SQL Server Recovery Procedure

Department: Database Administration

Purpose

This procedure outlines the details of the steps required to recover from a critical SQL Server failure. It provides instructions for restoring database functionality on either the primary server after repair or backing up the SQL server.

Scope

This procedure applies to the primary IT department SQL server (SQL-IT), hosting critical business databases. It covers the hardware aspect and the database corruption aspects of the server, requiring immediate restoration of service.

Prerequisites

- Administrator access credentials for SQL Server.
- Access to the secure password vault for service accounts.
- Knowledge of database backup and locations.
- Access to the data center.
- Basic understanding of SQL Server management tools.

Emergency Contacts

ROLE	NAME	PRIMARY CONTACT	SECONDARY CONTACT	Email
Database Manager	Shannon Kurtis	(556) 871-2396	(555) 659-8956	skurtis@company.com
Database Administrator	Rotation	(555) 432-8965	N/A	dba@company.com
Infrastructure Lead	Kenny Mayor	(555) 659-8976	(555) 465-7894	kmayor@company.com
Microsoft SQL Support	N/A	(800) 642-7676	N/A	N/A
Storage Team	Jonathan Davis	(555) 986-3569	(555) 325-7456	jdavis@company.com
Network Team	Maria Luiza	(555) 735-9164	N/A	mluiza@company.com

---

Procedure Steps

Phase 1: Initial Assessment and Notification

1. **Verify SQL Server Failure**
   - **1.1.** Attempt to connect to SQL-IT using SQL Server Management Studio (SSMS).
   - **1.2.** Check SQL service status in Windows Services (services.msc).
   - **1.3.** Review event logs for failure (Event Viewer → Application and System logs).
   - **1.4.** Document errors, time of occurrence, and affected databases.
   - **1.5.** Verify if the issue is related to hardware or software.
2. **Initiate Notification Protocol**
   - **2.1.** Call Database Manager (contact information above).
   - **2.2.** Sent email alert to database administrator (email information above) with subject "URGENT: SQL-IT FAILURE".
   - **2.3.** Create an incident ticket in ServiceNow using the "Database Critical Incident" template.
   - **2.4.** Update status in the IT dashboard about the incident with investigation ongoing and details.

Phase 2: Assess Recovery Options

3. **Determine the Appropriate Recovery Method**
   - **3.1.** If a hardware failure, like the server will not reboot, disk array failure, etc. and then proceed with failover to the backup SQL server (Phase 3 → Step 6).
   - **3.2.** If SQL Server service issues like the service will not start, database corruption, etc., then attempt quick service recovery (Phase 2 → Step 4).
   - **3.3.** If data corruption occurs, then attempt to restore the database in place (Phase 2 → Step 5).
4. **Attempt Quick Service Recovery**
   - **4.1.** Connect to SQL-IT via Remote Desktop: rdp://10.2.3.22.
   - **4.2.** Log in with administrator credentials secured from the secure password vault.
   - **4.3.** Open Service Console (services.msc).
   - **4.4.** Right click in "SQL SERVER (MSSQLSERVER)" and select "Restart".
   - **4.5.** Wait for 2 or 3 minutes for the service to restart.
   - **4.6.** Attempt to connect using SSMS to verify the recovery.
   - **4.7.** If successful, jump to step 12.
   - **4.8.** If unsuccessful, jump to step 5 or 6 according to assessment.
5. **Attempt to Restore the Database in Place**
   - **5.1.** Open SSMS and connect to SQL-IT.
   - **5.2.** Select the affected database and select (Tasks → Restore → Database).
   - **5.3.** Browse for the most recent backup at:
     - **5.3.1.** Network: \backup\SQLBackups\
     - **5.3.2.** Local: D:\SQLBackups\

5.4. Select the most recent backup (.bak) file and click "OK" to start the restore process.
5.5. If successful, jump to step 12.
5.6. If unsuccessful, jump to step 6.

Phase 3: Failover to Backup Server

6. **Verify Backup Server Status**
   6.1. Connect to SQL-IT-BU (10.2.3.23) via RDP.
   6.2. Log in with administrator credentials from the secure password vault.
   6.3. Verify the SQL Server is running (services.msc).
   6.4. Run preliminary health check with command:
   sqlcmd -S localhost -Q "SELECT @@VERSION; SELECT GETDATE ();".
   6.5. Verify you receive the current SQL server version and datetime response.
7. **Prepare the Backup Server for Failover**
   7.1. Open SSMS.
   7.2. Enable all required services (SQL Server, SQL Agent, etc.).
   7.3. Verify network protocols (TCP/IP) are enabled.
   7.4. Check that the SQL server instance is accepting connections:
   sqlcmd -S localhost -Q "SELECT @@SERVERNAME;"
8. **Restore the Latest Database Backups**
   8.1. Open SSMS and connect to SQL-IT-BU.
   8.2. Select the "Databases" folder and select "Restore Database".
   8.3. Select the appropriate source and browse to the backup location.
   8.4. Select the most recent full backup for the required databases.
   8.5. Restore transaction logs if available.
   8.6. Wait for all restore operations to be complete (15-30 minutes, depending on size).
9. **Verify Database Integrity**
   9.1. For each restored database, run integrity checks:
   DBCC CHECKDB('DatabaseName') WITH NO_INFOMSGS
   9.2. Verify clean results with no errors from each database.
   9.3. Run sample queries on each database to confirm operations.
   9.4. Verify that each database returns expected results for the queries.

Phase 4: Update Applications Connections

10. **Update Application Connection Strings**
    10.1. Log in to the Application Config Management Protocol:
    https://appconfig.company.com
    10.2. Navigate to "Database Connections".
    10.3. Select "Active SQL Failover Plan".
    10.4. Select "Yes" when prompted to redirect all applications to SQL-IT-BU.
    10.5. Wait for the confirmation message (2-3 minutes).

11. Verify Application Connectivity

11.1. Monitor application logs:

11.1.1. Log location: *\logserver\applogs*

11.1.2. Check for "Database connection successful" messages.

11.2. Test the critical applications functionality:

11.2.1. Employee portal:

https://employee.company.com

11.2.2. Inventory system:

https://inventory.company.com

11.3. For each application, perform basic operations that need database access.

Phase 5: Documentation and Communication

12. Update Stakeholders

12.1. Update the ServiceNow incident ticket with the status.

12.2. Send email to stakeholder distribution email at IT-critical-apps@company.com with status and expected performance impacts.

12.3. Update status in the IT dashboard with service restored.

13. Document of the Incident

13.1. Complete the incident response form in ServiceNow.

13.2. Document all steps taken during recovery.

13.3. Record timestamps for each step.

13.4. Note issues encountered during the recovery process.

13.5. Create follow-up tasks for future permanent fixes.

Phase 6: Post-Recovery Actions

14. Coordinate Permanent Repairs

14.1. If the primary server requires hardware replacement:

14.1.1. Notify IT procurement of replacement parts.

14.1.2. Schedule maintenance window.

14.1.3. Document all specifications for exact replacements.

14.2. If the primary server faced software issues:

14.2.1. Schedule SQL Server patching.

14.2.2. Document-specific errors for the case.

15. Plan Return to Primary Server

15.1. Once the primary server is repaired, schedule a switchback window.

15.2. Create a plan for synchronization between the backup server and the primary server.

15.3. Notify all stakeholders of the planned switchback window.

15.4. Follow the "Return to Primary SQL Server" procedure document.

Definitions

- **SQL-IT:** Primary IT department SQL Server.
- **SQL-IT-BU:** Backup to SQL Server.
- **SSMS:** SQL Server Management Studio, the primary management tool for SQL Server.
- **DBCC CHECKDB:** Database consistency checker command that verifies data integrity.