing a transaction.
- **Asynchronous Commit**: Allows for potential data loss because transactions on the primary are committed without waiting for acknowledgment from the secondary replica.

2. Pre-Drill Preparation

a. Ensure AG Replicas Are Healthy

Before initiating the DR drill, ensure all replicas are synchronized and healthy. Run the following queries to check the synchronization state:

1. **Check the synchronization state** of the AG replicas:

```
SELECT ag.name, ar.replica_server_name, ar.synchronization_state_desc, ar.role_desc

FROM sys.availability_groups ag

JOIN sys.dm_hadr_availability_replica_states ar

ON ag.group_id = ar.group_id;
```

Ensure all replicas are in a **SYNCHRONIZED** or **SYNCHRONIZING** state and that there are no issues with the availability group.

Praveen Madupu - +91 98661 30093
Sr SQL Server DBA, Dubai
praveensqldba12@gmail.com

2. **Check AG Failover Readiness**:

```
SELECT ag.name, ags.database_id, ags.synchronization_health_desc

FROM sys.dm_hadr_database_replica_cluster_states ags

JOIN sys.availability_groups ag

ON ag.group_id = ags.group_id;
```

1. The `synchronization_health_desc` should show **HEALTHY** for all databases involved in the AG.

b. Backup SQL Server Databases

Take a full backup of critical databases to ensure that data is safe in case of any issues during the drill.

1. **Perform a full backup**:

```
BACKUP DATABASE [YourDatabaseName] TO DISK = 'C:\Backup\YourDatabaseName_Full.bak';
```

2. **Backup the transaction logs** to ensure you have point-in-time recovery:

```
BACKUP LOG [YourDatabaseName] TO DISK = 'C:\Backup\YourDatabaseName_Log.trn';
```

c. Check Quorum Configuration

Ensure that the Windows Server Failover Cluster (WSFC) quorum is correctly configured. This ensures that the cluster can decide which replica should become the primary in the event of a failover.

- **Quorum Configuration** can be:
  - **Node Majority**: Majority of nodes must be online.
  - **Node and File Share Majority**: A file share witness helps in maintaining quorum.
  - **Node and Disk Majority**: A disk witness helps maintain quorum.

Use the **Failover Cluster Manager** to verify that the quorum is configured correctly and that the cluster is healthy.

d. Notification and Maintenance Window

Inform stakeholders and schedule the drill during a maintenance window to minimize disruptions.

Praveen Madupu - +91 98661 30093
Sr SQL Server DBA, Dubai
praveensqldba12@gmail.com

3. Steps for the Disaster Recovery Drill

Step 1: Stop User Activity on the Primary Replica

To avoid data inconsistencies during failover, stop user activity on the primary replica. You can:

- Notify application teams to stop processing or redirect them to another system.
- Use **SQL Server Management Studio (SSMS)** to set databases in the AG to **single-user mode**:

```
ALTER DATABASE [YourDatabaseName] SET SINGLE_USER WITH ROLLBACK IMMEDIATE;
```

Step 2: Simulate Failover Scenarios

You can simulate two types of failover scenarios:

Planned Manual Failover (for testing purposes) and **Forced Failover** (for disaster scenarios where the primary is unreachable).

a. Planned Manual Failover (No Data Loss)

This is performed when all replicas are in **synchronous commit mode** and the primary is accessible. It ensures no data loss as all changes are synchronized to the secondary replica.

1. **Initiate the failover** on the current primary using **SSMS** or **T-SQL**. In **SSMS**, right-click the Availability Group under **Always On High Availability** > **Failover**.
2. Alternatively, use **T-SQL** to perform a manual failover:

```
ALTER AVAILABILITY GROUP [YourAGName] FAILOVER;
```

This command switches the role of the **primary** to one of the **synchronous secondary replicas**.

3. Verify Failover Success: Check the role of the new primary replica by running:

```
SELECT replica_server_name, role_desc

FROM sys.dm_hadr_availability_replica_states

WHERE is_local = 1;
```

The **role_desc** should be **PRIMARY** for the new primary node.

Praveen Madupu - +91 98661 30093
Sr SQL Server DBA, Dubai
praveensqldba12@gmail.com

b. Forced Failover with Possible Data Loss (Disaster Scenario)

A **forced failover** is used when the primary replica is unavailable (disconnected or failed) and you need to bring the secondary replica online. This can result in data loss if the secondary is in **asynchronous commit mode**.

1. **Use T-SQL** to force the failover on the secondary replica:

```
ALTER AVAILABILITY GROUP [YourAGName] FORCE_FAILOVER_ALLOW_DATA_LOSS;
```

This command promotes the secondary replica to **primary**, even if it isn't fully synchronized. Be aware that this can cause some data loss.

Verify the failover on the secondary (now primary):

```
SELECT replica_server_name, role_desc

FROM sys.dm_hadr_availability_replica_states

WHERE is_local = 1;
```

The role should now show **PRIMARY**.

Step 3: Validate the Failover and Replica Synchronization

Once the failover is complete, validate the state of the new primary and secondary replicas:

1. **Check Database Synchronization** on the new primary replica:

```
SELECT db_name(database_id), synchronization_state_desc

FROM sys.dm_hadr_database_replica_states;
```

The **synchronization_state_desc** should show **SYNCHRONIZED** for databases if the new secondary replicas are in synchronous commit mode.

2**. Run Application Tests** to ensure that the application is connected to the new primary replica and that operations (reads, writes, updates) work as expected.

Praveen Madupu - +91 98661 30093
Sr SQL Server DBA, Dubai
praveensqldba12@gmail.com

4. Validation After Failover

a. Check Data Consistency

Run queries on the new primary to ensure that all critical tables are consistent and that data is accurate.

- Validate row counts and key data to ensure that nothing is missing after the failover.

```
SELECT COUNT(*) FROM YourCriticalTable;
```

b. Check SQL Server Jobs

Ensure that SQL Server Agent jobs are running properly on the new primary replica. After a failover, jobs that were scheduled to run on the old primary should now run on the new primary.

1. **Open SQL Server Agent** on the new primary replica.
2. Verify that all critical jobs are running as expected.

c. Verify AG Listener and Application Connectivity

The **AG Listener** ensures that applications can connect to the new primary replica automatically, without requiring connection string changes.

1. Test the application's connectivity to the new primary replica.
2. Check that the AG Listener's virtual IP has moved and the applications are pointing to the correct primary.

d. Run Database Integrity Checks

Run `DBCC CHECKDB` to verify the integrity of the databases after failover:

```
DBCC CHECKDB('YourDatabaseName');
```

This ensures that there are no corruption issues.

Praveen Madupu - +91 98661 30093
Sr SQL Server DBA, Dubai
praveensqldba12@gmail.com

5. Post-Drill Failback or Keep Current Setup

a. Failback to Original Primary Replica

If you need to return the primary role to the original primary replica, perform another failover back to the original primary once it is available:

1. **Perform a manual failover** back to the original primary using:

```
ALTER AVAILABILITY GROUP [YourAGName] FAILOVER;
```

2. **Verify the role switch** by checking the availability group state again.

b. Keep the Current Primary Replica

If you choose to keep the new primary replica as the permanent primary, no further action is needed. Ensure that the former primary replica (now a secondary) is synchronized.

6. Post-Drill Monitoring and Documentation

a. Monitor Availability Group Health

After the drill, continuously monitor the health of the availability group to ensure everything is functioning as expected.

1. **Monitor synchronization** status and failover readiness using:

```
SELECT ag.name, ar.replica_server_name, ar.synchronization_health_desc

FROM sys.availability_groups ag

JOIN sys.dm_hadr_availability_replica_states ar

ON ag.group_id = ar.group_id;
```

2.Check **WSFC quorum health** using **Failover Cluster Manager**.

b. Document the Drill

- Record the actions taken during the drill, including:
  - Time to failover.
  - Any issues encountered.
  - Results of validation checks.
  - Recommendations for improvement.
- This documentation will help improve future disaster recovery processes and ensure that the team knows how to handle an actual disaster.

Praveen Madupu - +91 98661 30093
Sr SQL Server DBA, Dubai
praveensqldba12@gmail.com

7. Key Points to Remember

- **Quorum and WSFC Health**: Ensure that the Windows Failover Cluster and quorum configuration are healthy before the drill.
- **Failover Types**: Planned manual failover is preferable during drills; forced failover is used for true disaster scenarios.
- **Application Connectivity**: The AG Listener should automatically redirect applications to the new primary replica.
- **Synchronization State**: Always check the synchronization state before and after the failover to prevent data loss.
- **Backup and Validation**: Always back up critical databases before starting the drill and validate data consistency after the failover.

By following this detailed process for performing an Always On Availability Group DR drill, you can ensure that your SQL Server environment is resilient, and you are prepared to handle real-world disaster recovery scenarios with minimal data loss and downtime.