

A **SQL Server Failover Cluster (Active-Passive) Disaster Recovery (DR) Drill** simulates a failover scenario in an SQL Server environment where high availability (HA) is configured using a failover cluster.

In an **Active-Passive Failover Cluster**, one node (the **active node**) runs the SQL Server instance, and the other node (the **passive node**) is on standby, ready to take over when the active node fails or is intentionally switched.

Performing a DR drill helps ensure that the failover process works smoothly, the passive node can take over when needed, and applications can reconnect seamlessly with minimal downtime.

This step-by-step guide provides details on how to conduct a DR drill for an **Active-Passive SQL Server Failover Cluster**.

## 1. Overview of Failover Cluster Components

- **Active Node:** The server currently hosting the SQL Server instance and accepting client connections.
- **Passive Node:** The server on standby, not actively handling the SQL Server instance but ready to take over if the active node fails.
- **Quorum:** The shared storage (disk witness or file share) or vote count that helps the cluster determine which node should control the resources.
- **Clustered Resources:** SQL Server and other associated services (e.g., SQL Server Agent, SQL Server Analysis Services) that are managed by the cluster.

## 2. Pre-Drill Preparation

### a. Check Cluster Health

Before starting the DR drill, ensure that the failover cluster is healthy and the nodes are communicating properly. This can be done using the **Failover Cluster Manager**.

1. **Open Failover Cluster Manager** on the active node.
2. In the **Summary of Cluster Nodes**, ensure all nodes are **online**.
3. Verify that the **SQL Server clustered instance** is currently running on the active node by checking the status of the SQL Server resource.

### b. Check the Quorum Configuration

Ensure that the quorum configuration is correct and the cluster can survive node failures. The quorum can be:

- **Node Majority:** A majority of nodes are online.
- **Node and Disk Majority:** A shared disk (witness) helps maintain quorum.
- **Node and File Share Majority:** A file share witness helps maintain quorum.

Check this in the **Failover Cluster Manager**.

### c. Backup SQL Server Data

Before performing the drill, ensure that a full backup of critical databases has been completed. This is a safeguard to ensure that data is protected during the drill.

1. **Full Backup** of all active databases:

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

2. **Transaction Log Backup** for additional protection:

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

### d. Schedule a Maintenance Window

The DR drill should be performed during a planned maintenance window, as failover involves a brief downtime during which the SQL Server instance is unavailable.

## 3. Steps for the Failover Cluster DR Drill

### Step 1: Prepare the Active Node for Failover

1. **Stop New Connections:** Prevent new connections to the SQL Server instance running on the active node to ensure a clean failover. You can either:
  - Set the databases on the active node to **single-user mode** (optional, but recommended for production scenarios), or
  - Simply inform users and applications to stop accessing the database temporarily.
2. **Terminate Active Sessions:** Identify and terminate any active user sessions to ensure that no in-progress transactions are disrupted. You can do this using **SQL Server Management Studio (SSMS)**:

```
KILL [session_id];
```

- Use the `sys.dm_exec_sessions` view to identify session IDs (SPIDs) to kill.

### Step 2: Initiate Failover

Failover the SQL Server instance from the active node to the passive node.

This can be done in two ways: using **Failover Cluster Manager** or **PowerShell**.

### Option 1: Failover Using Failover Cluster Manager

1. **Open Failover Cluster Manager** on the active node.
2. Navigate to **Roles** and locate the SQL Server resource.
3. Right-click the SQL Server role and select **Move > Select Node**.
4. Choose the passive node (the standby node) to move the SQL Server instance.
5. The **Failover Cluster Manager** will begin moving the clustered SQL Server instance and resources (IP address, storage, etc.) to the passive node.

### Option 2: Failover Using PowerShell

Alternatively, you can fail over using PowerShell:

```
Move-ClusterGroup -Name "SQL Server (INSTANCENAME)" -Node "Node2"
```

This command will trigger the failover to the specified node (Node2).

### Step 3: Verify Failover to the Passive Node

After initiating the failover, verify that the SQL Server instance and all associated resources have successfully moved to the passive node.

1. In **Failover Cluster Manager**, check the status of the SQL Server role:
  - The SQL Server instance should now be **online** on the passive node.
  - All resources (SQL Server, storage, IP addresses) should be in the **Running** state.
2. **Verify Database Availability:**
  - Connect to the SQL Server instance on the passive node using SSMS.
  - Run some basic queries to ensure the databases are online and functional.

```
SELECT name, state_desc
```

```
FROM sys.databases;
```

1. The database state should be **ONLINE** for all databases.

## 4. Validation After Failover

### a. Validate Application Connectivity

Ensure that the application can connect to the SQL Server instance on the new (passive) node. If your application uses a **Virtual Network Name (VNN)** or **Virtual IP (VIP)**, the connection should automatically redirect to the new active node without changing the connection string.

1. Test the application and ensure it can perform basic operations (insert, update, select, etc.).
2. Monitor application logs for any connection errors or performance issues.

### b. Check SQL Server Jobs and Services

Ensure that SQL Server Agent jobs, scheduled tasks, and other services (e.g., SSIS, SSRS) are running correctly after the failover. Check that jobs are running as expected and that there are no errors in job histories.

### c. Run Database Integrity Check

Check the integrity of the databases after the failover using **DBCC CHECKDB**:

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

Ensure there are no corruption issues or database errors.

### d. Monitor System Performance

Monitor the system's performance on the new active node:

- Check CPU, memory, and disk I/O usage.
- Look for any issues such as resource bottlenecks or performance degradation.

## 5. Post-Drill: Failback to the Original Active Node

Once the failover has been successfully validated, you can either choose to:

- **Keep the passive node** as the active node (if failover was intentional), or
- **Failback to the original active node** to restore the cluster to its pre-drill configuration.

### Option 1: Failback to the Original Node

To failback, repeat the same process as the initial failover:

1. **Open Failover Cluster Manager.**
2. Right-click the SQL Server role, select **Move > Select Node**, and choose the original active node.
3. Monitor the progress in **Failover Cluster Manager** to ensure the role is moved back successfully.

## Option 2: Keep the Current Node as Active

If you choose to keep the current node as the active node, no further action is required. Just ensure that all services are running smoothly.

## 6. Monitor and Document the DR Drill

### a. Monitor Cluster Health

Continue to monitor the cluster's health after the failover or failback to ensure all nodes are online, the quorum is maintained, and there are no issues with resources or services.

- Use **Failover Cluster Manager** to check the health and status of cluster resources.
- Monitor SQL Server logs and the **Windows Event Viewer** for any warning or error messages.

### b. Document the Drill Process

Document the entire DR drill, including:

- Steps taken during the drill.
- Time taken for failover and failback.
- Any issues encountered and how they were resolved.
- Observations and recommendations for future drills.

This documentation can help improve the DR drill process and serve as a guide in the event of an actual disaster.

## 7. Key Points to Remember

- **Planned Failover:** A DR drill involves a planned failover, which means stopping user activity on the SQL Server instance and failing over in a controlled manner.
- **Cluster Health:** Ensure both nodes and the cluster quorum are healthy before initiating the failover.
- **Failover Process:** The failover moves all SQL Server resources (databases, storage, network IP) from the active node to the passive node.
- **Application Connectivity:** Applications should automatically reconnect to the new active node if they use a virtual network name (VNN) or virtual IP (VIP).
- **Failback:** After validation, you can either keep the current node as active or fail back to the original node.

## Summary of the Process

1. **Prepare the active node** by stopping new connections and backing up databases.
2. **Initiate failover** to the passive node using **Failover Cluster Manager** or **PowerShell**.
3. **Validate the failover** by ensuring the databases and applications work on the new active node.
4. **Decide to failback** or keep the current setup.
5. **Monitor** the cluster and document the entire DR drill for future reference.

By regularly performing this drill, you can ensure that your SQL Server Failover Cluster is ready to handle real-world failovers and provide continuous availability.