# Reclustering and Backup Script

PURPOSE:

The Ed Fi Dashboard is designed to be re-created periodically, nightly. This means the Dashboard database must, at times, be removed from the Production SQL Server Cluster (named EDFISQLPROD), **swapped with a newer version of the database, and re-added to the cluster**.

Because the cluster relies on two separate SQL servers, in different subnets, this nightly transition of dashboard databases has a risk of failure if something significant has changed in the environment. For this reason, it is important that Database Administrators and Network Administrators have access to this **document that explains**:

* How the script operates
* If the script is not performing some intended function, how to determine and fix the unsafe condition that is causing it to abort that function
* If the cluster is offline due to a transient error due to network failure during the scripts operation, how to bring cluster online again (**the cluster is self-healing for almost all issues**)
* General understanding of SQL Server logs and recovery and related troubleshooting techniques

NOTE: With a slight change of parameters inside the script – the database name at the beginning of the script – this script **can be used to replace any database in the** EDFISQLPROD **cluster**. The following values are hard-coded in the script and would need to be changed if they are ever modified:

HARD-CODED VALUES in the script:

|  |  |
| --- | --- |
| Cluster Name | EDFISQLPROD |
| SQL Availability Group Name | EdFiProdAvGroup |
| Cluster SQL Node Names | EC2AMAZ-5QGRUAT, EC2AMAZ-OGSJCCJ |
| Cluster SQL Node IP Addresses | 10.222.105.4, 10.222.106.70 |
| Default database file locations | D:\MSSQL\Data, D:\MSSQL\Log, D:\MSSQL\Backup |
| User names and Passwords | These are **pulled dynamically from the AWS Secrets Manager service** so that they will not be exposed in source code control. |
| This scripts log file location | D:\MSSQL\SQLAgentJobScripts\AutomatedBackup.log |

## Scripts Log File

All operations performed by this script are added to a human-readable log file with time stamps to assist in maintenance and troubleshooting. The location of this log file is configurable, and can be changed at will, and defaults to D:\AutomatedBackup.log. Any errors caused by logging failures are ignored by the script. It is wise to view this log periodically to ensure that dashboard-database-replacement and automated backups are being carried out as planned. The automated backups taken here do not preclude Minneapolis scheduling its own backups if desired.

This script can be run from any node (server) on the cluster, but should only be scheduled to run on one – not more than one server – otherwise it would repeat the procedure twice.

## Normal Script Operation

* Determine which cluster node is acting as the SQL Server High Availability Group Primary Node, and which is Secondary Node
* Determine if all Windows Failover Cluster Nodes (3 nodes, two of which are SQL Servers) are up and running, **if not, log the issue and abort script**
* Determine if the old and new Dashboard Databases are both present and online on the primary cluster node, **if not, log the issue and abort the script**
* Establish database connections to both SQL Servers, as a system administrator
  + This high access level is required due to:
    - High Availability Group Cluster operations
    - Take databases online/offline
    - Perform Backup, Restore, and Recovery commands
    - and other system-level manipulation
* Assigns/re-assigns network drive names to (as the local machine user “webserver”) on both cluster nodes’ D:\MSSQL folder.
  + This access is required to:
    - Rename SQL Server physical files
    - Transfer backup files between cluster nodes (server) using the SMB network protocol (TCP port 445)
  + IMPORTANT: On both servers, the user “webserver” must have the following file system access:
    - Security access (read and write permissions) to the directory D:\MSSQL and its subdirectories (recall that granting a user permission to a directory in the Windows operating system recursively grants permission to all subdirectories and files)
    - The folder D:\MSSQL must be shared as “MSSQL” to local machine user “webserver” with read and write permissions.
* Note: The newly created Dashboard database (**EdFi\_Dashboard\_temp**), which will replace the current (old, yesterday’s) Dashboard database (EdFi\_Dashboard).
* The old database is removed from the High Availability Group on the primary node, and this change is communicated automatically to the secondary database server node.
* The old database is dropped from both servers. At this point, any operations on the Dashboard web application will fail until the database is restored – it will take a couple of minutes to make the transition.
* The new database is renamed from EdFd\_Dashboard\_temp to EdFi\_Dashboard.
* A backup is taken of the new Dashboard database and copies it to the secondary database server node.
* The secondary node restores the new database with REPLACE (replaces any leftovers of the existing database) and NORECOVERY (set to the ‘**recovering**’ state, and cannot yet be used).
* The primary node adds the new database to the High Availabiliity Group (to the Failover Cluster).
* This action causes the primary and secondary nodes to display “**synchronizing**” next to the database in SQL Server Management Studio’s Object Explorer pane.
* The Dashboard Web app is now operational again.

Next, the Backup Script will run. It simply takes a backup of all databases on the server and saves them to the private, encrypted, AWS bucket Minneapolis S3 bucket named **edfi-mpls-database-backups**.

## Troubleshooting

In normal operation, no databases should be marked with the label “**recovering**” in SQL Server Management Studio’s Object Explorer pane. Any databases that do have a label should say “**synchronized**”, meaning that any changes on one of the database server nodes are transferred to the other node by database mirroring (“Always On High Availability Groups”).

If this is not the case, recovery or troubleshooting may be required. Here are the things one can check to isolate the issue:

* Is the Cluster operational?
  + Open the application “Failover Cluster Manager” on either of the database servers.
  + Connect to the cluster on the current machine.
  + Expand to the “Nodes” section. Are all three servers online?
  + Expand to the “Roles” section.
  + Is the EdFiAvGroup Role online?
    - If not, right-click it and choose “Bring Online”. If this fails, view the reason in the “Information Details” section.
    - If so, in the bottom details pane, click the “Dependencies” tab. Expand the roles dependencies to ensure that one of the two subnets (105 or 106) is online. You may attempt the “Move” action to manually fail over the cluster to the secondary node. An hourglass should appear for 1-2 minutes, then the secondary node should come online and display a green up-pointing arrow instead of the red down-pointing arrow, and the Role should indicate “online”. If not, check the Information Details section for a reason.
  + Generally, if the Availability Group Cluster Role is offline, it means one of the SQL Servers background service is malfunctioning. Attempt to restart the database servers.
* Ensure network TCP port 5022 (database mirroring) is open between the database servers and their subnets. Also ensure ports 445 (SMB for DB Backup transfers) and 1433 (database cluster listener process) are open.
* Ensure all ICMP protocol traffic is enabled between the firewalls. SQL Servers use “ping” functionality to ensure that other members of the database cluster are operational.
* In SQL Server Management Studio Object Explorer pane shows a database is “offline”, attempt the T-SQL command ALTER DATABASE database-name SET ONLINE
* In SQL Server try re-adding the database to the availability group:
  + Open SQL Server Management Studio.
  + In the Object Explorer, Expand the Always On Availability Groups node >> EdFiAvGroup >> Availability Databases >> right-click the database that is not synchronizing, and choose “Join Availability Group”.
* As a last option, re-run the SQL Server Agent (or Windows Scheduled Task) job that re-creates the Dashboard database nightly, and if this doesn’t succeed, check the log file at D:\AutomatedBackup.log to see what has gone wrong during the process.
  + To manually add a new database to the high availability group (cluster), or learn about the process in detail, read the following the instructions at the Microsoft documentation website: https://docs.microsoft.com/en-us/sql/database-engine/availability-groups/windows/availability-group-add-a-database?view=sql-server-2014