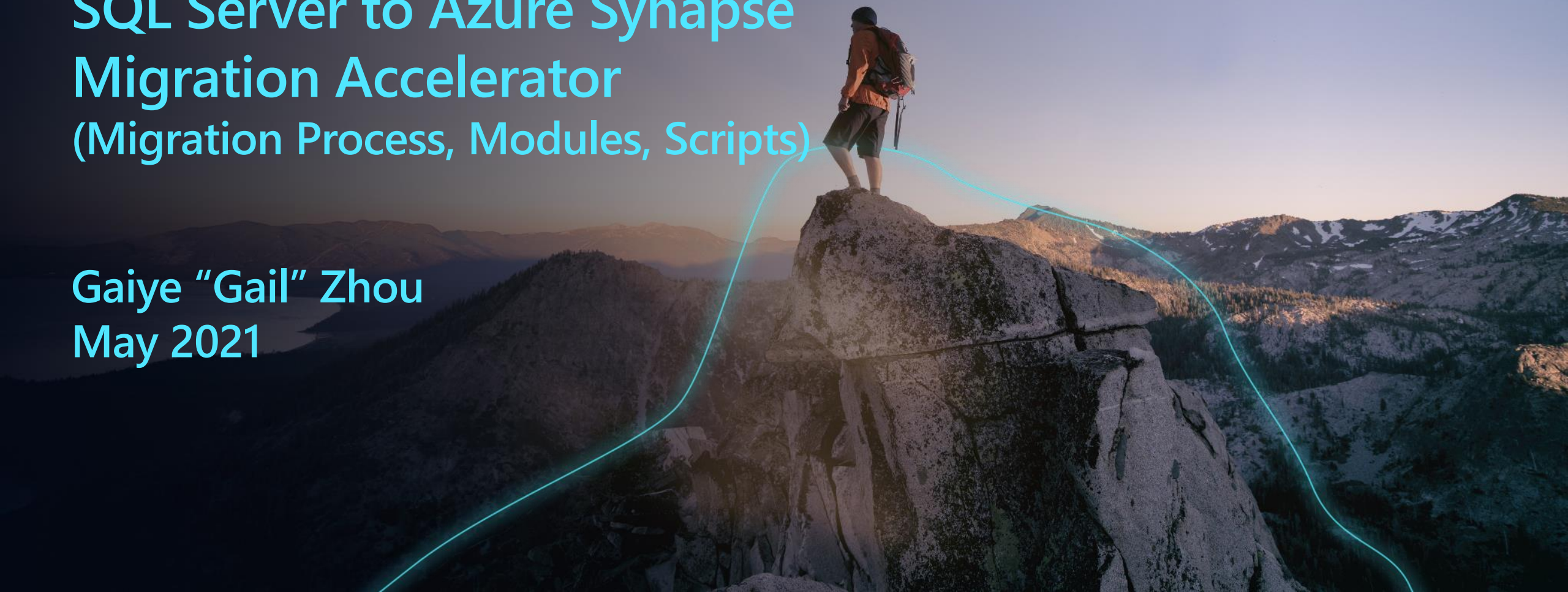
DATA & AI

Microsoft Azure Solutions Business

AAAP
Analytics & AI Accelerate Program

SQL Server to Azure Synapse Migration Accelerator
(Migration Process, Modules, Scripts)

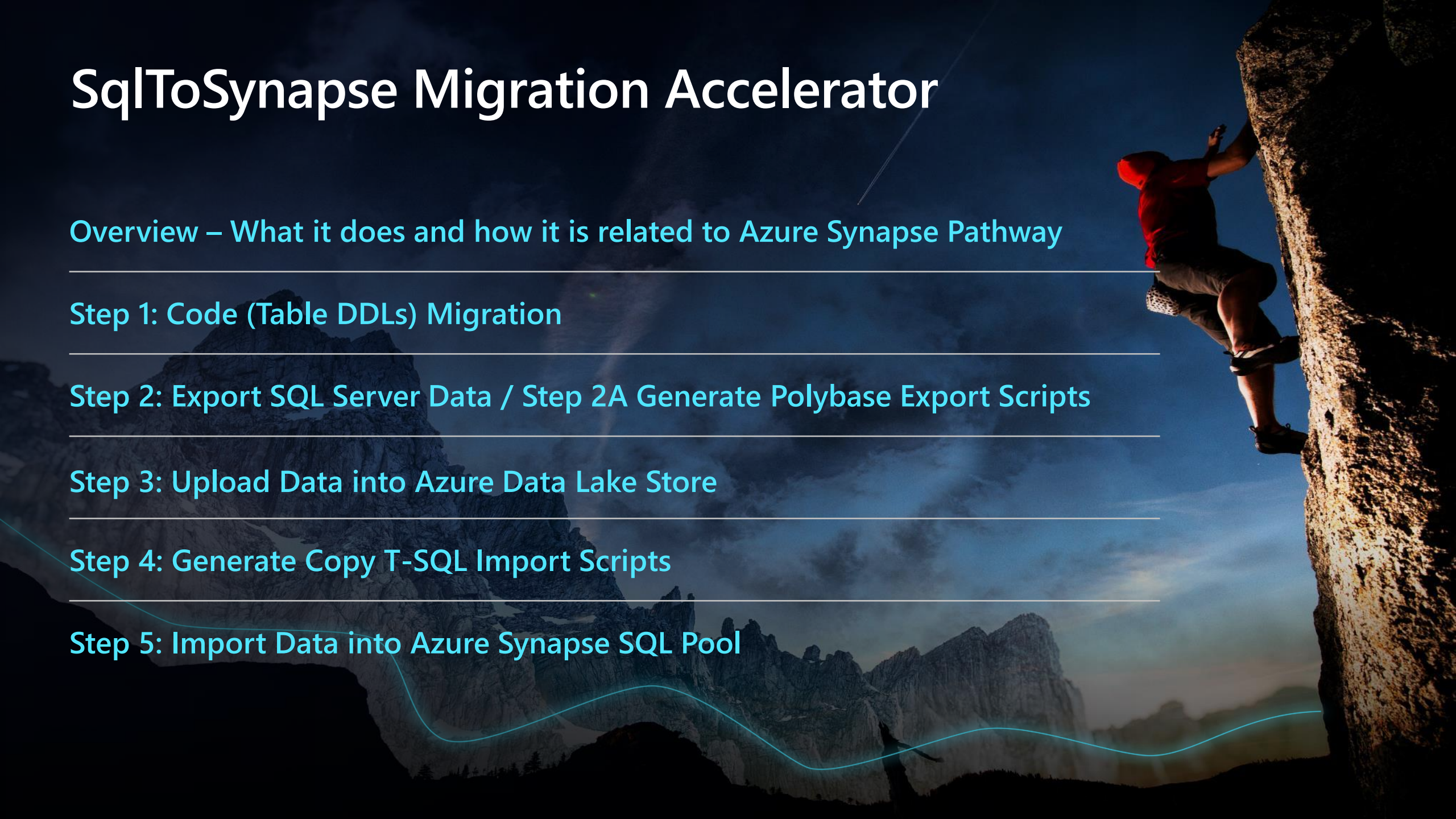


Gaiye “Gail” Zhou
May 2021

Disclaimer

Theis document was developed in consultation and collaboration with Microsoft Corporation technical architects. Because Microsoft must respond to changing market conditions, this document should not be interpreted as an invitation to contract or a commitment on the part of Microsoft. Microsoft has provided high-level guidance in this document with the understanding that MICROSOFT MAKES NO WARRANTIES, EXPRESS OR IMPLIED, WITH RESPECT TO THE INFORMATION CONTAINED HEREIN. This document is provided “as-is”. Information and views expressed in this document, including URL and other Internet Web site references, may change without notice. Some examples depicted herein are provided for illustration only and are fictitious. No real association or connection is intended or should be inferred. This document does not provide you with any legal rights to any intellectual property in any Microsoft product. You may copy and use this document for your internal, reference purposes.

SqlToSynapse Migration Accelerator



Overview – What it does and how it is related to Azure Synapse Pathway

Step 1: Code (Table DDLs) Migration

Step 2: Export SQL Server Data / Step 2A Generate Polybase Export Scripts

Step 3: Upload Data into Azure Data Lake Store

Step 4: Generate Copy T-SQL Import Scripts

Step 5: Import Data into Azure Synapse SQL Pool

SqlToSynapse Migration Accelerator (PowerShell Scripts)

What does the Scripts do?

- ✓ Translate SQL Server Table DDLs into Azure Synapse DDLs
- ✓ Execute Translated Table DDLs in Azure Synapse– to migrate Tables DDLs into Azure Synapse
- ✓ Genertate Polybase Export T-SQL Scripts
- ✓ Execute Polybase Export T-SQL Scripts – to export data directly into Azure Storage from SQL Server
- ✓ Upload Exported Data into Azure Data Lake Store (Blob Storge) if using BCP export
- ✓ Generate T-SQL Copy Scripts
- ✓ Execute T-SQL Copy Scripts - to Import Data into Azure Synapse from Azure Storage

Complementary to Azure Synapse Pathway

Why do we need these Scripts when we already have Azure Synapse Pathway?

These Scripts are complementary to Azure Synapse Pathway (ASP). ASP does not perform data migration today. We designed and implemented 6 modules to complete the end-to-end tasks of tables migration and data migration (using BCP or Polybase Export). Please check the latest release of Azure Synapse Pathway for more advanced SQL Server code translation capabilities.

You can use scripts described in this document (Module 5_RunSqlFilesInFolder) to execute all translated code by ASP or other methods. Please check the newest release of Azure Synapse Pathway so you can use the best available functions. sql-docs/azure-synapse-pathway-overview.md at live · MicrosoftDocs/sql-docs (github.com)

In addition, Module 3, 4, 5 are reusable for other types of migrations, for example, Netezza or Teradata or Exadata or Oracle to Azure Synapse migrations. After the code is translated, and data is exported out of source systems, the rest of the tasks are the same. Therefore module 3-5 can be utilized for any of those migrations.

What are in the Utilities (For BCP Export)

There are five modules that contain PowerShell Scripts and T-SQL Scripts designed to accomplish key task(s) if you will use BCP Export method.

The five modules are summarized as below:

1_TranslateMetaData: Translate SQL objects (DDLs) from source system format to Azure Synapse format. The output is stored as .sql files in specified output file folder (configurable).

2_ExportSourceData: Export SQL Server Tables into data files stored in predefined structure and format (.csv or .txt)

3_LoadDataIntoAzureStorage: Load exported data files into specified container in Azure Storage (Azure Data Lake Store).

4_GenerateCopyIntoScripts: Generate "COPY Into" T-SQL Scripts that will move data from Azure Storage into Azure Synapse SQL Pool tables, once executed.

5_RunSqlFilesInFolder: Run all T-SQL Scripts defined in .sql files stored in a specified file folder. The T-SQL Scripts can be DDL, DML, Data Movement Scripts (such as Copy Into scripts or Polybase Export Scripts), or any other scripts such as create/update statistics or indexes. In fact, this module is designed to run all T-SQL scripts in a folder (against Azure Synapse or SQL Server).

Note: **5_RunSqlFilesInFolder is reused twice in the process:** (1) Run T-SQL Scripts generated from 1_TranslateMetadata, and (2) Run T-SQL Scripts generated from 4_GenerateCopyIntoScripts.

What are in the Utilities (For Polybase Export)

There are four modules that contain PowerShell Scripts and T-SQL Scripts designed to accomplish key task(s) if you will use Polybase Export method.

The four modules are summarized as below:

1_TranslateMetaData: Translate SQL objects (DDLs) from source system format to Azure Synapse format. The output is stored as .sql files in specified file folder (configurable).

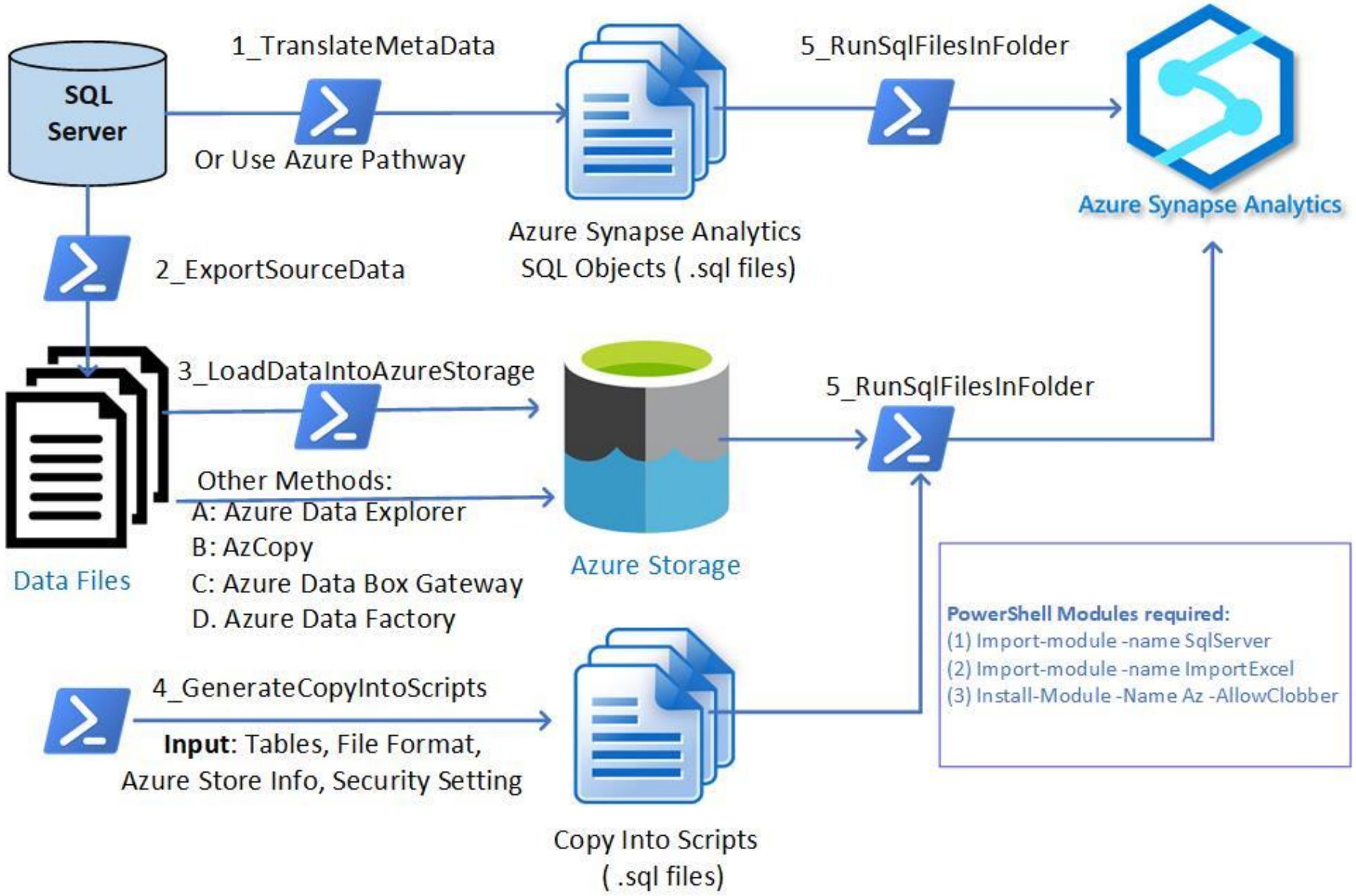
2A_GeneratePolybaseExportScripts: Generate Polybase Export T-SQL Script for each table. Polybase export set up examples are provided in subfolder "Utilities" inside this module.

4_GenerateCopyIntoScripts: Generate "COPY Into" T-SQL Scripts that will move data from Azure Storage into Azure Synapse SQL Pool tables, once executed.

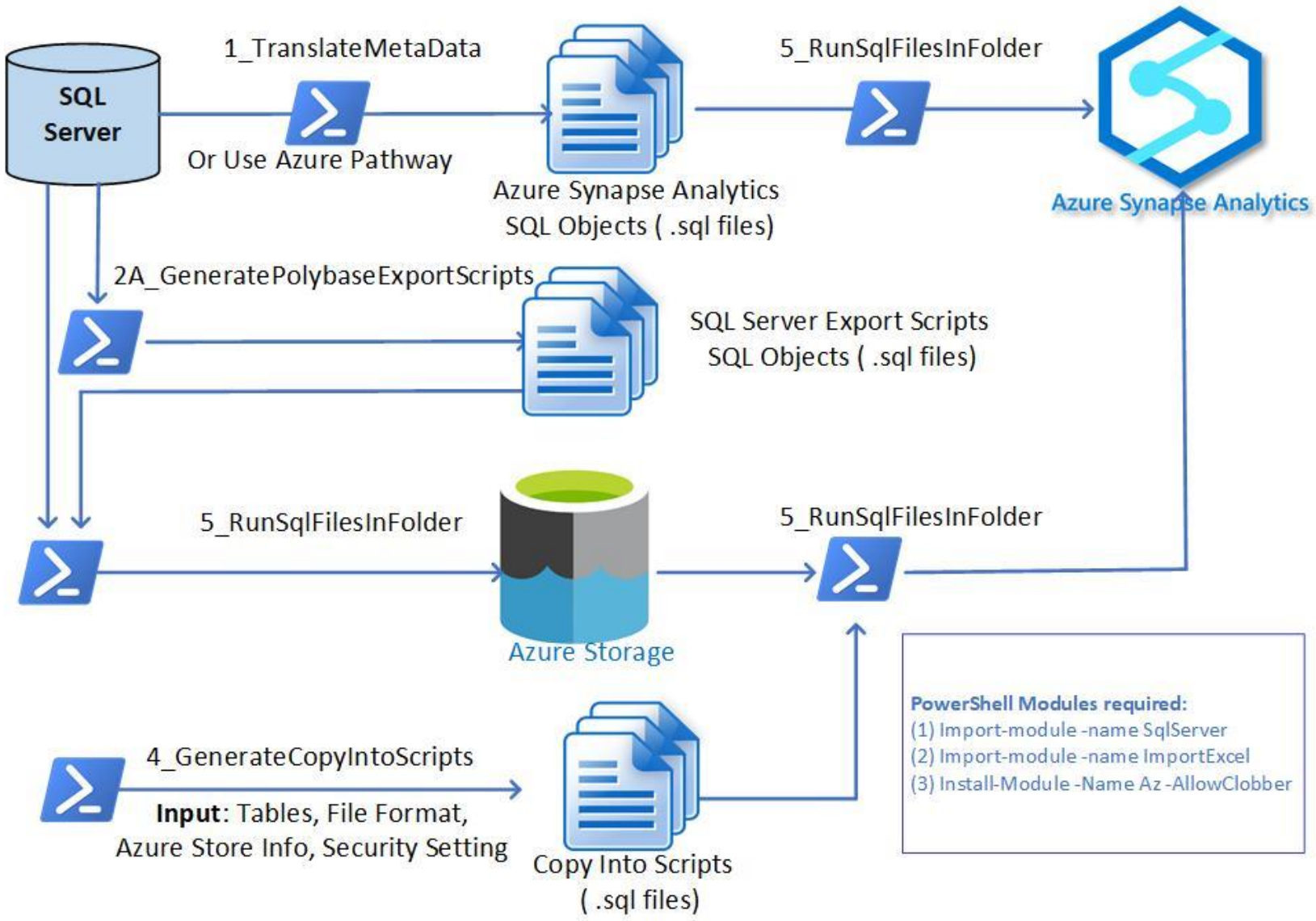
5_RunSqlFilesInFolder: Run all T-SQL Scripts defined in .sql files stored in a specified file folder. The T-SQL Scripts can be DDL, DML, Data Movement Scripts (such as Copy Into scripts or Polybase Export Scripts), or any other scripts such as create/update statistics or indexes. In fact, this module is designed to run all T-SQL scripts in a folder (against Azure Synapse or SQL Server).

Note: **5_RunSqlFilesInFolder is reused three times in the process:** (1) Run T-SQL Scripts generated from 1_TranslateMetadata, (2) Run T-SQL Scripts generated from 2A_GeneratePolybaseExportScripts, and (3) Run T-SQL Scripts generated from 4_GenerateCopyIntoScripts.

SQL Server to Synapse Migration Tasks Carried out by PowerShell Scripts (Modules) – BCP Export – All Versions of SQL Servers



SQL Server to Synapse Migration Tasks Carried out by PowerShell Scripts (Modules) – Polybase Export – SQL Servers 2016+



What Do I need to Use the Utilities?

Choose One of the Environments:

1. Windows PowerShell ISE (preferred)
2. Visual Studio Code (with PowerShell Extension Installed)

PowerShell Modules required:

(1) Import-module -name SqlServer
(2) Import-module -name ImportExcel
(3) Install-Module -Name Az –AllowClobber

PowerShell Permissions (Permissions may be denied if the Scripts are from GitHub or Email):
Use one of the options to set Powershell permissions (examples):

```
Set-ExecutionPolicy Unrestricted -Scope CurrentUser
Unblock-File -Path C:\migratemaster\modules\1_TranslateMetaData\TranslateMetaData.ps1
```

Download and Install AzCopy (Only if you will be using BCP Export Method). This task can be skipped if you will be using other methods to upload data into Azure Storge such as Azure Data Box Gateway or Azure Data Explore).

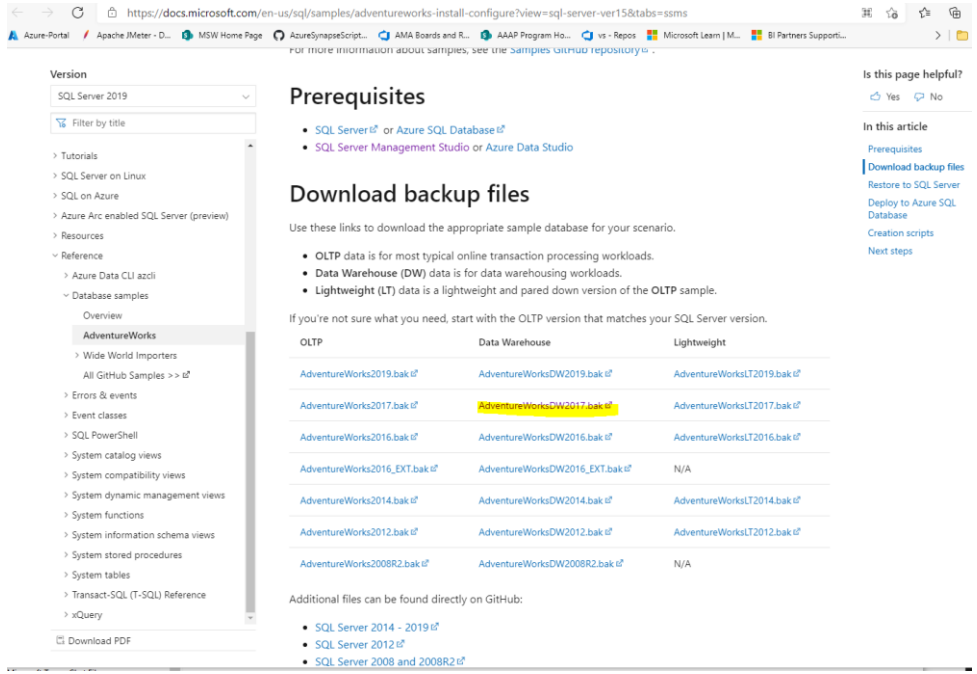
Copy or move data to Azure Storage by using AzCopy v10 | Microsoft Docs

To Test the Scripts with an actual SQL Server & Azure Synapse

(1) Set up your Azure Synapse Workspace with Azure SQL Pool Created

(2) Download **AdventureWorksDW2017.bak** from the link below and restore it to your SQL Server.

[AdventureWorks sample databases - SQL Server | Microsoft Docs]



How to Restore SQL Server Database to .bak file

[Restore a Database Backup Using SSMS - SQL Server | Microsoft Docs]

Download SQL Server 2017 if you do not have a SQL Server already

https://www.microsoft.com/en-us/sql-server/sql-server-2017

Download SSMS if you don't have it already

[Download SQL Server Management Studio (SSMS) - SQL Server Management Studio (SSMS) | Microsoft Docs]

5-Step Migration Process with BCP Export

- You can use this method for all versions of SQL Servers

If you have SQL Server 2016 or later, you have additional option to export SQL server data directly into Azure Storage. See details later in this presentation.

(Polybase is available only for SQL Server 2016 or later)

How many steps?

How to Run?

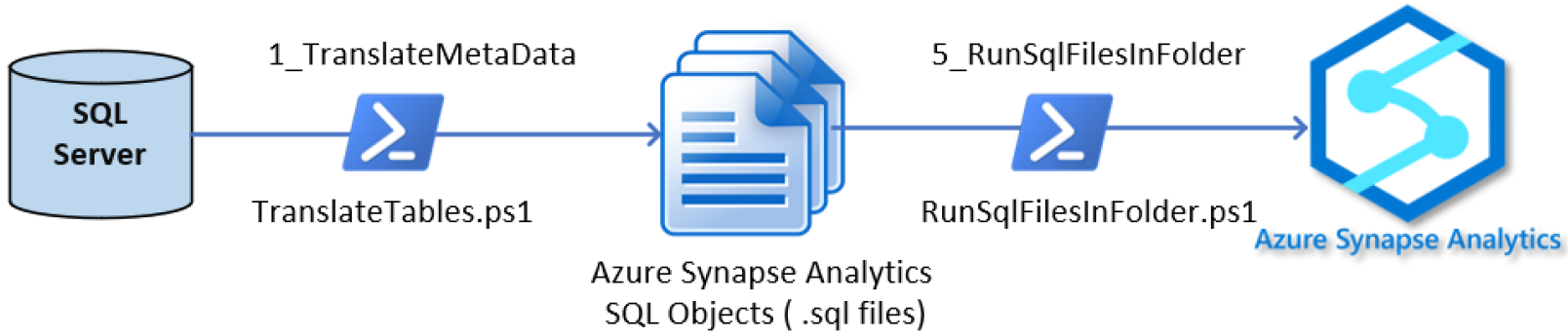
What To Config?

Can I automate?

What Input?

What Output?

Step 1: Code (DDLs) Migration – Translate SQL Server Tables and create them in Azure Synapse



Task 1
Execute PowerShell Scripts "**TranslateTables.ps1**" (Inside folder 1_Translate folder)

Output: Azure Synapse Create Table Statement (DDLs) stored in .sql format.

Config Files needed (Samples are provided):
SourceToTargetTablesConfig.xlsx
translate_config.json

Note 1: Need to access SQL Server for this. "db_datareader" role permission is needed

Note 2: Look for T-SQL Scripts "**GenerateSourceToTargetConfig.sql**" in the Utilities Subfolder to create starter SourceToTargetTablesConfig.xlsx.

Task 2
Execute PowerShell Scripts"**RunSqlFilesInFolder.ps1**" (Inside folder 5_RunSqlFilsInFolder)

Input: Azure Synapse Table DDL files (.sql) stored in one file folder, which were generated by Task 1.

Output: Timestamped log files in the "Log" subfolder where you run this PowerShell Scripts.

Results: Tables will be created in Azure Synapse Dedicated SQL Pool.

Config File(s) needed (Samples are provided):
sql_synapse.json

Note: Need "Create Schema" and "Create Table Permission" in Azure Synapse SQL Pool.

Step 2: Data Migration – Export Data from SQL Server

How many steps?

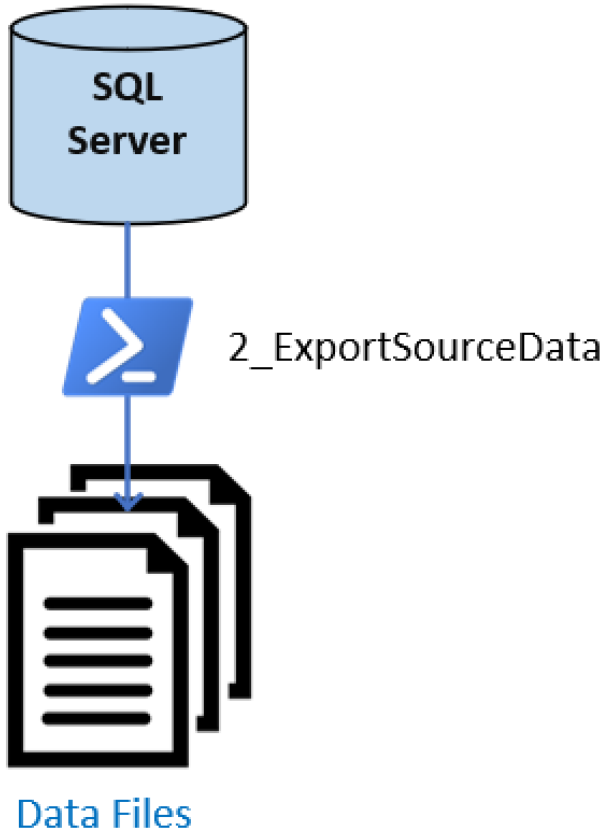
How to Run?

What To Config?

Can I automate?

What Input?

What Output?



Task: Execute PowerShell Scripts "**ExportSourceData.ps1**" (Inside 2_ExportSourceData).

Output: Data files in .csv or .txt format are produced and saved into local storage.

Config Files needed (Sample(s) are provided):
ExportTablesConfig.csv
sql_bcp.json

Note 1: Need to access SQL Server for this. "db_datareader" role permission is needed

Note 2: You need **bcp utility** (**bcp.exe)** for this. If you have SQL server installed, you may find it in this location: C:\Program Files\Microsoft SQL Server\Client SDK\ODBC\130\Tools\Binn

If you are not able to find bcp.exe, you can download a copy from https://docs.microsoft.com/en-us/sql/tools/bcp-utility?view=sql-server-ver15

How many steps?

How to Run?

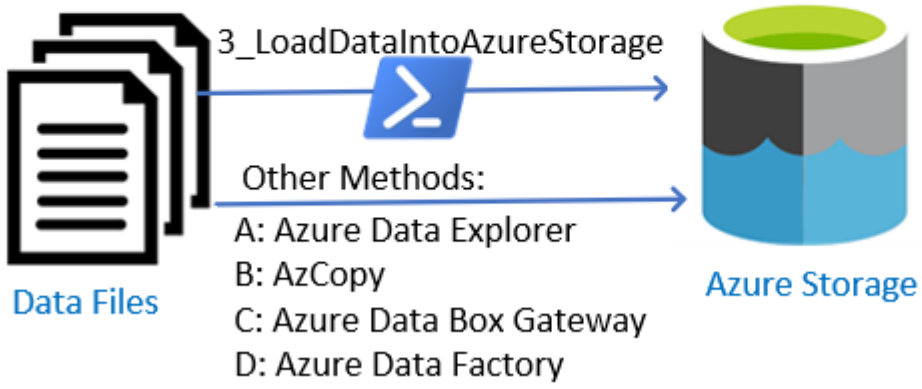
What To Config?

Can I automate?

What Input?

What Output?

Step 3: Upload Exported Data Files into Azure Storage (Azure Data Lake Store or Blob Storage)



Task: (If using 3_LoadDataIntoAzureStorage)

Execute PowerShell Scripts "**LoadDataIntoAzureStorage.ps1**" (inside folder 3_LoadDataIntoAzureStorage)

Output: None

Results: Data in Data Files are uploaded to Azure Storage (Data Lake Store or Blob Storage)

Config File(s) needed (Samples are provided):
sql_bcp.json

Note: The user needs to have the permission of "Storage Blob Data Contributor" or a SAS key with the right permissions is already generated for the user.

How many steps?

How to Run?

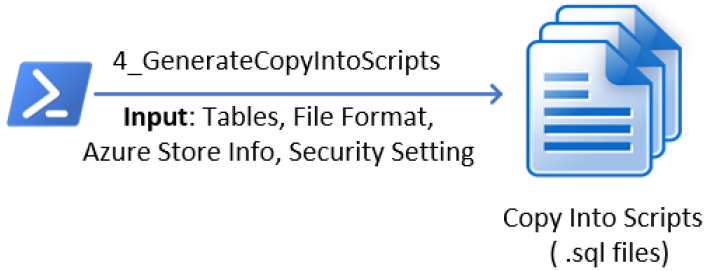
What To Config?

Can I automate?

What Input?

What Output?

Step 4: Generate COPY Into T-SQL Script for Tables (to run in Azure Synapse Dedicated SQL POOL)



Task
Execute PowerShell Scripts "**GenerateCopyIntoScripts.ps1**" (inside folder 4_GenerateCopyIntoScripts)

Output: T-SQL COPY Script for Each Table (in the form of .sql files)

Config Files needed (samples are provided)
(1) csv_mi.json (recommended) or csv_key.json (this requires Storage Account Key, not recommended but can be used for quick testing)
(2) TablesConfig.csv: This is a table list with information needed for each T-SQL Copy script.

Note 1: If using csv_mi.json file, and the Azure Storage was created independently (not as part of the Azure Synapse Workspace Creation), you will need to set up Azure Synapse Workspace as a managed instance, for the Scripts to work. See instruction in top of the "GenerateCopyIntoScripts.ps1". In addition, you can find sample PowerShell script "**SetManagedIdentity.ps1**" (inside subfolder Utilities) to set up Managed Instance.

Note 2: Sample configuration files are provided for data file formats other than CSV: parquet and orc. However, these file formats are not tested.

Step 5: Import Data From Azure Storage into Azure Synapse Dedicated SQL POOL)

How many steps?

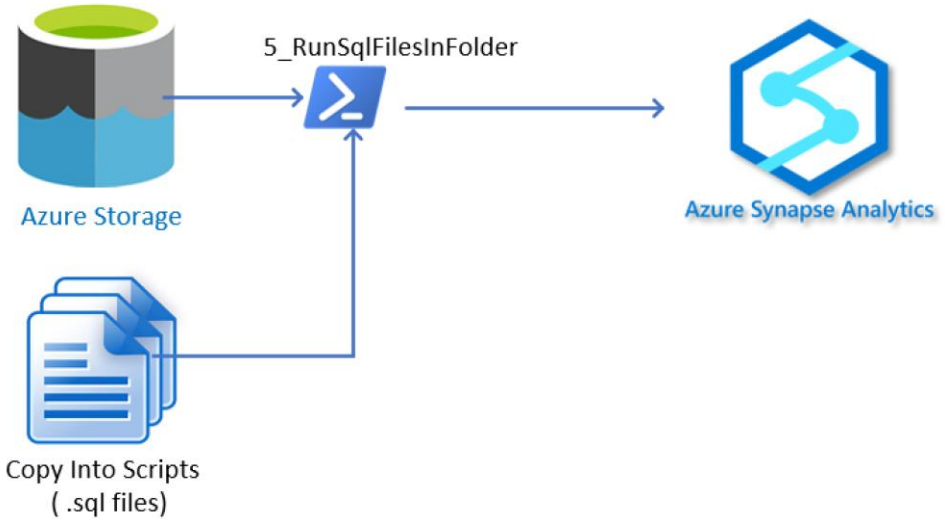
How to Run?

What To Config?

Can I automate?

What Input?

What Output?



Task: Execute PowerShell Scripts"**RunSqlFilesInFolder.ps1**" (Inside folder 5_RunSqlFilsInFolder)

Input: T-SQL script (Copy Into) generated by **GenerateCopyIntoScripts.ps1** (Inside Module 4_GenerateCopyIntoScripts) stored in one file folder.

Output: Timestamped log files in the "Log" subfolder where you run this PowerShell Scripts.

Results: Data is imported to Azure Synapse Tables (Dedicated SQL Pool).

Config File(s) needed: sql_synapse.json or sql_scripts.json.

Note: Need "Write Data" Permission" in Azure Synapse SQL Pool.

5-Step Migration Process with Polybase Export

(Polybase is available only for SQL Server 2016 or later)

Step 1: Code (DDLs) Migration – Translate SQL Server Tables and create them in Azure Synapse

How many steps?

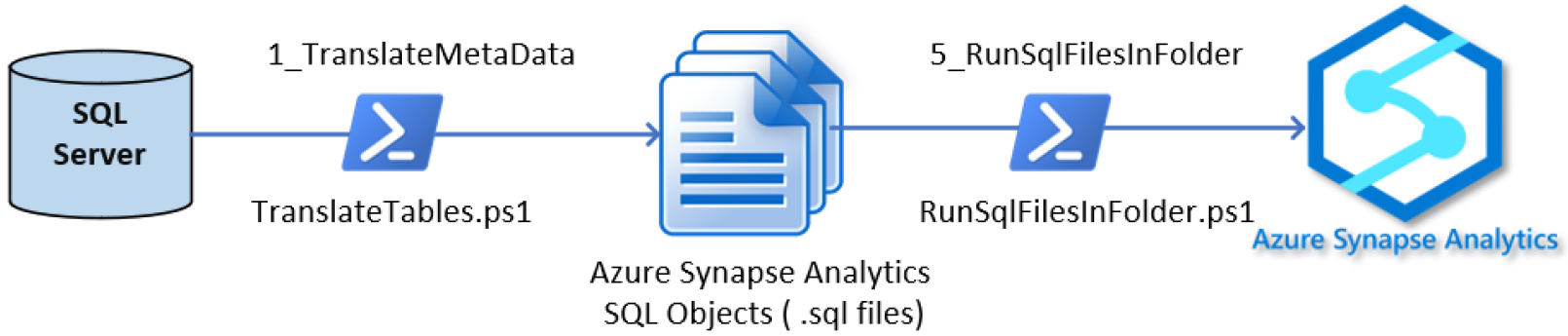
How to Run?

What To Config?

Can I automate?

What Input?

What Output?



Task 1
Execute PowerShell Scripts "**TranslateTables.ps1**" (Inside folder 1_Translate folder)

Output: Azure Synapse Create Table Statement (DDLs) stored in .sql format.

Config Files needed (Samples are provided):
SourceToTargetTablesConfig.xlsx
translate_config.json

Note 1: Need to access SQL Server for this. "db_datareader" role permission is needed

Note 2: Look for T-SQL Scripts "**GenerateSourceToTargetConfig.sql**" in the Utilities Subfolder to create starter SourceToTargetTablesConfig.xlsx.

Task 2
Execute PowerShell Scripts"**RunSqlFilesInFolder.ps1**" (Inside folder 5_RunSqlFilsInFolder)

Input: Azure Synapse Table DDL files (.sql) stored in one file folder, which were generated by Task1.

Output: Timestamped log files in the "Log" subfolder where you run this PowerShell Scripts.

Results: Tables will be created in Azure Synapse Dedicated SQL Pool.

Config File(s) needed (Samples are provided):
sql_synapse.json

Note: Need "Create Schema" and "Create Table Permission" in Azure Synapse SQL Pool.

How many steps?

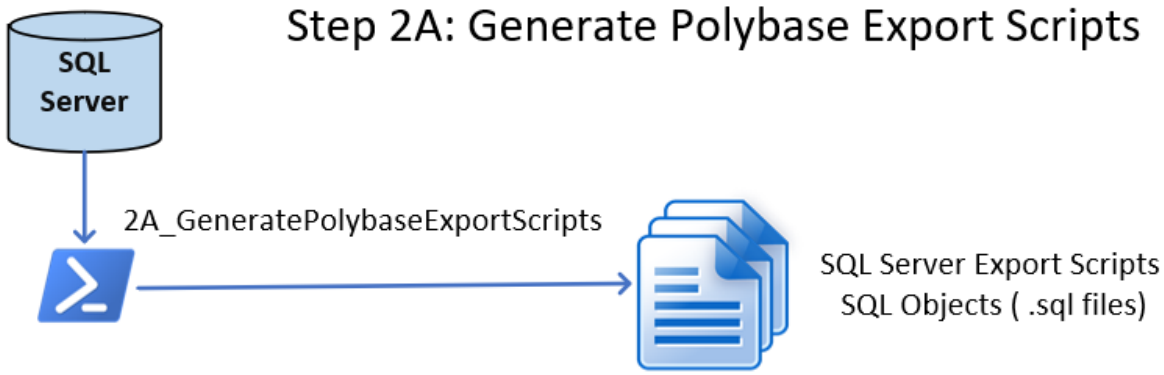
How to Run?

What To Config?

Can I automate?

What Input?

What Output?



Task
Execute PowerShell Scripts "**GenerateExportTablesScripts.ps1**" (Inside 2A_GeneratePolybaseExportScripts).

Output: Polybase Export T-SQL Scripts (Create External Table) for each table in the format of .sql.

Config Files needed (Sample(s) are provided):
ExportTablesConfig.csv
export_tables_config.json

Note 1: Need SQL Server Permissions for create schema, create/drop table, read data.

Note 2: You will need to set up Polybase export functions in SQL Server. Sample T-SQL Scripts are provided in the subfolder Utilities.

How many steps?

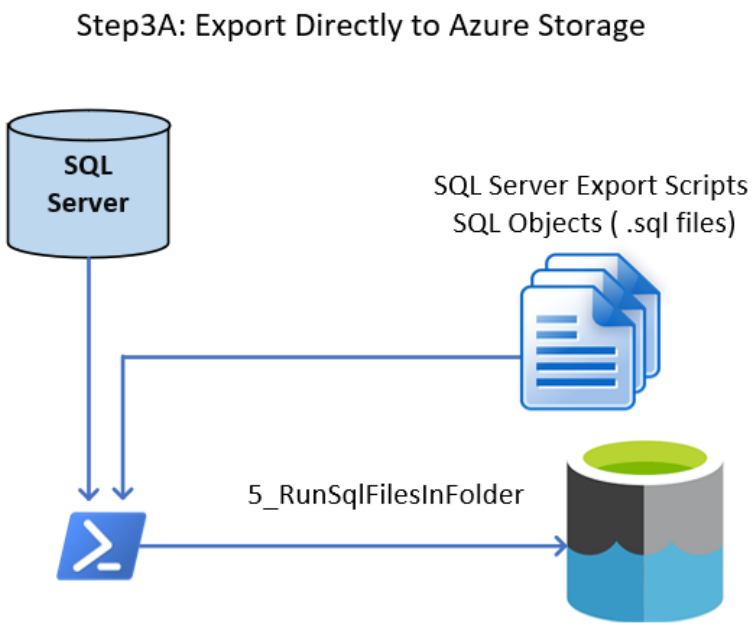
How to Run?

What To Config?

Can I automate?

What Input?

What Output?



Task: Execute PowerShell Scripts"**RunSqlFilesInFolder.ps1**" (Inside folder 5_RunSqlFilsInFolder)

Input: T-SQL script (Polybase Export) generated by **GenerateExportTablesScripts.ps1** (Inside Module 2A_GeneratePolybaseExportScripts), stored in one file folder.

Output: Timestamped log files in the "Log" subfolder where you run this PowerShell Scripts.

Results: Data is exported to Azure Storage from SQL Server Tables

Config File(s) needed: sql_sql.json

Note: (1) Need Blob Storage Contributor Role in Azure Storage
(2) Permissions from SQL Server for
Create Schema, Create Table, Read Data.
(3) Polybase Export is set up in SQL Server. External Data Source and File Format are created. See samples inside subfolder Utilities.

How many steps?

How to Run?

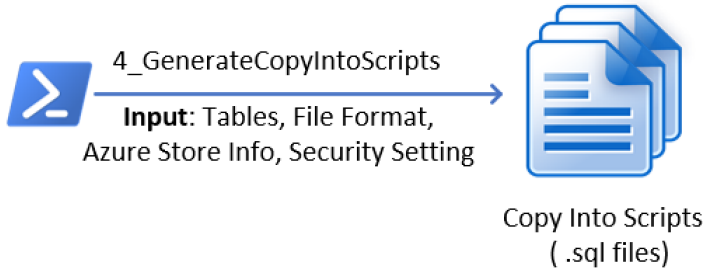
What To Config?

Can I automate?

What Input?

What Output?

Step 4: Generate COPY Into T-SQL Script for Tables (to run in Azure Synapse Dedicated SQL POOL)



Task
Execute PowerShell Scripts "**GenerateCopyIntoScripts.ps1**" (inside folder 4_GenerateCopyIntoScripts)

Output: T-SQL COPY Script for Each Table (in the form of .sql files)

Config Files needed (samples are provided)
(1) mi_csv.json (recommended) or key_csv.json (this requires Storage Account Key, not recommended but can be used for quick testing)
(2) TablesConfig.csv: This is a table list with information needed for each T-SQL Copy script.

Note 1: If using mi_csv.json file, and the Azure Storage was created independently (not as part of the Azure Synapse Workspace Creation), you will need to set up Azure Synapse Workspace as a managed instance, for the Scripts to work. See instruction in top of the "GenerateCopyIntoScripts.ps1". In addition, you can find sample PowerShell script "**SetManagedIdentity.ps1**" (inside subfolder Utilities) to set up Managed Instance.

Note 2: Sample configuration files are provided for data file formats other than CSV: parquet and orc. However, these file formats are not tested.

Step 5: Import Data From Azure Storage into Azure Synapse Dedicated SQL POOL)

How many steps?

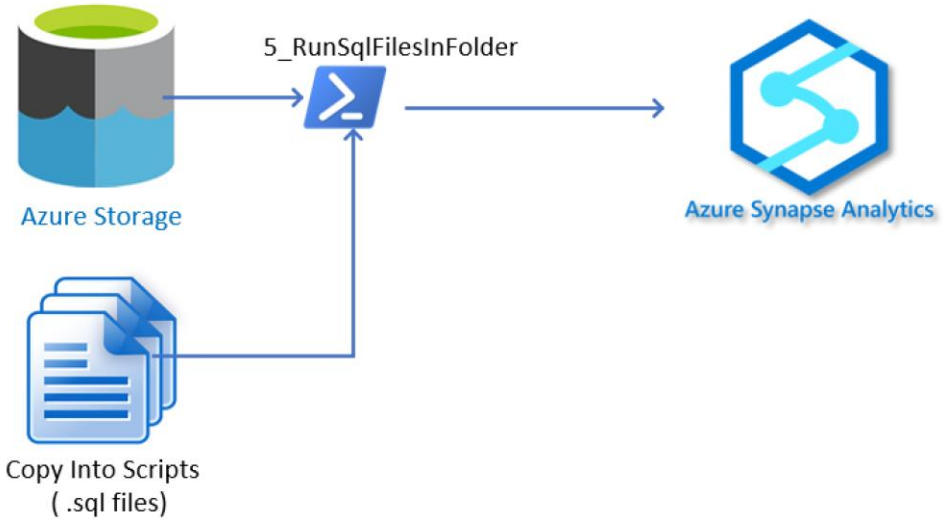
How to Run?

What To Config?

Can I automate?

What Input?

What Output?



Task: Execute PowerShell Scripts"**RunSqlFilesInFolder.ps1**" (Inside folder 5_RunSqlFilsInFolder)

Input: T-SQL script (Copy Into) generated by **GenerateCopyIntoScripts.ps1** (Inside Module 4_GenerateCopyIntoScripts) stored in one file folder.

Output: Timestamped log files in the "Log" subfolder where you run this PowerShell Scripts.

Results: Data is imported to Azure Synapse Tables (Dedicated SQL Pool).

Config File(s) needed: sql_synapse.json or sql_scripts.json

Note: Need "Write Data" Permission" in Azure Synapse SQL Pool.