

SQL Moderation Hack – SSIS Migration Lab

V2.4

Contents

PROBLEM STATEMENT	2
LAB INSTRUCTIONS	2
Stage 1 – Upgrade Package using the Upgrade Wizard.....	2
Stage 2 – Convert to Project Deployment mode & update connection string.....	4
Stage 3 – Deploy Package to the SSISDB on the Managed Instance	10
Stage 4 – Verify Deployment and test run package.....	21
Optional Stage 5 – Schedule Package using SQL Server Agent.....	24
LAB ENVIROMENT	24
Optional Stage 6 – How to Deploy Azure SSIS IR Manually	25
APPENDIX.....	31
Summary of Logins and Accounts Used.....	31
TEAMXX VM RDP details	31
Target SQL Server (Azure SQL Managed Instance)	31

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.

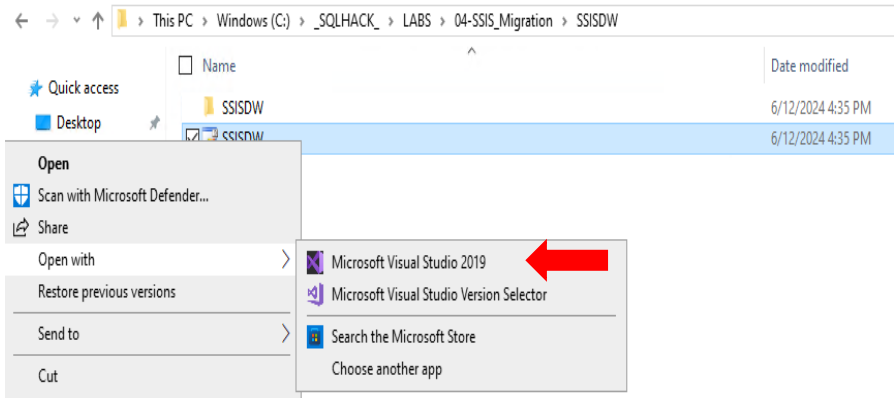
LAB INSTRUCTIONS

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.

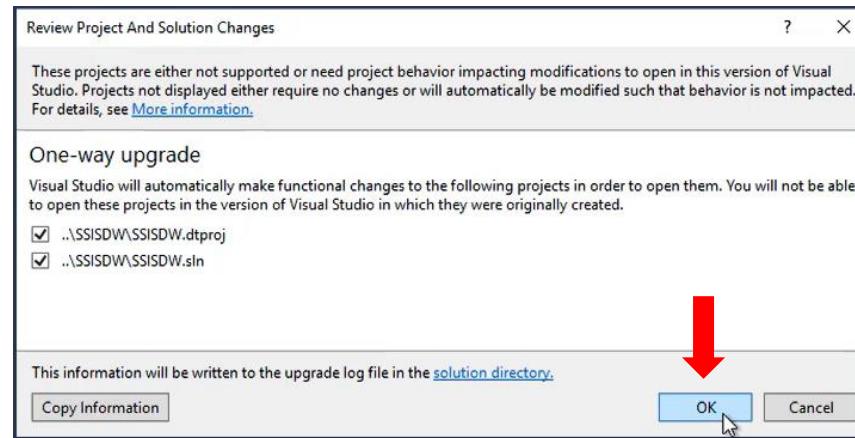
In this section we will be upgrading the Legacy SSIS package so that it can be migrated to Azure.

Narrative	Screenshot	Notes
<p>Open the SQL 2008 Solution using Visual Studio 2019.</p> <p>Open the folder: C:\SQLHACK_\LABS\Part 2 – SSIS Migration\SSISDW</p> <p>Right click the SSISDW.sln solution file Open with Visual Studio 2019</p>		<p>You will need to RDP onto the TEAM virtual machine to complete this task.</p> <p>For connection details see APPENDIX - TEAMXX VM RDP details</p>

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.

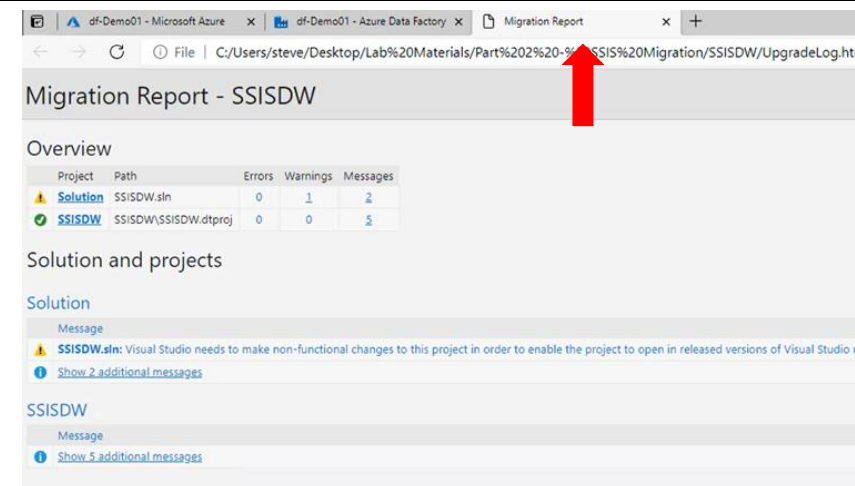


If you make a mistake there is a BACKUP folder which can be used to copy the sln and project files.

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.

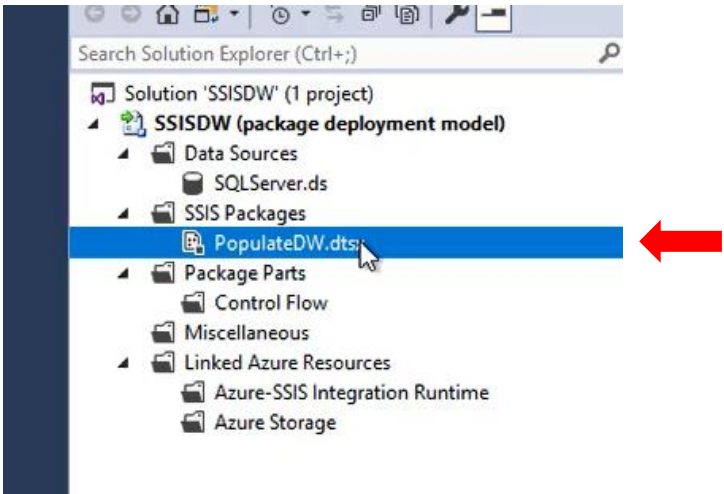


If you make a mistake there is a BACKUP folder which can be used to copy the sln and project files.

Exit the process, replace the files and restart at Step 1.

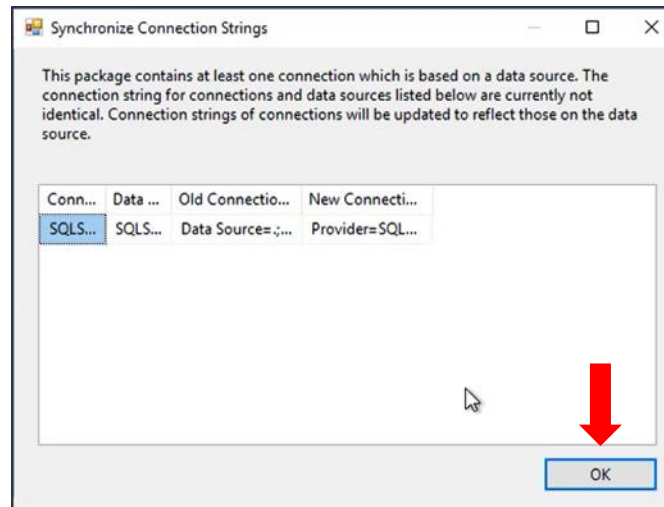
Stage 2 – Convert to Project Deployment mode & update connection string.

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 2019.

Narrative	Screenshot	Notes
<p>Now the Solution is upgraded, it will be open in Visual Studio 2019.</p> <p>In Solution Explorer:</p> <p>Double Click PopulateDW.dtsx to open it.</p>		<p>If Visual Studio 2019 is not open, please confirm Stage 1 has been completed:</p> <p>Open the folder: C:_SQLHACK_\LABS\Part 2 – SSIS Migration\SSISDW</p> <p>Right click the SSISDW.sln solution file Open with Visual Studio 2019</p>

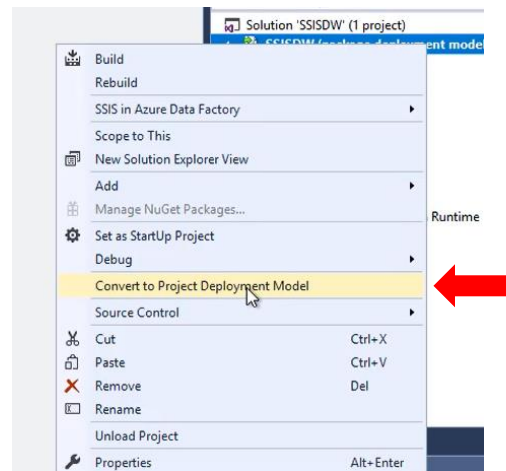
You will be prompted to Synchronise connection strings:

Click OK to acknowledge connection strings will be updated.



The SSIS package will require conversion to a Project Deployment Model.

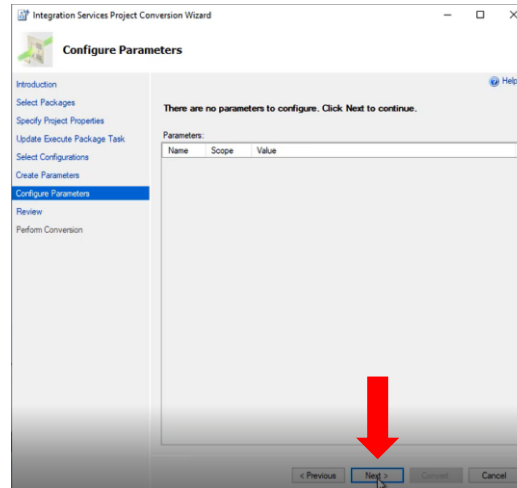
- **Right Click SSISDW** (package deployment model)
- **Select Convert to Project Deployment Model**



The Project Conversion Wizard will begin.

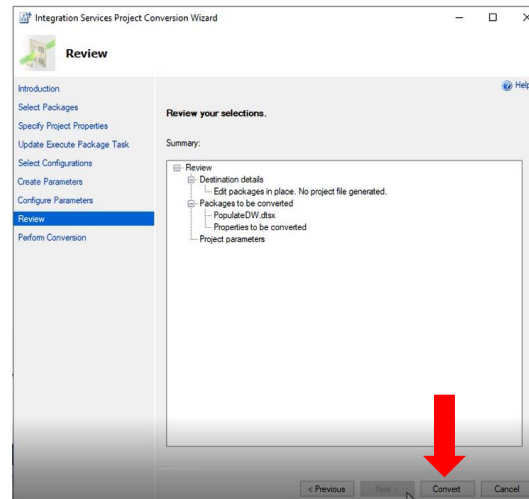
Accept all the defaults on each page
(until the review tab):

- Click Next
- Click Next
- Click Next
- Click Next
- Click Next
- Click Next

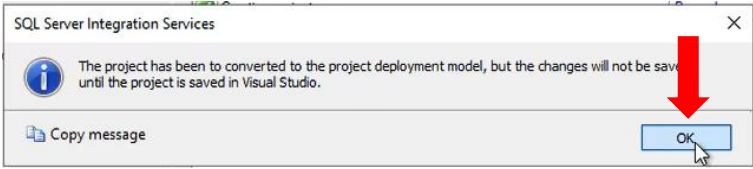
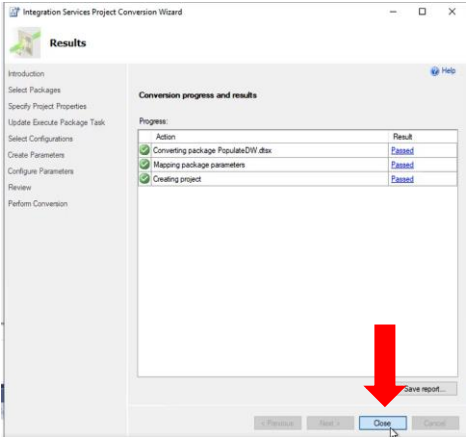
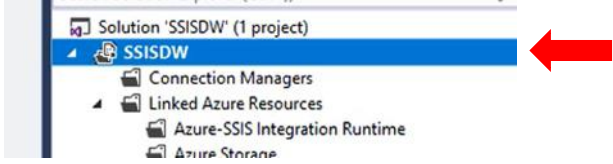
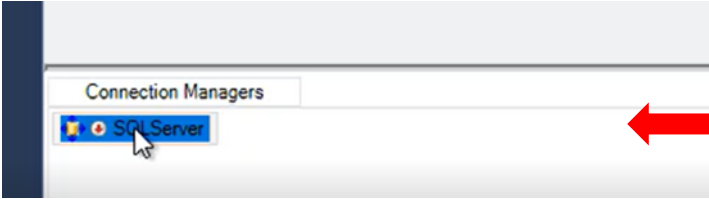


The Project Conversion Wizard is ready.

Click **Convert** to confirm your selections
and start conversion.



Conversion should take a few
moments.

<p>The Project Conversion is complete.</p> <p>Click Ok to Acknowledge the SQL Server Integration Services pop-up.</p>		
<p>The Project Conversion is complete.</p> <p>Click Close and Save the package.</p>		
<p>Verify the Project Conversion has completed successfully.</p> <p>Check the Package name no longer says, "Package Deployment Model."</p>		
<p>Correct the Connection Managers.</p> <p>Double Click the SQL Server Connection in the Connection Managers.</p>		

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\Microsoft OLE DB Driver for SQL Server

OLE DB Provider: Microsoft OLE DB Driver for SQL Server

Enter a server or file name

Server or file name: sqlhackmi-pqasp4yfdhumc.f94a50e692c1.database.windows.net

Location:

Log on to the server

☐ Use Windows NT Integrated Security

☒ Use a specific user name and password:

User name: DemoUser

Password:

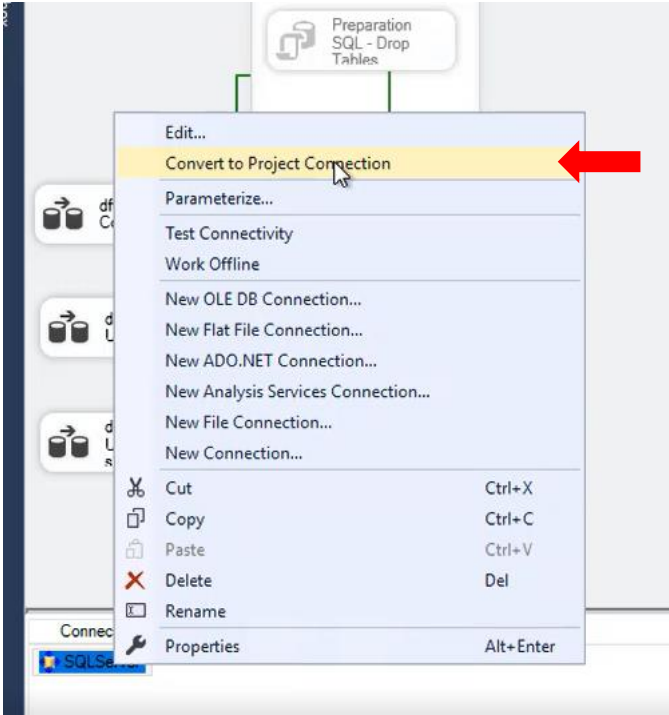
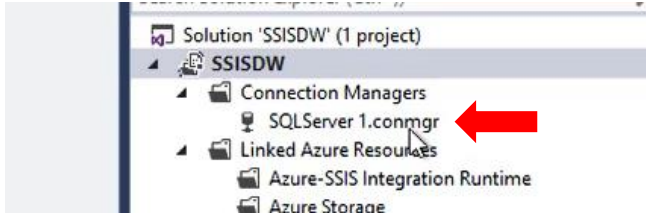
☐ Blank password

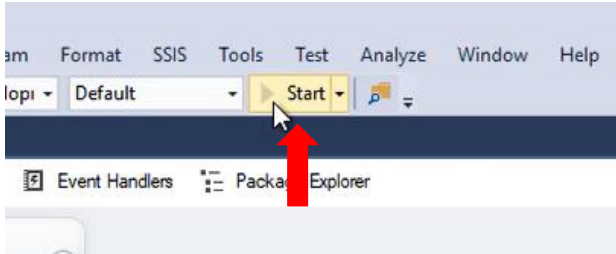
☐ Allow saving password

Initial catalog: 2008DW

Test Connection OK Cancel Help

Please See Appendix - Target SQL Server within this document for full details on the connection settings.

<p>Convert the connection to a project connection.</p> <ul style="list-style-type: none"> • Right Click Connection. • Click Convert to Project Connection. 		
<p>Verify you now have a project connection.</p> <p>Check Connection Manager has the SQL Server 1.conmgr.</p>		

<p>Test the package with the new connection manager.</p> <p>From the Command bar, select Start to Test the package.</p>	 <p>The screenshot shows the SSIS Command Bar with the 'Start' button highlighted. A red arrow points to the 'Start' button. The Command Bar includes tabs for 'Format', 'SSIS', 'Tools', 'Test', 'Analyze', 'Window', and 'Help'. Below the tabs, there is a 'Default' dropdown menu and a 'Start' button. The 'Start' button is highlighted with a red arrow. Below the Command Bar, there are icons for 'Event Handlers' and 'Package Explorer'.</p>	
---	---	--

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

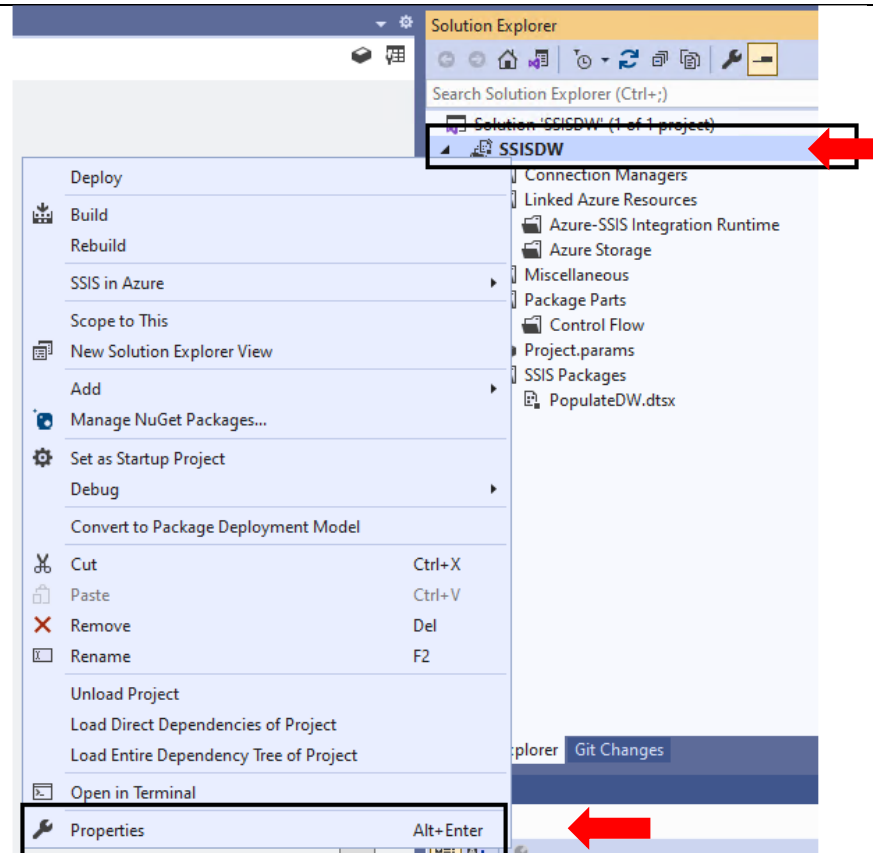
In this section we will be deploying the fixed package onto the SSIS integration runtime and SSISDB held within the Managed Instance.

Narrative	Screenshot	Notes
-----------	------------	-------

Set the Target Server version to SQL Server 2017 as SQL server 2022 is not yet supported.

In Solution Explorer:

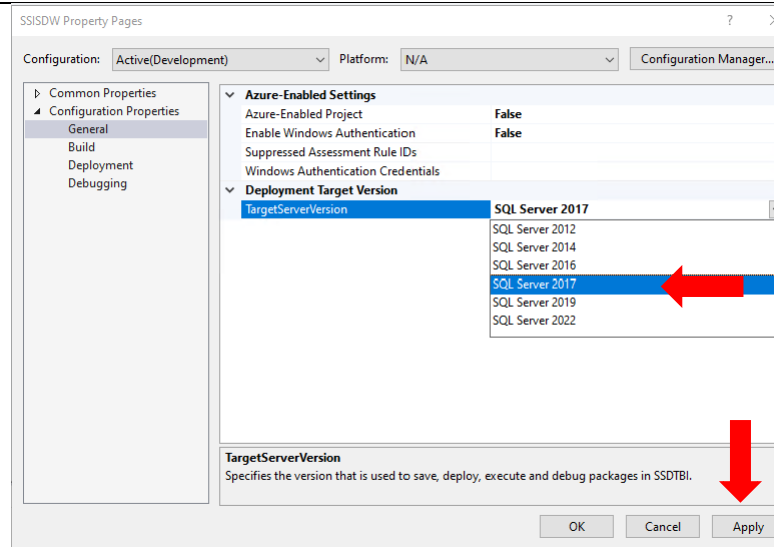
Right Click SSISDW package and select properties.



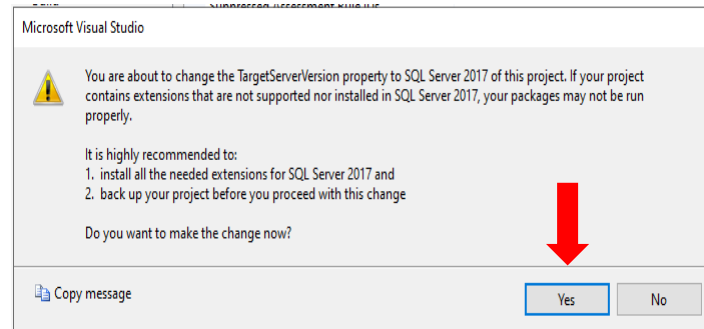
Please ensure you have completed Stage 1 and Stage 2 successfully.

In the SSISDW property page.

- Select “General”
- Change “TargetServerVersion” to **SQL Server 2017**
- Click Apply

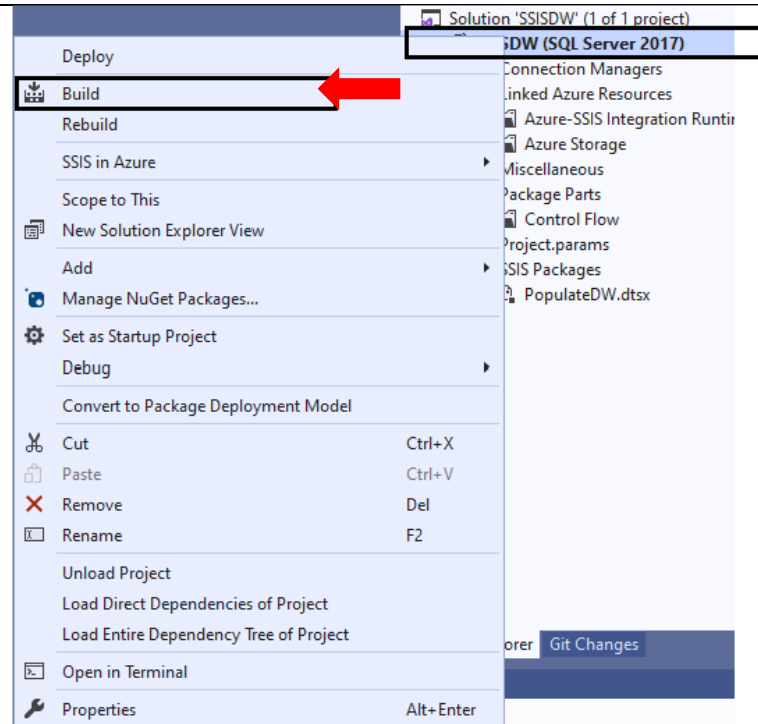


Click **Yes** to complete setting the SQL Target Version.



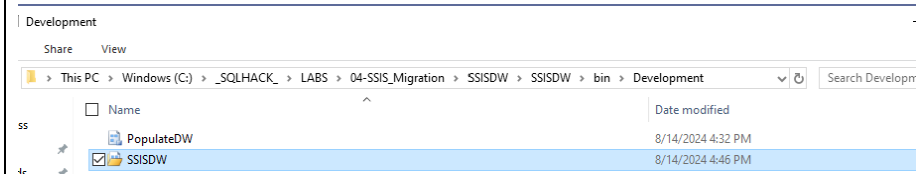
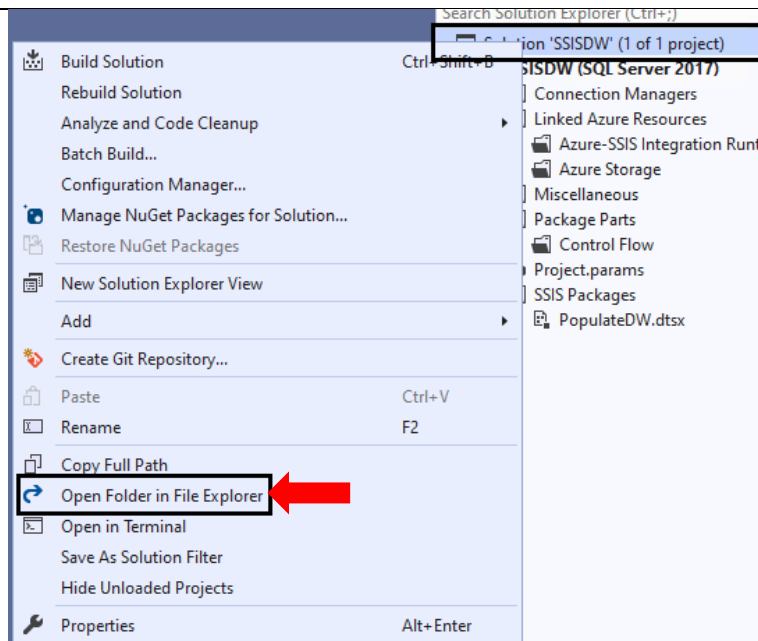
We now need to build the solution before deploy.

Right Click SSISDW package solution and select Build.



Go to Project solution File Explorer and

look for File Path as
'C:_SQLHACK_\LABS\04-
SSIS_Migration\SSISDW\SSISDW\bin\De
velopment'

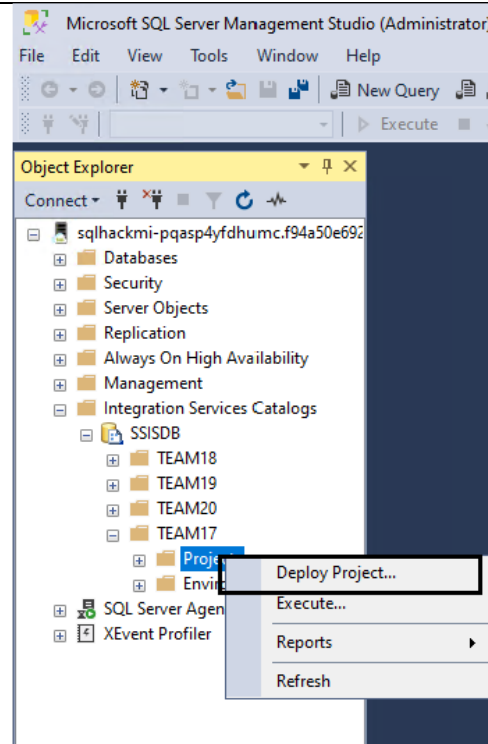


Using SQL Server Management Studio (SSMS), connect to the SQL Server Managed Instance.

Server Name: (See Appendix - Target SQL Server)

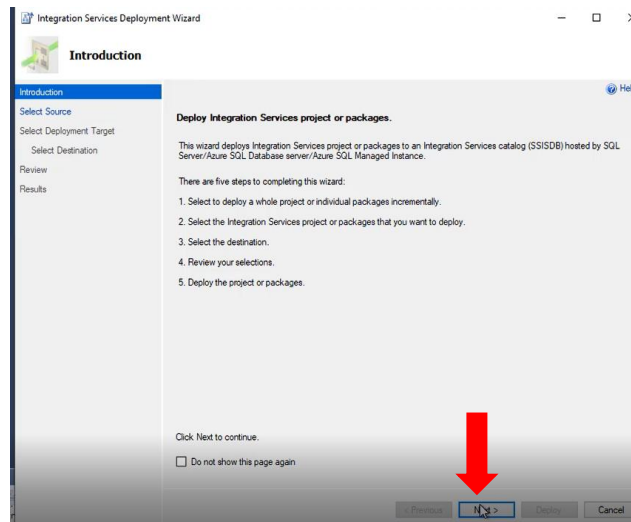
- **Authentication:** SQL Server Authentication
- **User Name:** (See Appendix - Target SQL Server)
- **Password:** (See Appendix - Target SQL Server)

In SSMS, navigate to Integration Service Catalogs SSISB. Create Folder as TEAM folder and Right Click on Projects



Integration Services Deployment Wizard will be started.

Click **Next** to acknowledge introduction.

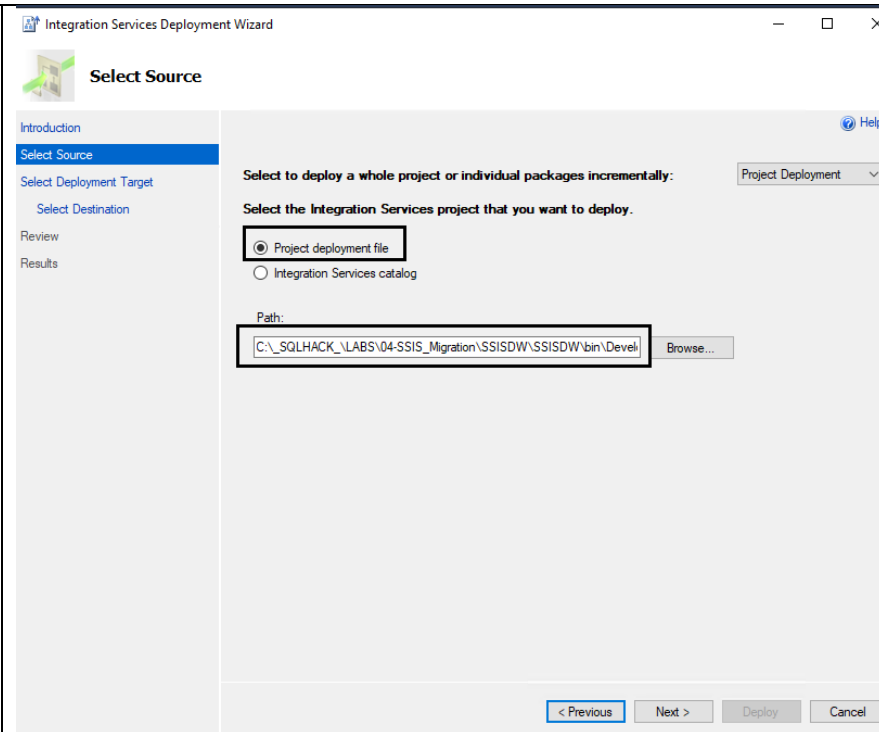


Choose Project Deployment File

And provide ISPAC file Path as

'C:_SQLHACK_\LABS\04-SSIS_Migration\SSISDW\SSISDW\bin\Development'

Click on Next

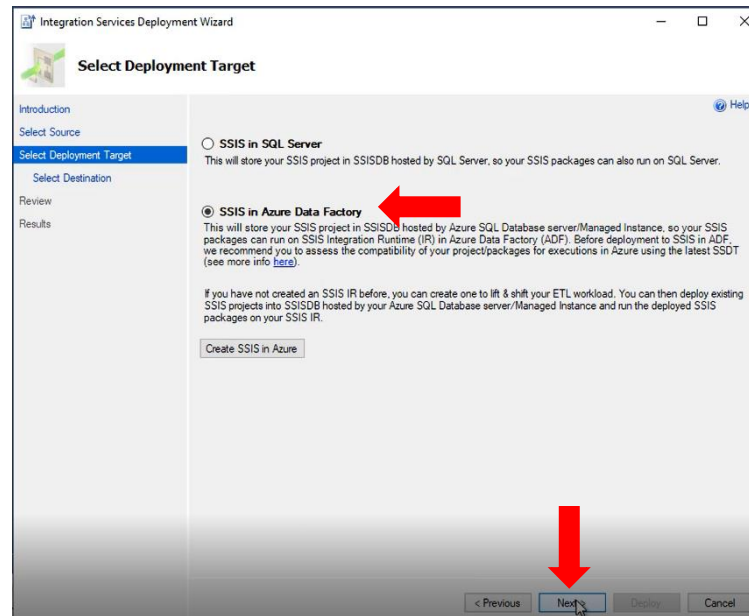


The screenshot shows the 'Integration Services Deployment Wizard' window, specifically the 'Select Source' step. The left-hand navigation pane includes links for 'Introduction', 'Select Source' (which is highlighted), 'Select Deployment Target', 'Select Destination', 'Review', and 'Results'. The main content area has the title 'Select Source' and a 'Help' icon. It contains the following text and controls:

- 'Select to deploy a whole project or individual packages incrementally:' followed by a dropdown menu set to 'Project Deployment'.
- 'Select the Integration Services project that you want to deploy.'
- Two radio buttons: 'Project deployment file' (which is selected) and 'Integration Services catalog'.
- A 'Path:' label followed by a text box containing the path 'C:_SQLHACK_\LABS\04-SSIS_Migration\SSISDW\SSISDW\bin\Devel' and a 'Browse...' button.
- At the bottom, there are four buttons: '< Previous' (highlighted with a blue border), 'Next >', 'Deploy', and 'Cancel'.

Integration Services Deployment Wizard.

- **Select SSIS in Azure Data Factory.**
- **Click Next.**



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.

The screenshot shows the 'Integration Services Deployment Wizard' window, specifically the 'Select Destination' tab. The wizard is titled 'Enter the destination server name and where the project will be located in the Integration Services catalog.' The left sidebar shows the progression: Introduction, Select Source, Select Deployment Target, Select Destination (current), Review, and Results. The main area contains the following fields and controls:

- Server name:** A text box containing 'sqlhackmi-pqasp4yfdhunc:f94a50e692c1.database.windows.net' with a 'Browse...' button to its right. A red arrow points to this field.
- Authentication:** A dropdown menu set to 'SQL Server Authentication' with a 'Connect' button to its right. A red arrow points to this dropdown.
- Login:** A text box containing 'DemoUser' with a red arrow pointing to it.
- Password:** A text box with masked characters (dots) with a red arrow pointing to it.
- Path:** A text box containing '/SSISDB/TEAM17/SSISDW' with a 'Browse...' button to its right. A red arrow points to this field.

At the bottom of the wizard are navigation buttons: '< Previous', 'Next >', 'Deploy', and 'Cancel'.

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.

The screenshot shows the 'Select Destination' dialog box. The left sidebar has 'Select Destination' highlighted. The main area contains the following fields:

- Server name: demomihack02.a9b39c3422a8.database.windows.net
- Authentication: SQL Server Authentication
- Login: steve
- Password: (masked with asterisks)
- Path: /SSISDB/Team19/SSISDW

Buttons include 'Browse...' for Server name and Path, and 'Connect' for Authentication. At the bottom, there are '< Previous', 'Next >', 'Deploy', and 'Cancel' buttons. A red arrow points to the 'Next >' button.

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.

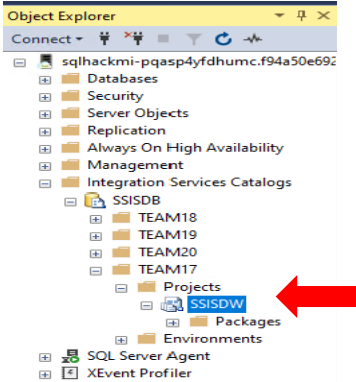
The screenshot shows the 'Review' dialog box. The left sidebar has 'Review' highlighted. The main area contains a 'Summary' section with the following details:

- Command line: /Silent /ModelType:Project /SourcePath:"C:\Users\steve\Desktop\Lab Materials\Part 2 - SSIS Migration\SSISDW\SSISDW\bin\Development\SSIS"
- Source: 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

At the bottom, there are '< Previous', 'Next >', 'Deploy', and 'Cancel' buttons. A red arrow points to the 'Deploy' button.

Stage 4 – Verify Deployment and test run package.

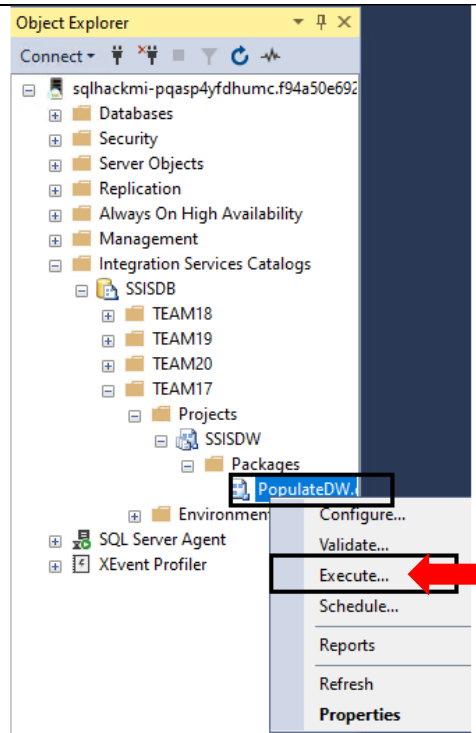
In this section we will be verifying the package has been deployed successfully to the Managed Instance and running the Package to ensure it is working correctly.

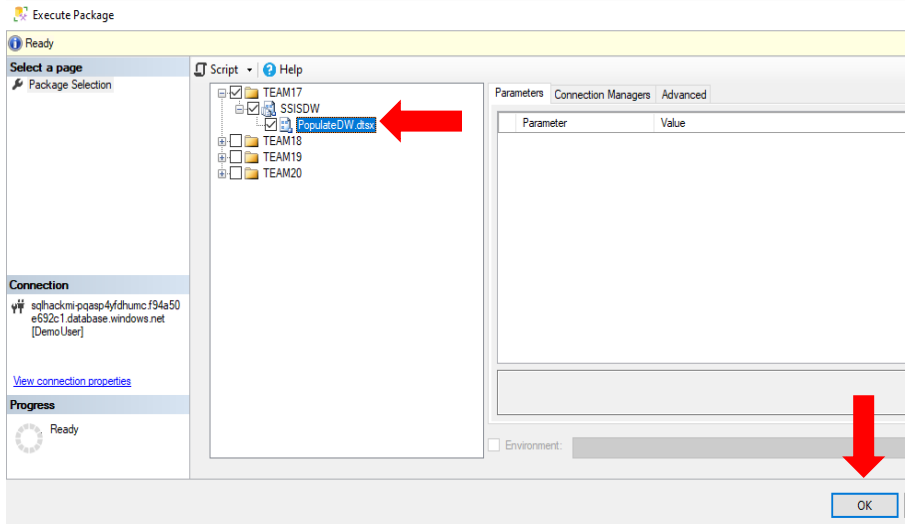
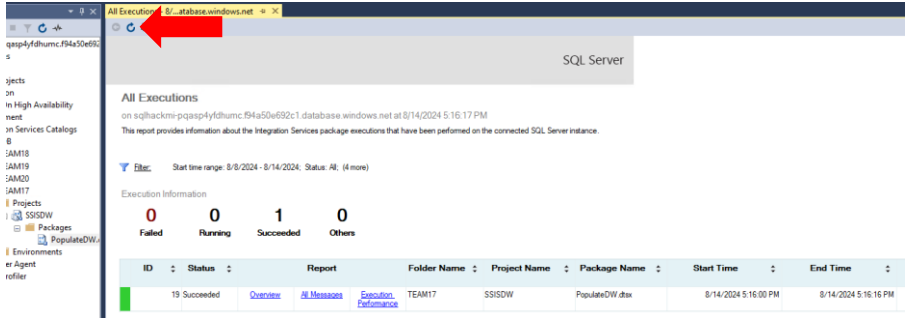
Narrative	Screenshot	Notes
<p>Using SQL Server Management Studio (SSMS), connect to the SQL Server Managed Instance.</p> <p>In SSMS, navigate to Integration Service Catalogs:</p> <ul style="list-style-type: none"> • Select Projects • Select Your TEAM folder <p>Verify the SSISDW Package has been deployed.</p>	 <p>The screenshot shows the Object Explorer in SQL Server Enterprise Manager. The tree view is expanded to show the 'Integration Services Catalogs' folder. Under 'SSISDB', the 'Projects' folder is expanded, and the 'SSISDW' package is highlighted with a red arrow. The package is located under the 'TEAM17' folder.</p>	<p>For connection details please See Appendix - Target SQL Server.</p>

Test the SISS Package migrate to the Managed Instance.

Execute the SSIS Package

- **Right Click** the **SSISDW** package
- Select **Execute**



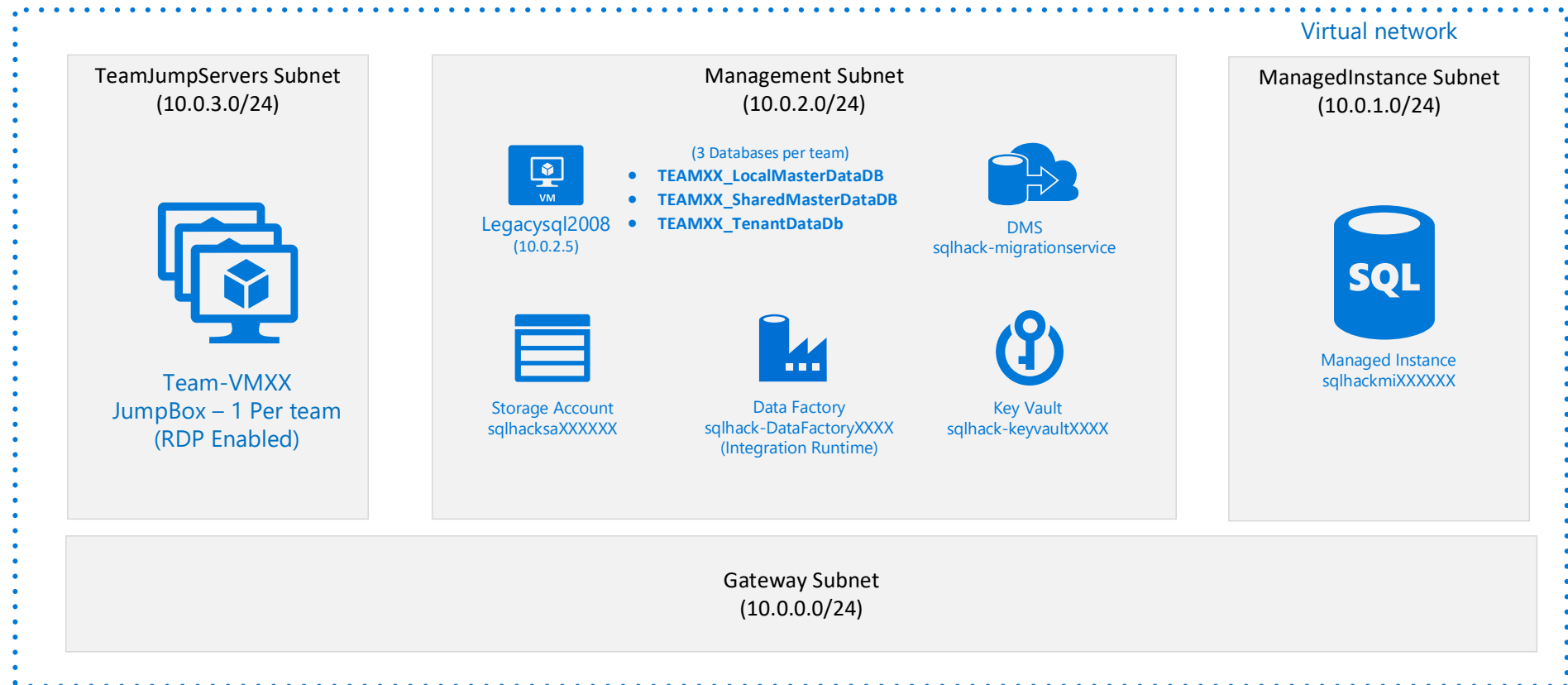
<p>Test the SISS Package migrate to the Managed Instance.</p> <p>Within the Execute Package window:</p> <ul style="list-style-type: none"> Ensure the Package PopulateDW.dtsx is selected. Click OK 		
<p>View the execution report once complete.</p> <p>You be notified that the Selected Packages have been queued to Execute.</p> <ul style="list-style-type: none"> Click Yes to view the Execution Report. <p>Once the Execution report has loaded:</p> <ul style="list-style-type: none"> Click refresh until the package has completed. <p>Congratulations on successfully migrating and upgrading an SSIS package to Azure.</p>		

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.

Optional Stage 6 – How to Deploy Azure SSIS IR Manually

It's currently identified that Azure SSIS IR are not successfully deployed to be use in Lab due to right SQL MI FQDN are not correctly being passed as parameter value in environment deployment script. Deployment script fix in currently back and taken care with next artefacts release. Until then, this step is essential for CSA assigned for deliver to follow following steps and deploy Azure SSIS IR manually if "SSIS Migration topics" in scope for any hack or event.

Please note, without Actively running Azure SSIS IR no participant can deploy their SSIS packages into Azure SQL MI – SSIS catalog. Azure SSIS IR successful deployment can take 15 mins to 60 mins. Hence, its strong recommendation to CSA assigned for delivery should perform the following steps as their mandatory Lab environment setup action items.

a. Go to SQLHACK-SHARED → Data Factory Instance → Launch Studio

The screenshot shows the Microsoft Azure portal interface. The top navigation bar includes the Microsoft Azure logo, a search bar, and a Copilot button. The main content area displays the breadcrumb path: Home > SQLHACK-SHARED > sqlhack-datafactory-pqasp4yfdhumc. The resource name 'sqlhack-datafactory-pqasp4yfdhumc' is highlighted with a red box and labeled '2'. Below the breadcrumb, the 'Launch studio' button is highlighted with a red box and labeled '3'. The left sidebar shows the 'Overview' tab selected, with a list of navigation options including Activity log, Access control (IAM), Tags, Diagnose and solve problems, Settings, Getting started, Monitoring, Automation, and Help. The right sidebar shows the 'Essentials' section with details about the resource group, status, location, subscription, and subscription ID.

b. In ADF Studio , Go to Manage Tab → Integration Runtimes →Click on New →Select Azure – SSIS →Click on Continue

SQL Modernisation Hack – SSIS Migration Lab

Microsoft Azure | Data Factory | sqlhack-datafactory-pqasp4yfdhunc

If you've configured UDRs between Azure Batch and your SSIS IR inside a VNet, please update them to add new IP ranges as recommended [here](#).

Validate all Publish all

General

Factory settings

Connections

Linked services

Integration runtimes

Microsoft Purview

Source control

Git configuration

ARM template

Author

Triggers

Global parameters

Data flow libraries

Security

Credentials

Customer managed key

Outbound rules

Managed private endpoints

Workflow orchestration manager

Apache Airflow

Integration runtimes

The integration runtime (IR) is the compute infrastructure to provide the following data integration capabilities across different network environment. [Learn more](#)

+ New Refresh

Filter by name

Showing 1 - 3 of 3 items

Name	Type	Sub-type	Status	Relat
AutoResolveIntegrationRun...	Azure	Public	Running	0
AzureSSISIR	Azure-SSIS	---	Stopped	0
SSISIR	Azure-SSIS	---	Stopped	0

Integration runtime setup

Integration Runtime is the native compute used to execute or dispatch activities. Choose what integration runtime to create based on required capabilities. [Learn more](#)

Azure, Self-Hosted
Perform data flows, data movement and dispatch activities to external compute.

Azure-SSIS
Lift-and-shift existing SSIS packages to execute in Azure.

Airflow (Preview)
Use this for running your existing DAGs

Continue Cancel

c. Give some valid , Name → Click on Continue

SQL Modernisation Hack – SSIS Migration Lab

Integration runtime setup

General settings

Name * ⓘ

AzureSSISIRLab 9

Description ⓘ

Type

Azure-SSIS

Location * ⓘ

East US

Node size * ⓘ

D8_v3 (8 Core(s), 32768 MB)

Node number * ⓘ

2

Edition/license * ⓘ

Standard

Save money

Save with a license you already own. Already have a SQL Server license?

Yes No

By selecting "yes", I confirm I have a SQL Server license with Software Assurance to apply this [Azure Hybrid Benefit for SQL Server](#).

Please be aware that the cost estimate for running your Azure-SSIS Integration Runtime is **(2 * US\$ 1.938)/hour = US\$ 3.876/hour**, see [here](#) for current prices.

10

Continue Back Cancel

- d. **Select Valid Azure Lab Subscription → Select Valid SQL MI Server Name: (See Appendix - Target SQL Server) → User Name: (See Appendix - Target SQL Server) → Password: (See Appendix - Target SQL Server) → Click on Continue**

SQL Modernisation Hack – SSIS Migration Lab

Integration runtime setup

Deployment settings

☒ Create SSIS catalog (SSISDB) hosted by Azure SQL Database server/Managed Instance to store your projects/packages/environments/execution logs
(See more info [here](#))

Subscription * ⓘ
Azure Pass - Sponsorship (b4be74f9-f993-4677-8c27-c61bed43b2cb) 11

Location ⓘ
East US

Catalog database server endpoint * ⓘ
sqlhackmi-pqasp4yfdhumc.f94a50e692c1.database.windows.net 12

☐ Use Microsoft Entra authentication with the system managed identity for Data Factory ⓘ
(See how to enable it [here](#))

☐ Use Microsoft Entra authentication with a user-assigned managed identity for Data Factory ⓘ
(See how to enable it [here](#))

Admin username * ⓘ
DemoUser 13

Admin password * ⓘ
***** 14

☐ Use dual standby Azure-SSIS Integration Runtime pair with SSISDB failover ⓘ
(See more info [here](#))

Since your Azure SQL Managed Instance is joined to a VNet, your Azure-SSIS Integration Runtime must join the same VNet, but in a different subnet in the next step.

☐ Create package stores to manage your packages that are deployed into file system/Azure Files/SQL Server database (MSDB) hosted by Azure SQL Managed Instance ⓘ
(See more info [here](#))

Continue Back Cancel 15

- e. Make sure valid Azure Lab scripction selected → Select Vnet Name as “SQLHACK-SHARED-vnet” → Subnet Name as “Management” → Vnet Injection Method as “Standard” → Click on “Vnet validation” → Click on Continue

SQL Modernisation Hack – SSIS Migration Lab

Edit integration runtime

4

☐ Customize your Azure-SSIS Integration Runtime with additional system configurations/component installations (See more info [here](#)) ⓘ

☒ Select a VNet for your Azure-SSIS Integration Runtime to join, allow Data Factory to create certain network resources, and optionally bring your own static public IP addresses (See more info [here](#)) ⓘ

Subscription * ⓘ

Azure Pass - Sponsorship (b4be74f9-f993-4677-8c27-c61bed43b2cb) ▼

Location * ⓘ

East US ▼

Type * ⓘ

Azure Resource Manager Virtual Network ▼

VNet name * ⓘ

SQLHACK-SHARED-vnet ▼ ⓘ 17

[Create new](#)

Subnet name * ⓘ

Management ▼ 18

VNet injection method * ⓘ

Standard ▼ 19

☐ Bring static public IP addresses for your Azure-SSIS Integration Runtime ⓘ (See more info [here](#))

⚠ VNet setting warning:

- Please ensure that your Azure-SSIS Integration Runtime in the selected VNet/subnet can access SSISDB hosted by the selected Azure SQL Database server with VNet service endpoints/firewall rules/private endpoint.

☐ Set up Self-Hosted Integration Runtime as a proxy for your Azure-SSIS Integration Runtime ⓘ (See more info [here](#))

20

21

Continue Back **VNet validation** Cancel

f. And, finally under summary page click on “Create”

SQL Modernisation Hack – SSIS Migration Lab

Integration runtime setup

Summary

Your Azure-SSIS Integration Runtime (IR) is created with the following settings:

Azure Data Factory Settings

- **Subscription:** b4be74f9-f993-4677-8c27-c61bed43b2cb
- **Resource group:** SQLHACK-SHARED
- **Name:** sqlhack-datafactory-pqasp4yfdhumc
- **Location:** eastus

General settings

- **Name:** integrationRuntime1
- **Location:** East US
- **Node size:** Standard_D8_v3
- **Node number:** 2
- **Edition:** Standard
- **Azure Hybrid Benefit:** LicenseIncluded

Deployment settings

- **Catalog database server endpoint:** sqlhackmi-pqasp4yfdhumc.f94a50e692c1.database.windows.net
- **Catalog database server location:** East US

Advanced settings

- **Maximum parallel executions per node:** 8
- **VNet name:** SQLHACK-SHARED-vnet
- **Subnet name:** Management
- **VNet injection method:** Standard

If you want to change any of the above settings, click **Previous** to do so.

Once your Azure-SSIS IR is running, you can execute your packages on it after [deploying](#) them into your file system/Azure Files/SSISDB hosted by **sqlhackmi-pqasp4yfdhumc.f94a50e692c1.database.windows.net**.

Please be aware that the cost estimate for running your Azure-SSIS Integration Runtime is **(2 * US\$ 1.938)/hour = US\$ 3.876/hour**, see [here](#) for current prices.

To manage the running cost of your Azure-SSIS IR, you can [stop & restart](#) it whenever convenient or [schedule](#) it just in time.

22

Create

Previous

Cancel

g. It takes from 15 to 60 minutes to Azure SSIS IR to show as running status.

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	<i>SQL MI FQDN from Azure portal</i>
Resource Group	SQLHACK-SHARED
Sysadmin Login Name: (Use for Migrations)	DemoUser
Admin Login Password:	Demo@pass1234567