SQL Moderation Hack – SSIS Migration Lab

V2.3

Contents

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.

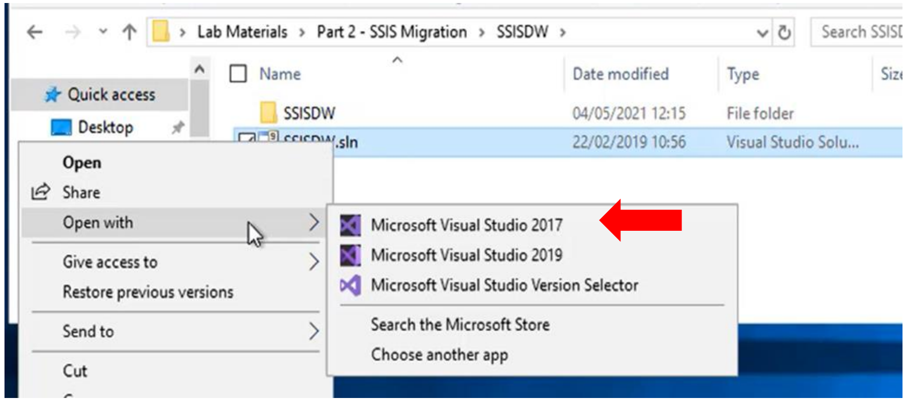
LAB INSTRUCTIONS

Time: 30 Mins

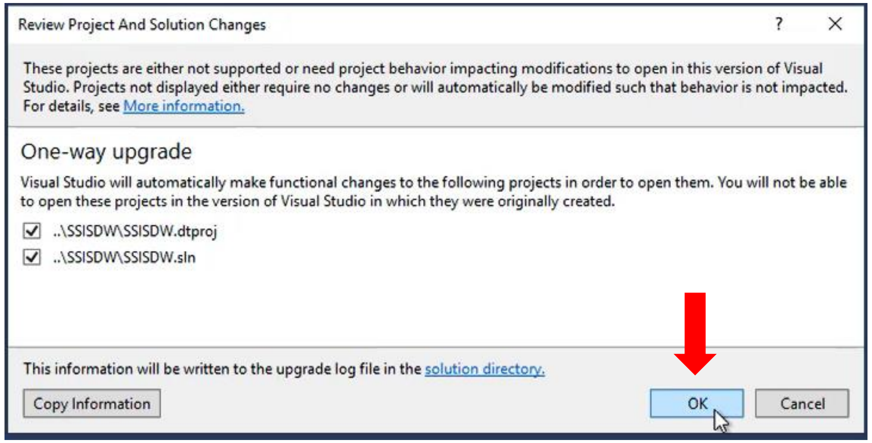
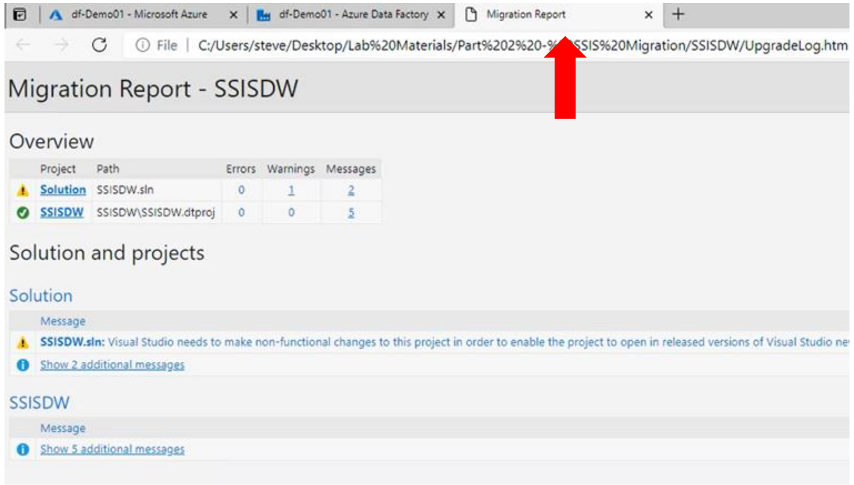
For Connection Strings and Passwords see sections LAB ENVIROMENT and APPENDIX at the end of this document

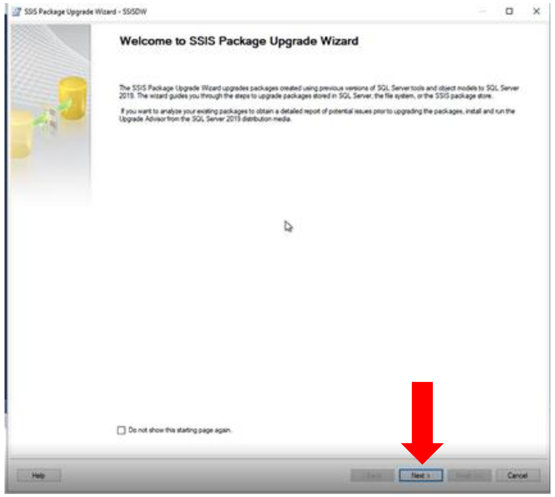
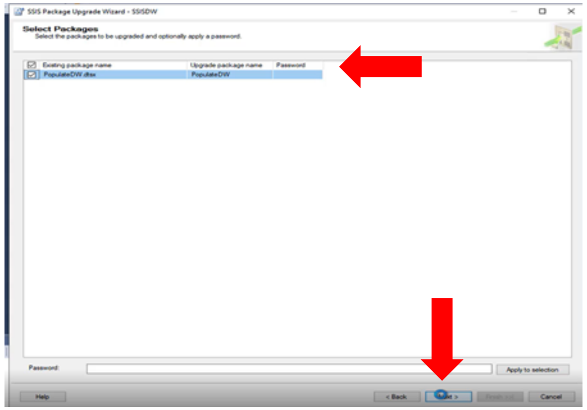
Stage 1 – Upgrade Package using the Upgrade Wizard.

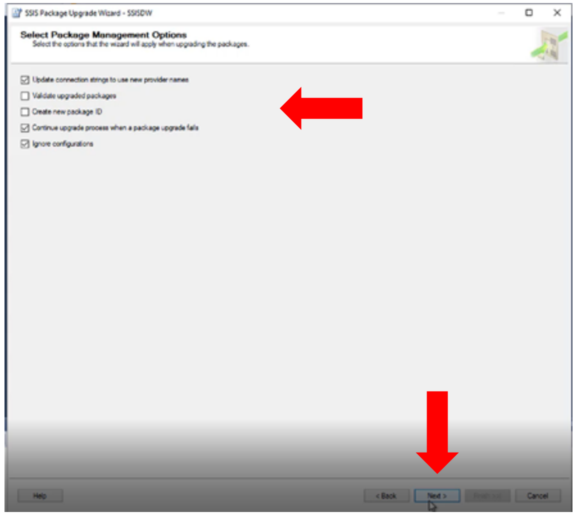
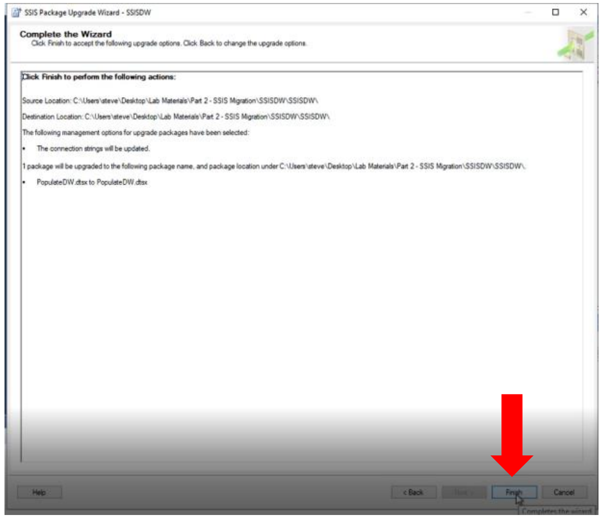
In this section we will be upgrading the Legacy SSIS package so that it can be migrated to Azure.

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**

Lab Materials > Part 2 - SSIS Migration > SSISDW >
Search SSISD
Quick access
Desktop
Name
Date modified
Type
Size
SSISDW
04/05/2021 12:15
File folder
SSISDW.sln
22/02/2019 10:56
Visual Studio Solu...
Open
Share
Open with
Give access to
Restore previous versions
Send to
Cut
Microsoft Visual Studio 2017
Microsoft Visual Studio 2019
Microsoft Visual Studio Version Selector
Search the Microsoft Store
Choose another app

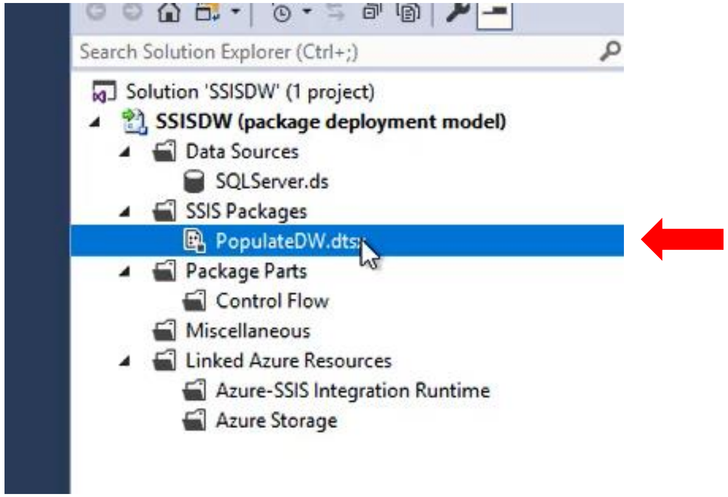
The package upgrade will require confirmation as this is a one-way process. You will be prompted with "**Review Project and Solution Changes**." **Click OK** to acknowledge this is a one-way process.	Review Project And Solution Changes These projects are either not supported or need project behavior impacting modifications to open in this version of Visual Studio. Projects not displayed either require no changes or will automatically be modified such that behavior is not impacted. For details, see More information. One-way upgrade Visual Studio will automatically make functional changes to the following projects in order to open them. You will not be able to open these projects in the version of Visual Studio in which they were originally created. ☑ ..\SSISDW\SSISDW.dtproj ☑ ..\SSISDW\SSISDW.sln This information will be written to the upgrade log file in the solution directory. 	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.	 df-Demo01 - Microsoft Azure df-Demo01 - Azure Data Factory Migration ReportFile | C:/Users/steve/Desktop/Lab%20Materials/Part%202%20-% SSIS%20Migration/SSISDW/UpgradeLog.htm Migration Report - SSISDW Overview Project · Path · Errors · Warnings · Messages Solution · SSISDW.sln · 0 · 1 · 2 SSISDW · SSISDW\SSISDW.dtproj · 0 · 0 · 5 Solution and projects Solution Message SSISDW.sln: Visual Studio needs to make non-functional changes to this project in order to enable the project to open in released versions of Visual Studio ne Show 2 additional messages SSISDW Message Show 5 additional messages	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.

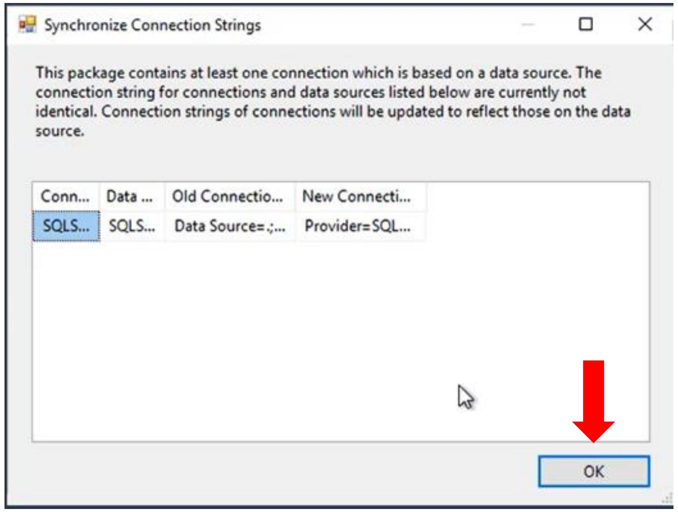
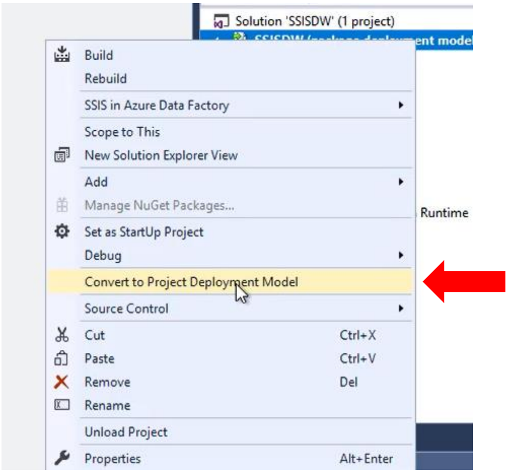
The SSIS Package Upgrade Wizard will be presented. **Click Next** to acknowledge the Starting page.	 SSIS Package Upgrade Wizard - SSISDW Welcome to SSIS Package Upgrade Wizard The SSIS Package Upgrade Wizard upgrades packages created using previous versions of SQL Server tools and object models to SQL Server 2019. The wizard guides you through the steps to upgrade packages stored in SQL Server, the file system, or the SSIS package store. If you want to analyze your existing packages to obtain a detailed report of potential issues prior to upgrading the packages, install and run the Upgrade Advisor from the SQL Server 2019 distribution media. ☐ Do not show this starting page again. 	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.	 SSIS Package Upgrade Wizard - SSISDW Select Packages Select the packages to be upgraded and optionally apply a password. ☑ Existing package name | Upgrade package name | Password ☑ PopulateDW.dtsx | PopulateDW | Password: Apply to selection Help < Back Next > Finish >>| Cancel	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.

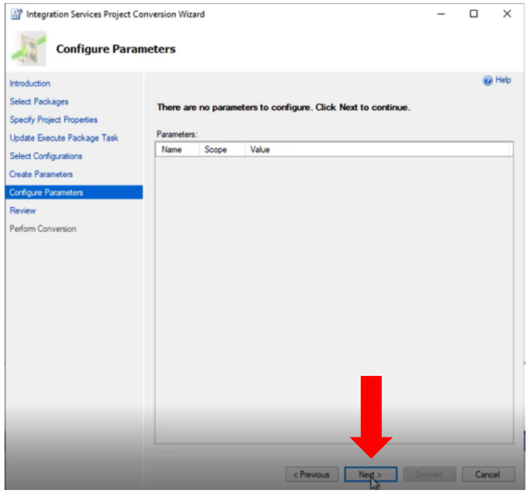
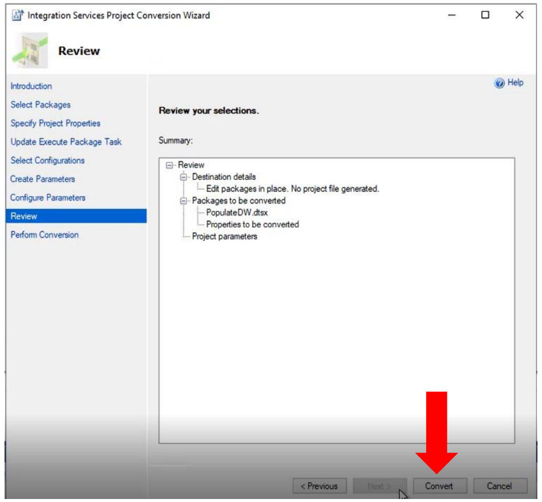
Select Package Management options. Select the following options: • **Update connection strings to use new provider names.** • **Continue upgrade process when a package upgrade fails.** • **Ignore configurations.** **Click Next** to confirm the management options.	 SSIS Package Upgrade Wizard - SSISDW **Select Package Management Options** Select the options that the wizard will apply when upgrading the packages. ☑ Update connection strings to use new provider names □ Validate upgraded packages □ Create new package ID ☑ Continue upgrade process when a package upgrade fails ☑ Ignore configurations 	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.	 SSIS Package Upgrade Wizard - SSISDW **Complete the Wizard** Click Finish to accept the following upgrade options. Click Back to change the upgrade options. **Click Finish to perform the following actions:** Source Location: C:\Users\steve\Desktop\Lab Materials\Part 2 - SSIS Migration\SSISDW\SSISDW\ Destination Location: C:\Users\steve\Desktop\Lab Materials\Part 2 - SSIS Migration\SSISDW\SSISDW\ The following management options for upgrade packages have been selected: • The connection strings will be updated. 1 package will be upgraded to the following package name, and package location under C:\Users\steve\Desktop\Lab Materials\Part 2 - SSIS Migration\SSISDW\SSISDW\. • PopulateDW.dtsx to PopulateDW.dtsx Help < Back Next > Finish Cancel Completes the wizard	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.

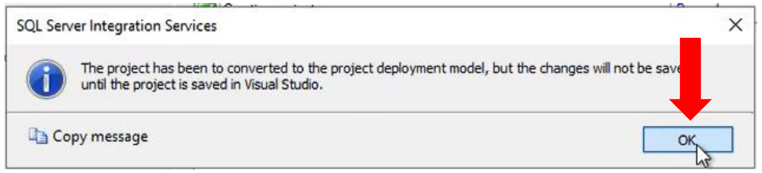
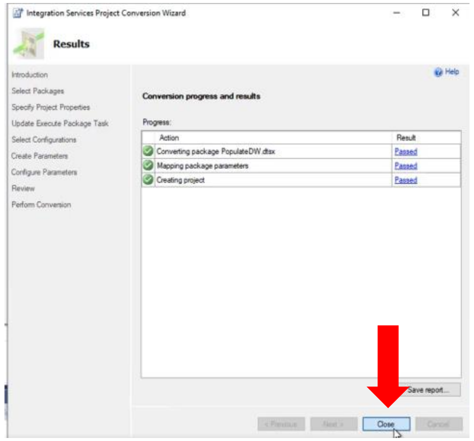
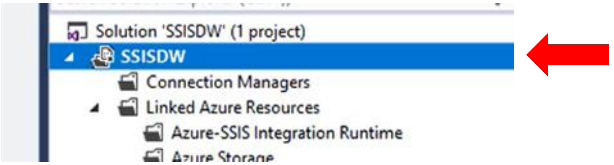
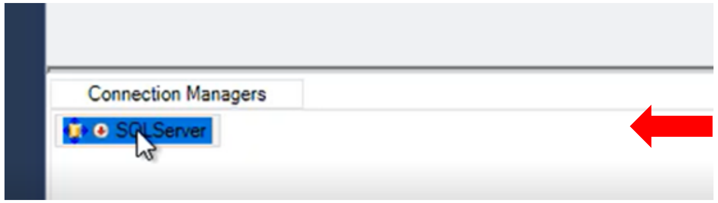
Stage 2 – Convert to Project Deployment mode & update connection string.

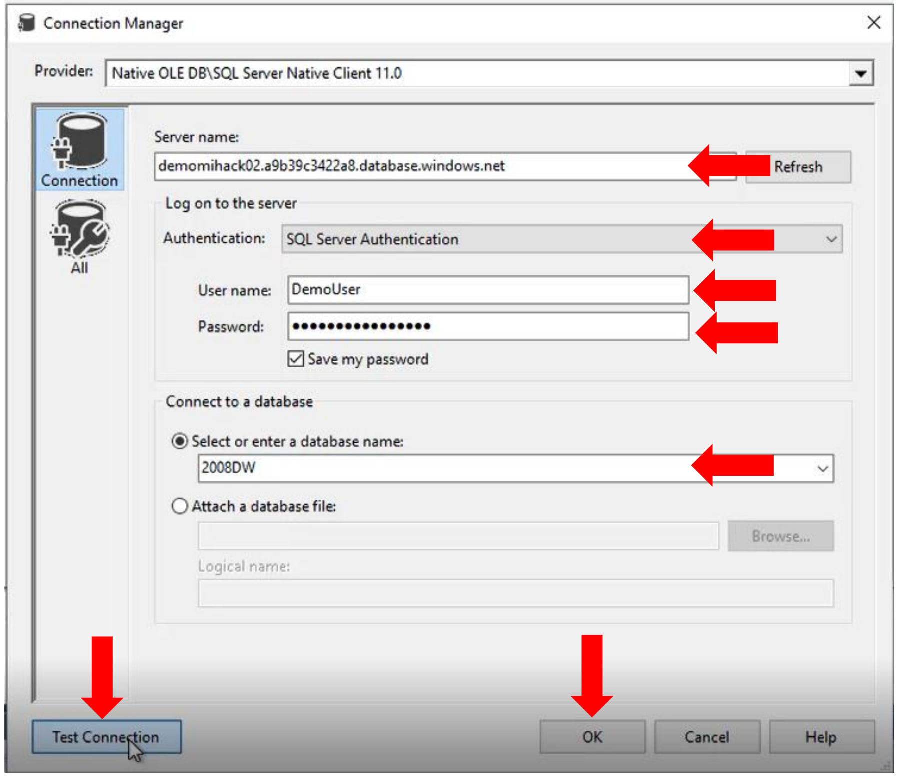
In this section we will be converting the DTSX package into a Project Deployment model and correcting the DTSX package connection strings to use the new SQL Server Managed Instance using Visual Studio 2017.

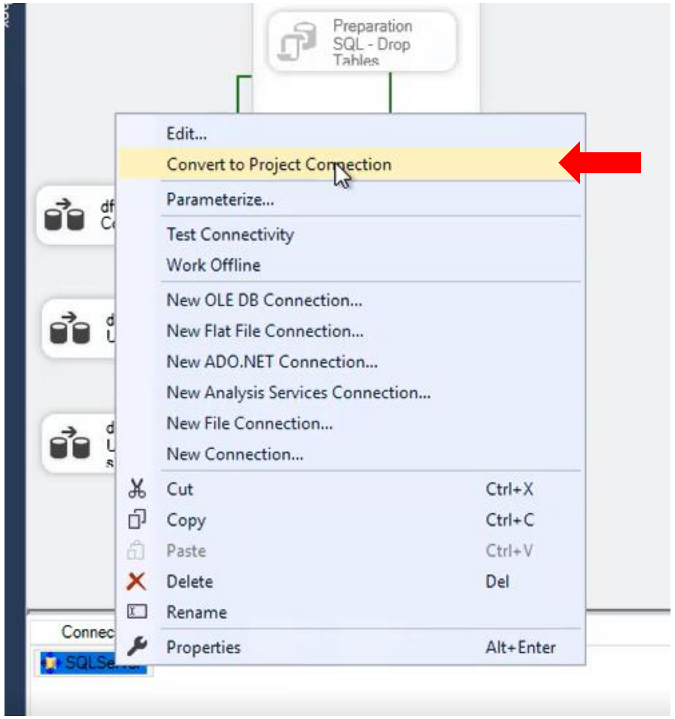
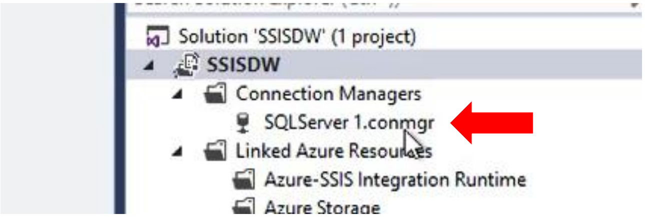
Narrative	Screenshot	Notes
Now the Solution is upgraded, it will be open in Visual Studio 2017. In Solution Explorer: **Double Click PopulateDW.dtsx** to open it.	 Search Solution Explorer (Ctrl+;) Solution 'SSISDW' (1 project) SSISDW (package deployment model) Data Sources SQLServer.ds SSIS Packages PopulateDW.dtsx Package Parts Control Flow Miscellaneous Linked Azure Resources Azure-SSIS Integration Runtime Azure Storage	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**

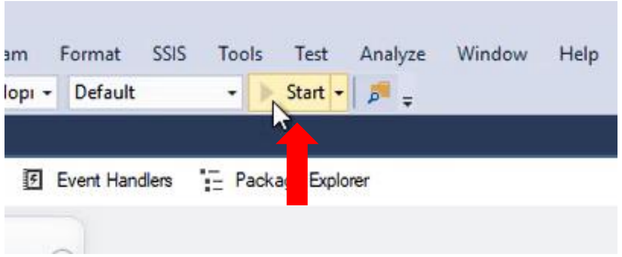
You will be prompted to Synchronise connection strings: **Click OK** to acknowledge connection strings will be updated.	 Synchronize Connection Strings This package contains at least one connection which is based on a data source. The connection string for connections and data sources listed below are currently not identical. Connection strings of connections will be updated to reflect those on the data source. Conn... | Data ... | Old Connectio... | New Connecti... SQLS... | SQLS... | Data Source=.;... | Provider=SQL... 	
The SSIS package will require conversion to a Project Deployment Model. - **Right Click SSISDW** (package deployment model) - **Select Convert to Project Deployment Model**	 Solution 'SSISDW' (1 project) Build Rebuild SSIS in Azure Data Factory Scope to This New Solution Explorer View Add Manage NuGet Packages... Set as StartUp Project Debug Convert to Project Deployment Model Source Control Cut Ctrl+X Paste Ctrl+V Remove Del Rename Unload Project Properties Alt+Enter Runtime	

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**	 Integration Services Project Conversion Wizard **Configure Parameters** There are no parameters to configure. Click Next to continue. Parameters: Name · Scope · Value Help Introduction Select Packages Specify Project Properties Update Execute Package Task Select Configurations Create Parameters Configure Parameters Review Perform Conversion < Previous Next > Convert Cancel	
The Project Conversion Wizard is ready. **Click Convert** to confirm your selections and start conversion.	 Integration Services Project Conversion Wizard **Review** Review your selections. Summary: ⊟ Review ⊟ Destination details Edit packages in place. No project file generated. ⊟ Packages to be converted PopulateDW.dtsx Properties to be converted Project parameters Help Introduction Select Packages Specify Project Properties Update Execute Package Task Select Configurations Create Parameters Configure Parameters Review Perform Conversion < Previous Next > Convert Cancel	Conversion should take a few moments.

The Project Conversion is complete. Click Ok to Acknowledge the SQL Server Integration Services pop-up.	 SQL Server Integration Services The project has been to converted to the project deployment model, but the changes will not be save until the project is saved in Visual Studio. 	
The Project Conversion is complete. Click Close and Save the package.	 Integration Services Project Conversion Wizard Results Conversion progress and results Progress: Action / Result Converting package PopulateDW.dtsx / Passed Mapping package parameters / Passed Creating project / Passed Introduction Select Packages Specify Project Properties Update Execute Package Task Select Configurations Create Parameters Configure Parameters Review Perform Conversion Help Save report... < Previous Next > Close Cancel	
Verify the Project Conversion has completed successfully. **Check the Package name no longer says, "Package Deployment Model."**	 Solution 'SSISDW' (1 project) SSISDW Connection Managers Linked Azure Resources Azure-SSIS Integration Runtime Azure Storage	
Correct the Connection Managers. **Double Click** the **SQL Server Connection** in the **Connection Managers**.	 	

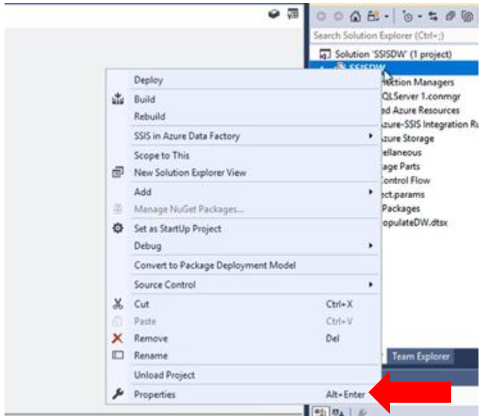
Within the Connection Manager, update to the new connection details. • **Server Name: (See Appendix - Target SQL Server)** • **Authentication: SQL Server Authentication** • **User Name: (See Appendix - Target SQL Server)** • **Password: (See Appendix - Target SQL Server)** • **Select or Enter Database name: 2008DW** Once the settings above are complete. ▪ **Click Test Connection** to test the connection. ▪ **Click OK** to save.	 Connection Manager Provider: Native OLE DB\SQL Server Native Client 11.0 Server name: demomihack02.a9b39c3422a8.database.windows.net Log on to the server Authentication: SQL Server Authentication User name: DemoUser Password: •••••••••••••••• ☑ Save my password Connect to a database ● Select or enter a database name: 2008DW ○ Attach a database file: Logical name: Connection All Refresh Browse... Test Connection OK Cancel Help	Please See Appendix - Target SQL Server within this document for full details on the connection settings.

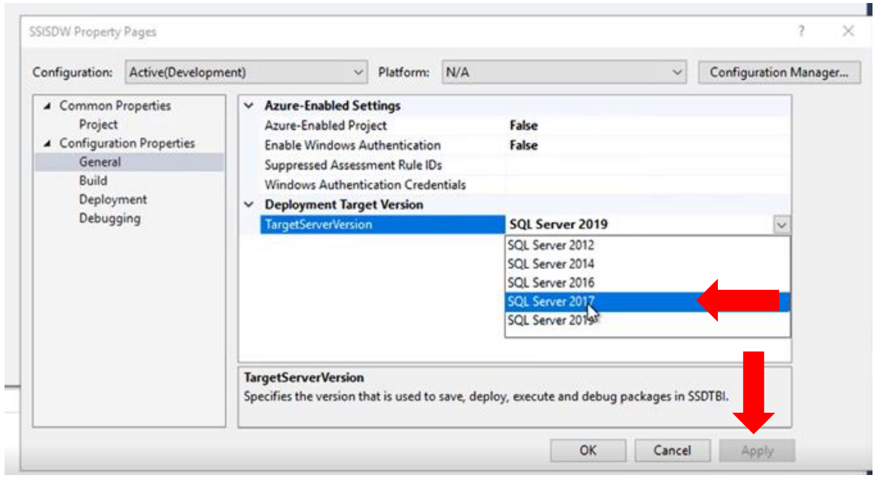
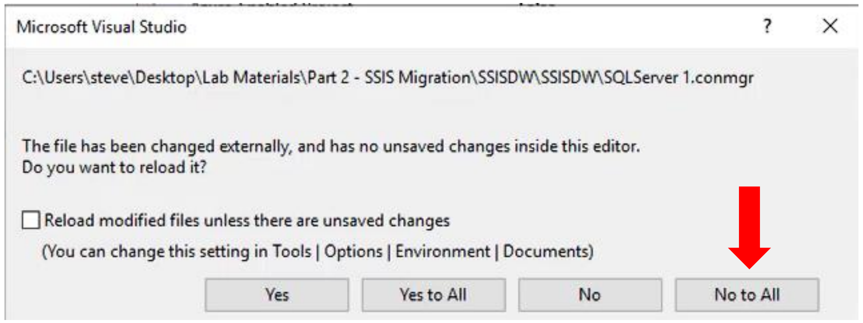
Convert the connection to a project connection. • **Right Click** Connection. • **Click Convert** to Project Connection.	 Preparation SQL - Drop Tables Edit... Convert to Project Connection Parameterize... Test Connectivity Work Offline New OLE DB Connection... New Flat File Connection... New ADO.NET Connection... New Analysis Services Connection... New File Connection... New Connection... Cut Ctrl+X Copy Ctrl+C Paste Ctrl+V Delete Del Rename Properties Alt+Enter Connec SQLSe	
Verify you now have a project connection. **Check Connection Manager** has the **SQL Server 1.conmgr.**	 Solution 'SSISDW' (1 project) SSISDW Connection Managers SQLServer 1.conmgr Linked Azure Resources Azure-SSIS Integration Runtime Azure Storage	

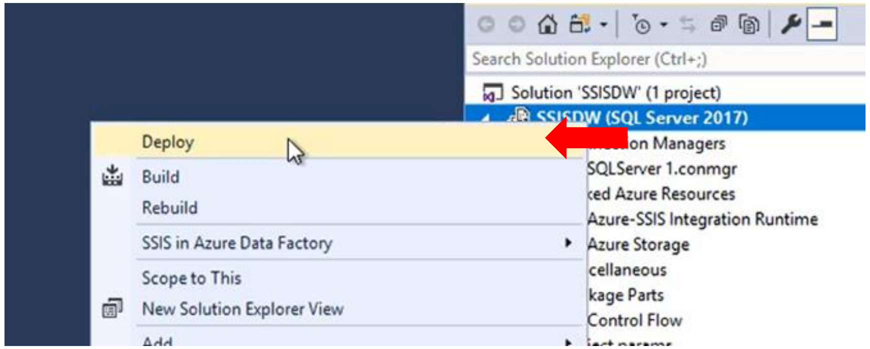
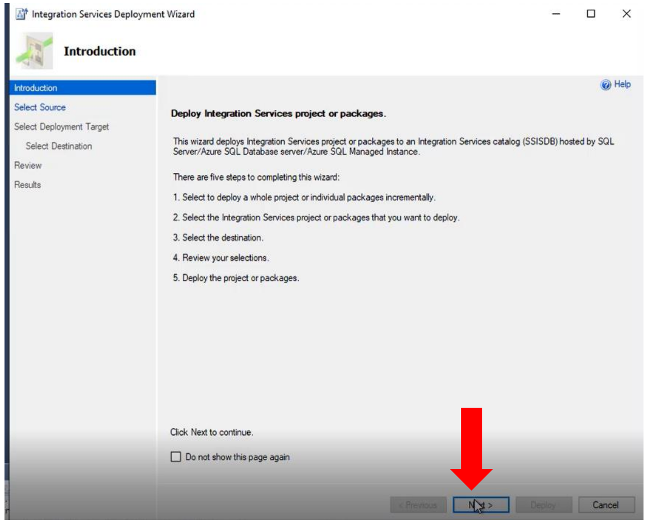
Test the package with the new connection manager. From the **Command bar**, **select Start** to Test the package.	 am Format SSIS Tools Test Analyze Window Help lopı Default Start Event Handlers Package Explorer	

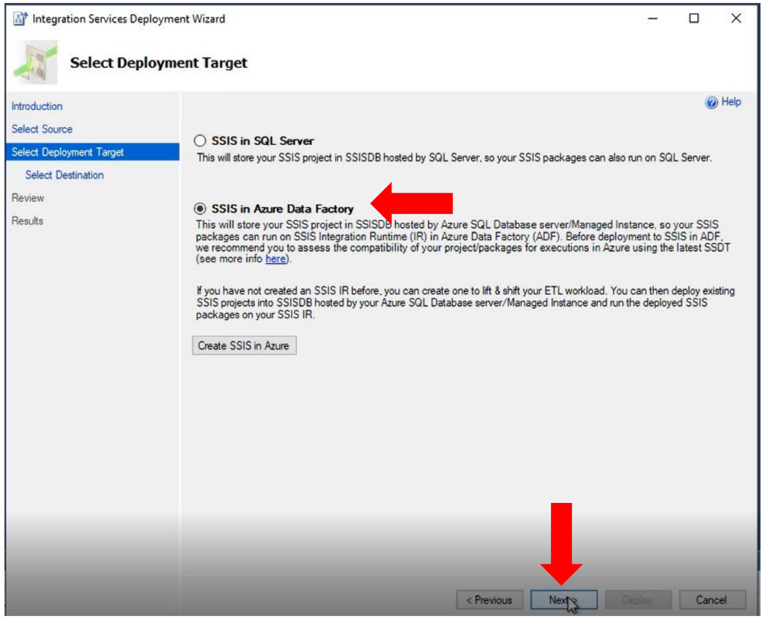
Stage 3 – Deploy Package to the SSISDB on the Managed Instance

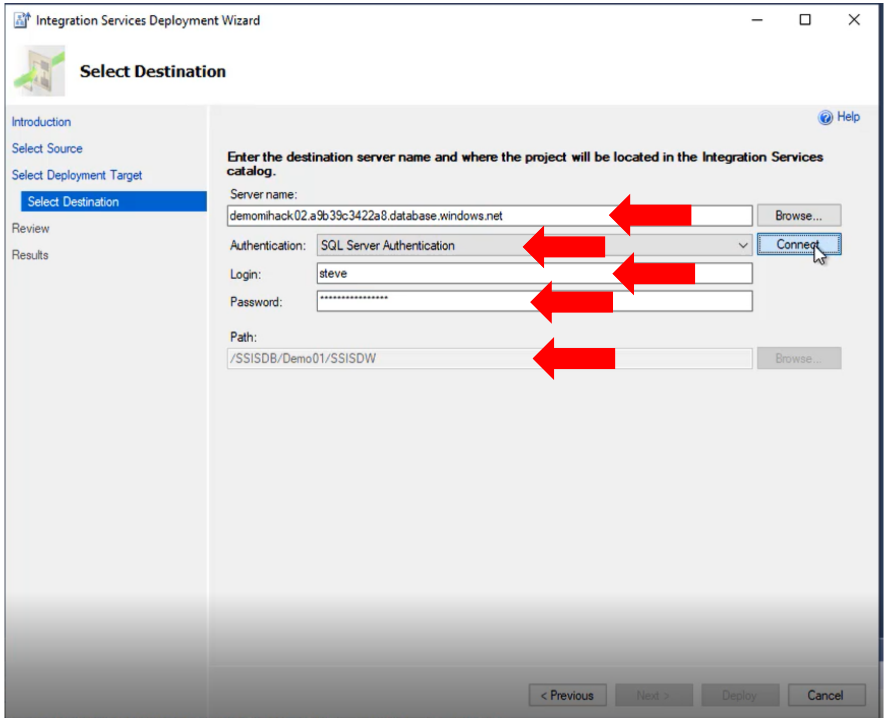
In this section we will be deploying the fixed package onto the SSIS integration runtime and SSISDB held within the Managed Instance.

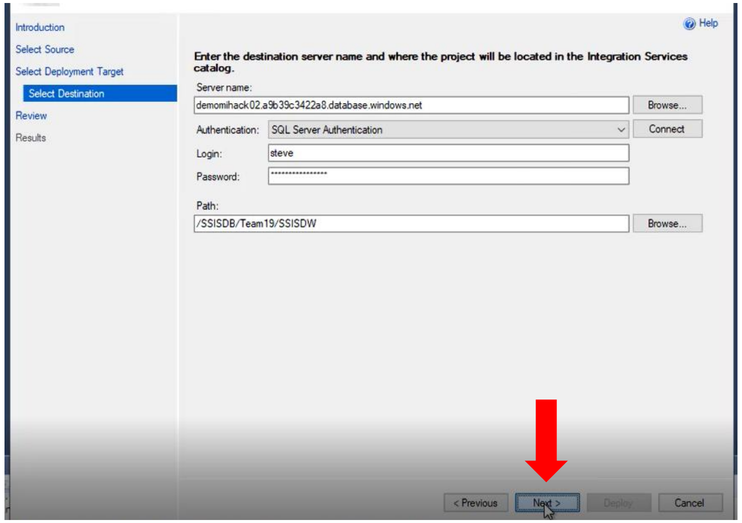
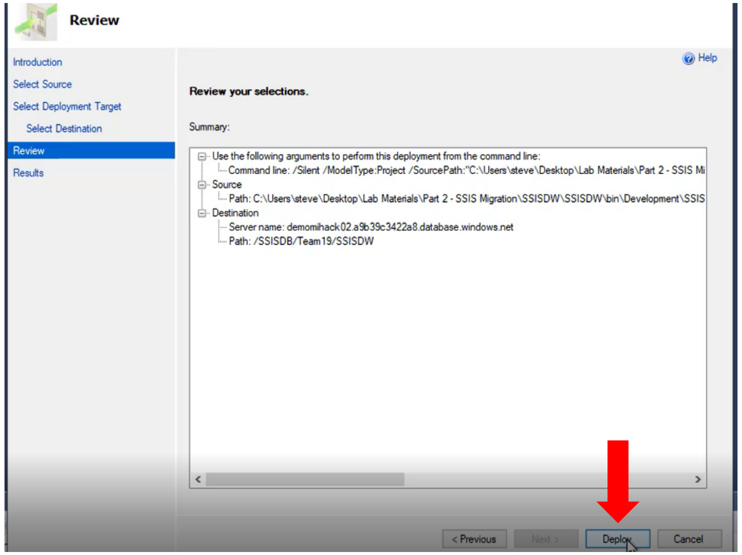
Narrative	Screenshot	Notes
Set the Target Server version to SQL Server 2017 as SQL server 2019 is not yet supported. In Solution Explorer: **Right Click SSISDW package** and **select properties.**	 Search Solution Explorer (Ctrl+;) Solution 'SSISDW' (1 project) SSISDW ction Managers QLServer 1.conmgr ed Azure Resources zure-SSIS Integration R zure Storage ellaneous age Parts ontrol Flow ect.params Packages opulateDW.dtsx Deploy Build Rebuild SSIS in Azure Data Factory Scope to This New Solution Explorer View Add Manage NuGet Packages... Set as StartUp Project Debug Convert to Package Deployment Model Source Control Cut Ctrl+X Paste Ctrl+V Remove Del Rename Unload Project Properties Alt+Enter Team Explorer	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**	 SSISDW Property Pages Configuration: Active(Development) Platform: N/A Common Properties Project Configuration Properties General Build Deployment Debugging Azure-Enabled Settings Azure-Enabled Project False Enable Windows Authentication False Suppressed Assessment Rule IDs Windows Authentication Credentials Deployment Target Version TargetServerVersion SQL Server 2019 SQL Server 2012 SQL Server 2014 SQL Server 2016 SQL Server 2017 SQL Server 2019 TargetServerVersion Specifies the version that is used to save, deploy, execute and debug packages in SSDTBI. 	
If you do get the “**Do you want to reload**” message, **click No to All.** **Click OK** to complete setting the SQL Target Version.	 Microsoft Visual Studio C:\Users\steve\Desktop\Lab Materials\Part 2 - SSIS Migration\SSISDW\SSISDW\SQLServer 1.conmgr The file has been changed externally, and has no unsaved changes inside this editor. Do you want to reload it? □ Reload modified files unless there are unsaved changes (You can change this setting in Tools | Options | Environment | Documents) 	

We now need to deploy the package. **Right Click SSISDW package** and **select Deploy.**	 Search Solution Explorer (Ctrl+;) Solution 'SSISDW' (1 project) SSISDW (SQL Server 2017) Deploy Build Rebuild SSIS in Azure Data Factory Scope to This New Solution Explorer View Add on Managers SQLServer 1.conmgr ced Azure Resources Azure-SSIS Integration Runtime Azure Storage cellaneous kage Parts Control Flow	
Integration Services Deployment Wizard will be started. **Click Next** to acknowledge introduction.	 Integration Services Deployment Wizard Introduction Introduction Select Source Select Deployment Target Select Destination Review Results Help < Previous Next > Deploy Cancel Deploy Integration Services project or packages. This wizard deploys Integration Services project or packages to an Integration Services catalog (SSISDB) hosted by SQL Server/Azure SQL Database server/Azure SQL Managed Instance. There are five steps to completing this wizard: 1. Select to deploy a whole project or individual packages incrementally. 2. Select the Integration Services project or packages that you want to deploy. 3. Select the destination. 4. Review your selections. 5. Deploy the project or packages. Click Next to continue. □ Do not show this page again	

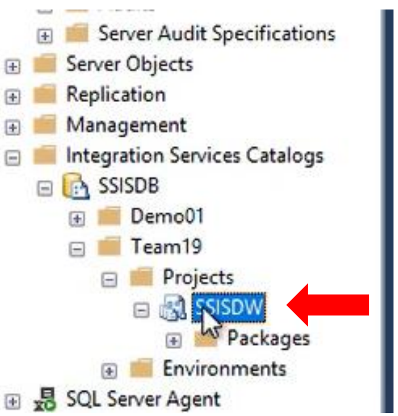
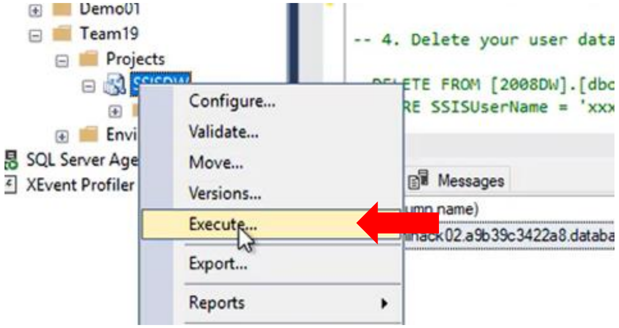
Integration Services Deployment Wizard. - **Select SSIS in Azure Data Factory.** - **Click Next.**	 Integration Services Deployment Wizard Select Deployment Target Introduction Select Source Select Deployment Target Select Destination Review Results Help Create SSIS in Azure < Previous Next Deploy Cancel ○ **SSIS in SQL Server** This will store your SSIS project in SSISDB hosted by SQL Server, so your SSIS packages can also run on SQL Server. ● **SSIS in Azure Data Factory** This will store your SSIS project in SSISDB hosted by Azure SQL Database server/Managed Instance, so your SSIS packages can run on SSIS Integration Runtime (IR) in Azure Data Factory (ADF). Before deployment to SSIS in ADF, we recommend you to assess the compatibility of your project/packages for executions in Azure using the latest SSDT (see more info here). If you have not created an SSIS IR before, you can create one to lift & shift your ETL workload. You can then deploy existing SSIS projects into SSISDB hosted by your Azure SQL Database server/Managed Instance and run the deployed SSIS packages on your SSIS IR.	

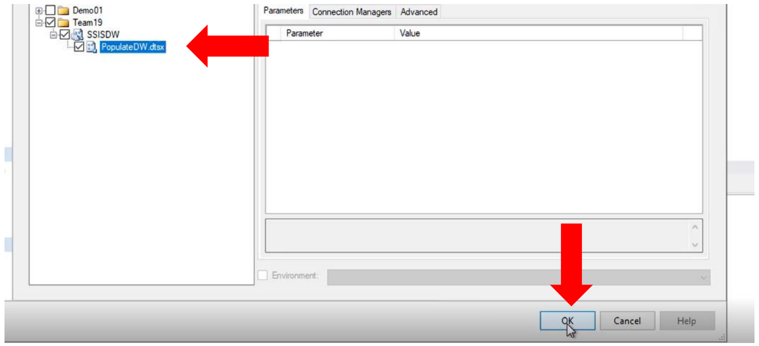
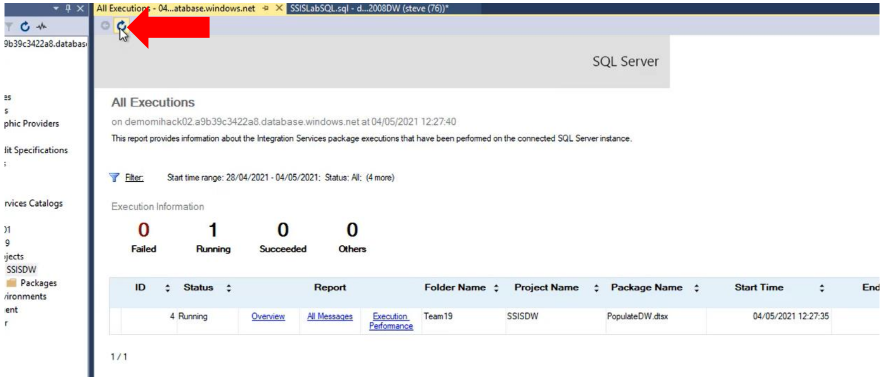
Within the Select Destination tab with the destination details: - **Server Name: (See Appendix - Target SQL Server)** - **Authentication: SQL Server Authentication** - **User Name: (See Appendix - Target SQL Server)** - **Password: (See Appendix - Target SQL Server)** - **Path: Select Browse and Add your TEAM name as a Folder. Example if you are in TEAM 1, enter a folder name of TEAM01.**	 Integration Services Deployment Wizard Select Destination Enter the destination server name and where the project will be located in the Integration Services catalog. Server name: demomihack02.a9b39c3422a8.database.windows.net Authentication: SQL Server Authentication Login: steve Password: **************** Path: /SSISDB/Demo01/SSISDW	Please See Appendix - Target SQL Server within this document for full details on the connection settings.

Check the Select Destination details. **Check details** & **click Next** to continue.	 Enter the destination server name and where the project will be located in the Integration Services catalog. Server name: demomihack02.a9b39c3422a8.database.windows.net Authentication: SQL Server Authentication Login: steve Password: **************** Path: /SSISDB/Team19/SSISDW Introduction Select Source Select Deployment Target Select Destination Review Results Help Browse... Connect Browse... < Previous Next > Deploy Cancel	
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.	 Review Review your selections. Summary: Use the following arguments to perform this deployment from the command line: Command line: /Silent /ModelType:Project /SourcePath:"C:\Users\steve\Desktop\Lab Materials\Part 2 - SSIS Mi Source Path: C:\Users\steve\Desktop\Lab Materials\Part 2 - SSIS Migration\SSISDW\SSISDW\bin\Development\SSIS Destination Server name: demomihack02.a9b39c3422a8.database.windows.net Path: /SSISDB/Team19/SSISDW Introduction Select Source Select Deployment Target Select Destination Review Results Help < Previous Next > Deploy Cancel	

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.	 Server Audit Specifications Server Objects Replication Management Integration Services Catalogs SSISDB Demo01 Team19 Projects SSISDW Packages Environments SQL Server Agent	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**	 Demo01 Team19 Projects SSISDW Envi SQL Server Age XEvent Profiler Configure... Validate... Move... Versions... Execute... Export... Reports Messages-- 4. Delete your user data DELETE FROM [2008DW].[dbc RE SSISUserName = 'xxx ump name) hack02.a9b39c3422a8.databa	

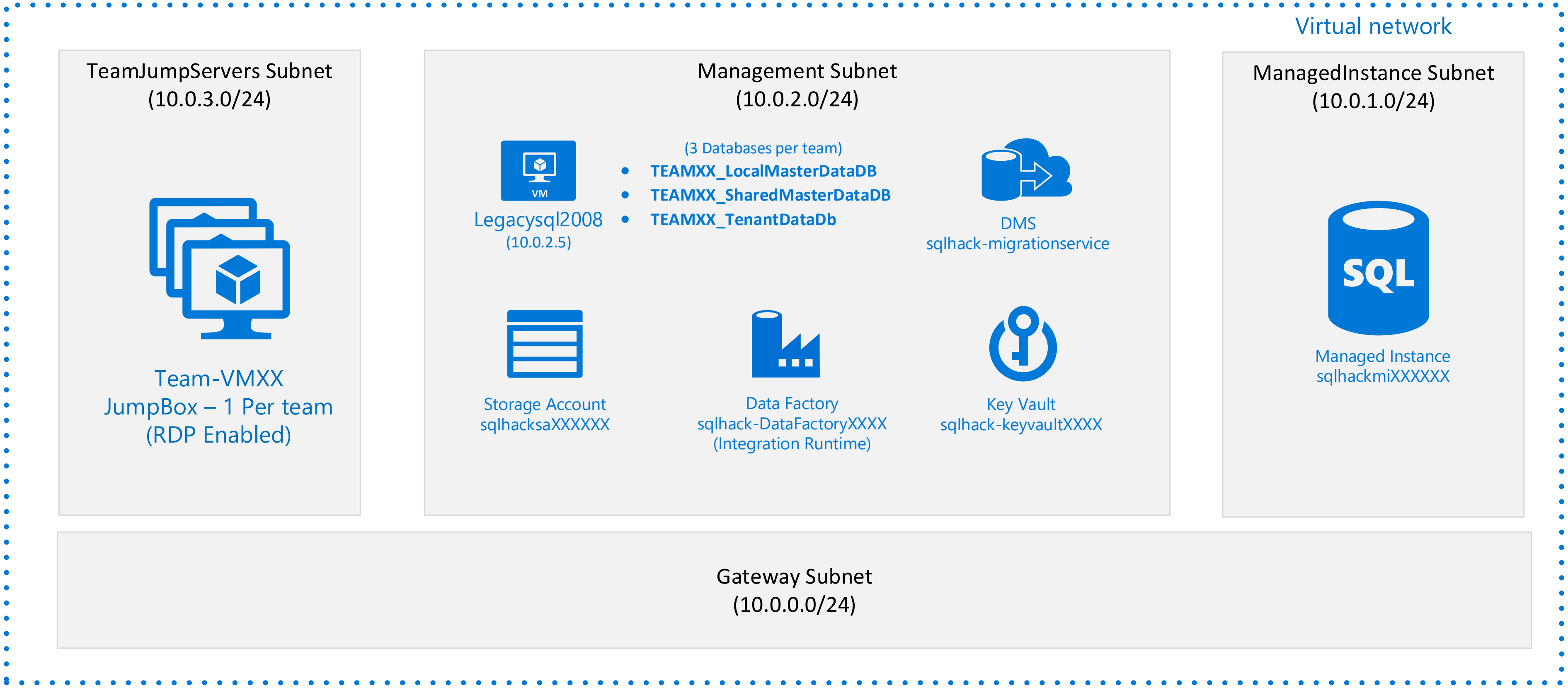
Test the SISS Package migrate to the Managed Instance. Within the Execute Package window: - Ensure the Package **PopulateDW.dtsx is selected.** - **Click OK**	 Demo01 ☑ Team19 ☑ SSISDW ☑ PopulateDW.dtsx Parameters | Connection Managers | Advanced Parameter | Value □ Environment: OK | Cancel | Help	
View 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.**	 All Executions - 04...atabase.windows.net | SSISLabSQL.sql - d...2008DW (steve (76))* SQL Server All Executions on demomihack02.a9b39c3422a8.database.windows.net at 04/05/2021 12:27:40 This report provides information about the Integration Services package executions that have been performed on the connected SQL Server instance. Filter: Start time range: 28/04/2021 - 04/05/2021; Status: All; (4 more) Execution Information 0 Failed | 1 Running | 0 Succeeded | 0 Others ID | Status | Report | Folder Name | Project Name | Package Name | Start Time | End 4 | Running | Overview | All Messages | Execution Performance | Team19 | SSISDW | PopulateDW.dtsx | 04/05/2021 12:27:35 1 / 1	

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**