SQL Moderation Hack – SSIS Migration Lab

V2.3

Contents

[PROBLEM STATEMENT 2](#_Toc73090297)

[LAB INSTRUCTIONS 2](#_Toc73090298)

[Stage 1 – Upgrade Package using the Upgrade Wizard. 2](#_Toc73090299)

[Stage 2 – Convert to Project Deployment mode & update connection string. 6](#_Toc73090300)

[Stage 3 – Deploy Package to the SSISDB on the Managed Instance 12](#_Toc73090301)

[Stage 4 – Verify Deployment and test run package. 18](#_Toc73090302)

[Optional Stage 5 – Schedule Package using SQL Server Agent 19](#_Toc73090303)

[LAB ENVIROMENT 20](#_Toc73090304)

[APPENDIX 21](#_Toc73090305)

[Summary of Logins and Accounts Used 21](#_Toc73090306)

[TEAMXX VM RDP details 21](#_Toc73090307)

[Target SQL Server (Azure SQL Managed Instance) 21](#_Toc73090308)

# PROBLEM STATEMENT

In Lab 1 of this hack, you have migrated 3 databases to Azure for the application Transaction Reporting Application.

Now that the databases for the Transactional Reporting Application have been migrated, there is a set of additional SSIS packages on the LEGACYSQL2008 server that also require migration to the SQL Managed Instance for the central Data Warehouse.

Task: Migrate SSIS from SQL Server 2008r2 to suitable environment, with a successful run of the package, verifying of the data and scheduling of package.

# LAB INSTRUCTIONS

Time: 30 Mins

**For Connection Strings and Passwords see sections LAB ENVIROMENT and APPENDIX at the end of this document**

## Stage 1 – Upgrade Package using the Upgrade Wizard.

In this section we will be upgrading the Legacy SSIS package so that it can be migrated to Azure.

|  |  |  |
| --- | --- | --- |
| **Narrative** | **Screenshot** | **Notes** |
| Open the SQL 2008 Solution using Visual Studio 2017.  Open the folder:  **C:\\_SQLHACK\_\LABS\Part 2 – SSIS Migration\SSISDW**  Right click the **SSISDW.sln** solution file  Open with **Visual Studio 2017** |  | You will need to RDP onto the TEAM virtual machine to complete this task.  For connection details see **APPENDIX - TEAMXX VM RDP details** |
| The package upgrade will require confirmation as this is a one-way process.  You will be prompted with “**Review Project and Solution Changes**.”  **Click OK** to acknowledge this is a one-way process. | Graphical user interface, text, application, email  Description automatically generated | If you make a mistake there is a BACKUP folder which can be used to copy the sln and project files.  Exit the process, replace the files and restart at Step 1. |
| A **Migration report** will be presented.  **Read and close the Migration report** – No further action is required. |  | If you make a mistake there is a BACKUP folder which can be used to copy the sln and project files.  Exit the process, replace the files and restart at Step 1. |
| The SSIS Package Upgrade Wizard will be presented.  **Click Next** to acknowledge the Starting page. |  | If you make a mistake there is a BACKUP folder which can be used to copy the sln and project files.  Exit the process, replace the files and restart at Step 1. |
| Select the projects to upgrade.  Ensure the following are ticked:   * **Existing Package name** * **PopulateDW.dtsx**   **Click Next** to confirm the above are to be upgraded. |  | If you make a mistake there is a BACKUP folder which can be used to copy the sln and project files.  Exit the process, replace the files and restart at Step 1. |
| Select Package Management options.  Select the following options:   * **Update connection strings to use new provider names.** * **Continue upgrade process when a package upgrade fails.** * **Ignore configurations.**   **Click Next** to confirm the management options. |  | If you make a mistake there is a BACKUP folder which can be used to copy the sln and project files.  Exit the process, replace the files and restart at Step 1. |
| Complete the upgrade Wizard.  **Click Finish** to complete the Wizard.  Once the upgrade wizard has completed:  **Click Close** to complete this stage of the process. | Graphical user interface, text, application, email  Description automatically generated | If you make a mistake there is a BACKUP folder which can be used to copy the sln and project files.  Exit the process, replace the files and restart at Step 1. |

## Stage 2 – Convert to Project Deployment mode & update connection string.

In this section we will be converting the DTSX package into a Project Deployment model and correcting the DTSX package connection strings to use the new SQL Server Managed Instance using Visual Studio 2017.

|  |  |  |
| --- | --- | --- |
| **Narrative** | **Screenshot** | **Notes** |
| Now the Solution is upgraded, it will be open in Visual Studio 2017.  In Solution Explorer:  **Double Click PopulateDW.dtsx** to open it. | Graphical user interface, text, application  Description automatically generated | If Visual Studio 2017 is not open, please confirm Stage 1 has been completed:  Open the folder:  **C:\\_SQLHACK\_\LABS\Part 2 – SSIS Migration\SSISDW**  Right click the **SSISDW.sln** solution file  Open with **Visual Studio 2017** |
| You will be prompted to Synchronise connection strings:  **Click OK** to acknowledge connection strings will be updated. |  |  |
| The SSIS package will require conversion to a Project Deployment Model.   * **Right Click SSISDW** (package deployment model) * **Select Convert to Project Deployment Model** | Graphical user interface, application  Description automatically generated |  |
| The Project Conversion Wizard will begin.  Accept all the defaults on each page (until the review tab):   * **Click Next** * **Click Next** * **Click Next** * **Click Next** * **Click Next** * **Click Next** | Graphical user interface, application  Description automatically generated |  |
| The Project Conversion Wizard is ready.  **Click Convert** to confirm your selections and start conversion. | Graphical user interface, application  Description automatically generated | Conversion should take a few moments. |
| The Project Conversion is complete.  **Click Ok** to Acknowledge the SQL Server Integration Services pop-up. | Graphical user interface, text, application, email  Description automatically generated |  |
| The Project Conversion is complete.  **Click Close** and **Save** the package. | Graphical user interface  Description automatically generated |  |
| Verify the Project Conversion has completed successfully.  **Check the Package name no longer says, “Package Deployment Model.”** |  |  |
| Correct the Connection Managers.  **Double Click** the **SQL Server Connection** in the **Connection Managers**. |  |  |
| Within the Connection Manager, update to the new connection details.   * **Server Name:** **(See Appendix - Target SQL Server)** * **Authentication: SQL Server Authentication** * **User Name:** **(See Appendix - Target SQL Server)** * **Password:** **(See Appendix - Target SQL Server)** * **Select or Enter Database name: 2008DW**   Once the settings above are complete.   * **Click Test Connection** to test the connection. * **Click OK** to save. | Graphical user interface, application, email  Description automatically generated | Please See Appendix - Target SQL Serverwithin this document for full details on the connection settings. |
| Convert the connection to a project connection.   * **Right Click** Connection. * **Click Convert** to Project Connection. | Graphical user interface, application  Description automatically generated |  |
| Verify you now have a project connection.  **Check Connection Manager** has the **SQL Server 1.conmgr.** | Text  Description automatically generated |  |
| Test the package with the new connection manager.  From the **Command bar**, **select Start** to Test the package. | Graphical user interface, application, Word  Description automatically generated |  |

## Stage 3 – Deploy Package to the SSISDB on the Managed Instance

In this section we will be deploying the fixed package onto the SSIS integration runtime and SSISDB held within the Managed Instance.

|  |  |  |
| --- | --- | --- |
| **Narrative** | **Screenshot** | **Notes** |
| Set the Target Server version to SQL Server 2017 as SQL server 2019 is not yet supported.  In Solution Explorer:  **Right Click SSISDW package** and **select properties.** |  | Please ensure you have completed Stage 1 and Stage 2 successfully. |
| In the SSISDW property page.   * **Select “General”** * **Change** “TargetServerVersion” to **SQL Server 2017** * **Click Apply** |  |  |
| If you do get the “**Do you want to reload**” message, **click No to All.**  **Click OK** to complete setting the SQL Target Version. | Graphical user interface, text, application, Word  Description automatically generated |  |
| We now need to deploy the package.  **Right Click SSISDW package** and **select Deploy.** |  |  |
| Integration Services Deployment Wizard will be started.  **Click Next** to acknowledge introduction. | Graphical user interface, text, application, email  Description automatically generated |  |
| Integration Services Deployment Wizard.   * **Select SSIS in Azure Data Factory.** * **Click Next.** | Graphical user interface, text, application, email, Teams  Description automatically generated |  |
| Within the Select Destination tab with the destination details:   * **Server Name:** **(See Appendix - Target SQL Server)** * **Authentication: SQL Server Authentication** * **User Name:** **(See Appendix - Target SQL Server)** * **Password:** **(See Appendix - Target SQL Server)** * **Path: Select Browse and Add your TEAM name as a Folder. Example if you are in TEAM 1, enter a folder name of TEAM01.** | Graphical user interface, text, application  Description automatically generated | Please See Appendix - Target SQL Serverwithin this document for full details on the connection settings. |
| Check the Select Destination details.  **Check details** & **click Next** to continue. | Graphical user interface, text, application, email  Description automatically generated |  |
| The package is now ready to deploy to the SSISDB on the Managed instance.   * **Click Deploy** to begin deployment. * **Click Close** to confirm Successful Deployment. | Graphical user interface, text, application  Description automatically generated |  |

## Stage 4 – Verify Deployment and test run package.

In this section we will be verifying the package has been deployed successfully to the Managed Instance and running the Package to ensure it is working correctly.

|  |  |  |
| --- | --- | --- |
| **Narrative** | **Screenshot** | **Notes** |
| Using SQL Server Management Studio (SSMS), connect to the SQL Server Managed Instance.  In **SSMS, navigate to Integration Service Catalogs**:   * **Select Projects** * **Select Your TEAM folder**   **Verify** the **SSISDW** Package has been deployed. | Graphical user interface, application  Description automatically generated | For connection details please See Appendix - Target SQL Server**.** |
| Test the SISS Package migrate to the Managed Instance.  Execute the SSIS Package   * **Right Click** the **SSISDW** package * Select **Execute** |  |  |
| Test the SISS Package migrate to the Managed Instance.  Within the Execute Package window:   * Ensure the Package **PopulateDW.dtsx is selected.** * **Click OK** | Graphical user interface, application  Description automatically generated |  |
| View the execution report once complete.  You be notified that the Selected Packages have been queued to Execute.   * **Click Yes** to view the Execution Report.   Once the Execution report has loaded:   * **Click refresh** until the package has completed.   **Congratulations on successfully migrating and upgrading an SSIS package to Azure.** | Graphical user interface, application  Description automatically generated |  |

## Optional Stage 5 – Schedule Package using SQL Server Agent

If you have time, schedule the package to run with a Job using SQL Server Agent

**Note: No instructions provided for this task.**

# LAB ENVIROMENT



**NOTE: There are 20 workshop environments using a SHARED source SQL Server and target Azure SQL Database Managed Instance. Please be respectful of only migrating your teams Databases and Logins.**

# APPENDIX

## Summary of Logins and Accounts Used

There are several different environments that you need to login/connect to during the labs. Sometimes you will need to login into the same environment with different accounts depending on what you are doing e.g., logging into SQL Server with a standard or sysadmin privileged account.

## TEAMXX VM RDP details

|  |  |
| --- | --- |
| **Machine IP address**  (Use for RDP connection) |  |
| **Machine Name**  (Replace XX with Team number) | **vm-TEAMxx** |
| **Win10 Username:**  (Use for RDP connection) | **Demouser** |
| **Win10 Password:**  (Use for RDP connection) | **Demo@pass1234567** |
| **Resource Group** | **SQLHACK-TEAM-VMs** |

## Target SQL Server (Azure SQL Managed Instance)

|  |  |
| --- | --- |
| **Server Name** | ***C:\\_SQLHACK\_\LABS\01-Data\_Migration\ManagedInstanceFQDN.txt*** |
| **Resource Group** | **SQLHACK-SHARED** |
| **Sysadmin Login Name:** (Use for Migrations) | **DemoUser** |
| **Admin Login Password:** | **Demo@pass1234567** |