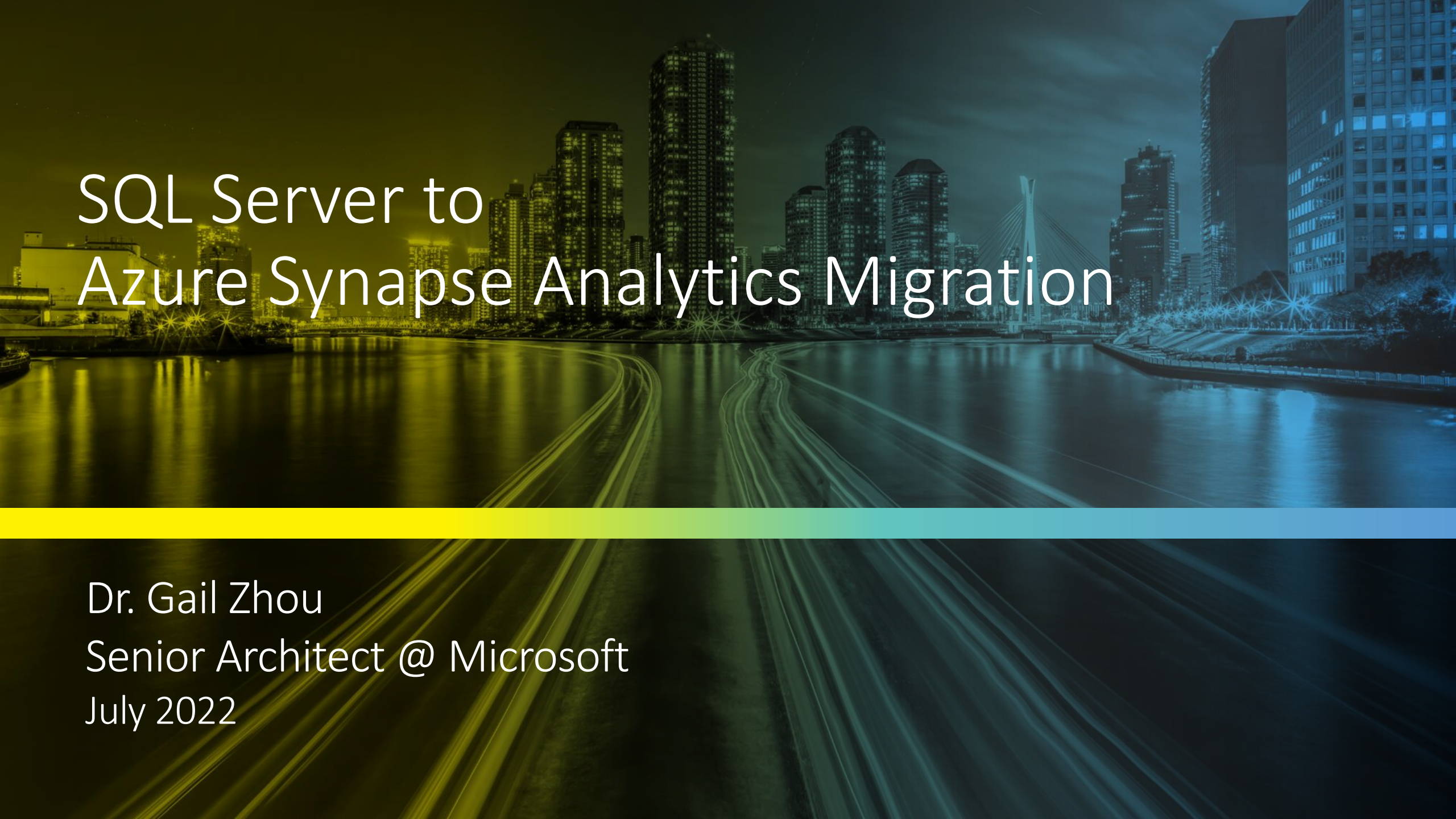


SQL Server to Azure Synapse Analytics Migration

Dr. Gail Zhou
Senior Architect @ Microsoft
July 2022

Agenda (50 min)

Topic	Time
About Presenter – Dr. Gail Zhou	5 min
What is Azure Synapse Analytics? Why Migration? What is the roadmap?	10 min
SQL Server to Azure Synapse Migration Accelerators SQL Server to Synapse Migration Automation Process and Scripts – Assessments GitHub IP Repo & YouTube Training Videos	10 min
Migration Planning Time Accelerators Light Weight Assessment to determine Complexity, Risks, Efforts Migration Task List to help validate SOW & Staffing Plan Estimation Templates to help with database migration effort SSIS and SSRS Assessments IP	10 min
Migration Implementation Time Best Practices and Lessons Learned Key Items to check Identify and Mitigate Key Risks Early Production Migration Best Practices and Tips	5 min
Q & A	10 min

Gaiye “Gail” Zhou

a.k.a. Dr. Gail

SR Architect
Industry Solution Delivery

- **My Home: Atlanta, Georgia, USA.**
- **My Hobbies:** Cooking, Reading, Hiking, Landscaping, and Gardening.
- **My Passion #1:** Deliver innovative business solutions that are simple, scalable, extensible, and transformational.
- **My Passion #2:** I am relentless on innovation, automation, quality, consistency, and repeatable processes.

Prior to Microsoft (22 Years)

- **PhD** in Electrical and Computer Engineering and Executive **MBA** (Employer sponsored).
- 15+ Years of Hands-on and Leadership Experiences in **software engineering** (C++/Java/C#/Python/PowerShell/SQL)
- **10+ Years delivering consulting Projects** with **repeating customers & projects.**
- **10+ years leadership roles:** Director/VP/SVP.
- 3 years as **Chief Architect** for US AF Supply Chain Modernization ($140M Program).
- 2 Years in a **Start-up** building new product line later **acquired by GM.** Started my own company that served 3 corporations.
- **2 Awarded Patents** (7,139,722 & 6,925,586), and Many Innovative Solutions delivered.

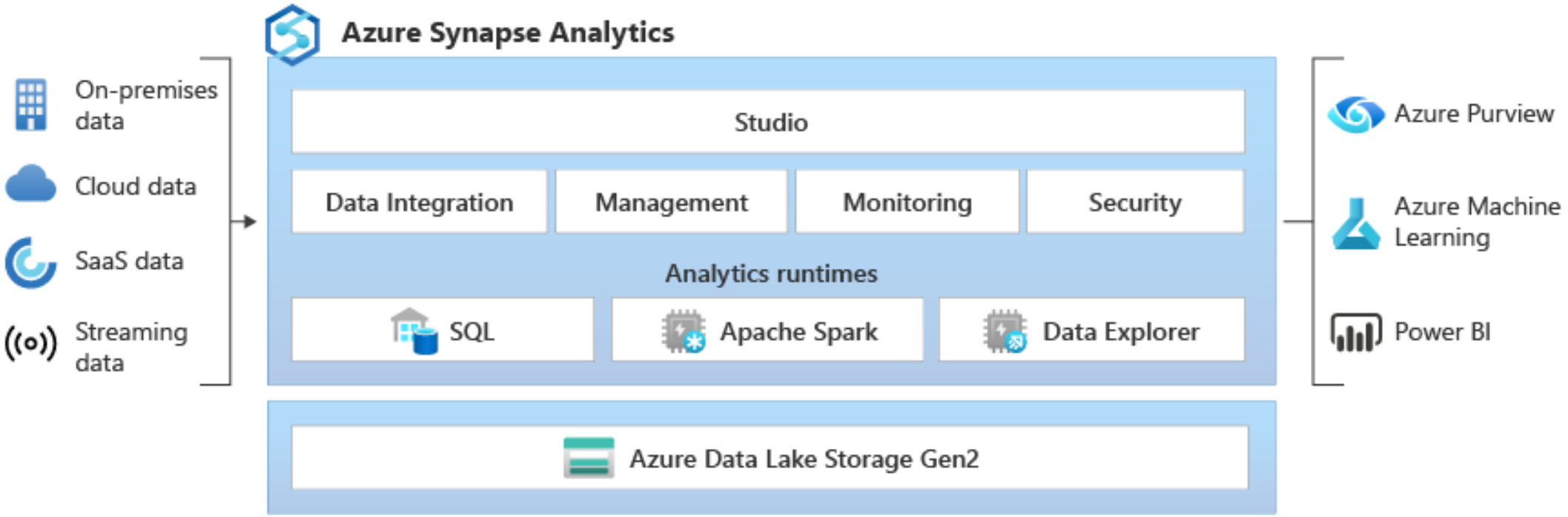
Journey @ Microsoft

Timeline (Joined Jan 2017)
  - Dec 2021–Present: **Early Access Engineering/Innovation Program**
  - Dec 2020–Mar 2022: **Azure Analytics and AI Accelerators (AAAP) Program**
  - Jan 2017–Dec 2020: **Azure Synapse Customer Success Engr. Program**
- Solution Accelerator Lead, “Azure Invoice Process Automation” (Invoice SA 2022).
- 60+ Customer Engagements, 3 GitHub Repositories, YouTube Channel.
- **Pioneered** APS/Netezza/Redshift/Google Big Query/SQL Server to Synapse Migrations.
- **Published** SQL Server to Synapse IP (SQL2Synaspe2021) with YouTube Training.
- Led Coca Cola Amatil APS->Synapse Migration and created reusable IP (APS2Synapse 2018). MCS (now ISD) completed migration using this IP in 3 months, **one month ahead of schedule.** This same IP has been reused to migrate hundreds more APS to Azure Synapse. Upgraded (APS2Synapse2021) with YouTube Training.
- Led **Walgreens**’ Netezza to Synapse Migration, delivered workshops & technical guidance till go-live: Testimonials and Case Study.
- Led **Neogrid’s** Netezza and Redshift to Azure Synapse Migration: Case Study.

SQL Server to Azure Synapse Analytics Migration – What, Why, How?

[What is Azure Synapse Analytics? - Azure Synapse Analytics | Microsoft Docs]

- **Azure Synapse** is an enterprise analytics service that accelerates time to insight across data warehouses and big data systems. Azure Synapse brings together the best of **SQL** technologies used in enterprise data warehousing, **Spark** technologies used for big data, **Data Explorer** for log and time series analytics, **Pipelines** for data integration and ETL/ELT, and deep integration with other Azure services such as **Power BI**, **CosmosDB**, and **AzureML**.



Why Migrating SQL Server DW to Azure Synapse?

Benefits of Migrating into Azure Synapse Analytics:

- Utilizing Azure Infrastructure, PaaS, Security, and many features offered by Azure Synapse Analytics out of box.
- Pay as you go
- Start a new environment in minutes
- Scalability: You can scale your SQL Pool up or down
- Cost saving: Start and Scale up or down QA, Pre-Production, UAT, and Production
- Integration with Cosmos DB, Power BI, and all major BI technologies such as Tableau, Business Object, MicroStrategy
- Speed to market
- Familiar T-SQL Skills
- Spark for Big Data Processing
- Serverless SQL Engine with Auto Start when needed and Stop when not needed
- Built In ELT/ETL Tooling (Azure Data Factory)
- Eliminate the needs of sizing and procure hardware
- Eliminate the needs to OS updated and SQL Server updates
- Access to support personnel using Azure Portal

Migration Journey Overview

Step 1: Assessment, Decision, Architecture, MVP/Pilot

Migration Business Drivers → Product Fit, Azure Synapse Benefits → SQL Server, SSIS, SSRS Assessments → Modern DW Architecture → Key Tasks

1. Code Migration
2. Data Migration
3. Delta Process

Scripts Available (GitHub)

Support Available for other ETL/ETL such as Informatica/DataStage/ODI

Support and Accelerators available from Microsoft, Partners, and Open Source (GitHub)

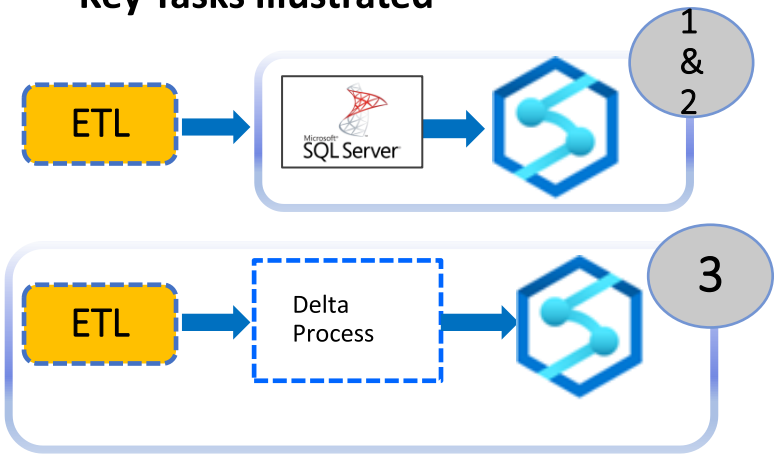
Step 2: Production Migration: Detailed Design and Implementation

Project Planning → SQL Server Workload To Azure Synapse Migration → Operational Azure Synapse

Support and Accelerators available from Microsoft, Partners, and Open Source (GitHub)

Support Available for other ETL/ETL such as Informatica/DataStage/ODI

Key Tasks Illustrated

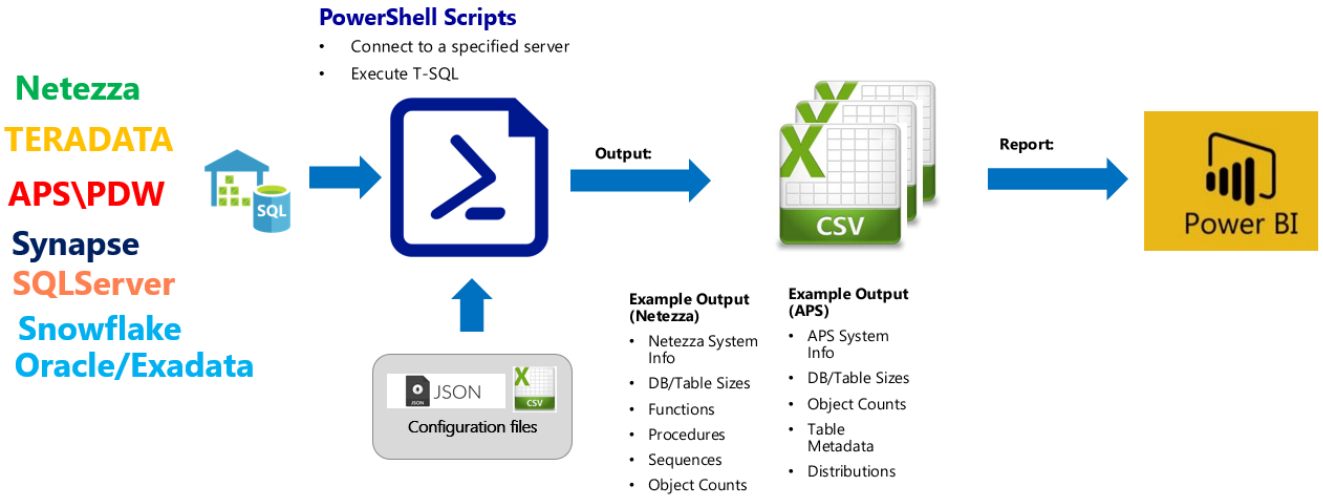



GitHub SQL Server to Azure
Synapse Migration Accelerators

Walk through - GitHub Accelerators

1. SQL Server Assessment Scripts (PowerShell Scripts -> .csv files -> PowerBI Reports):

AzureSynapseScriptsAndAccelerators/Assessment at main · microsoft/AzureSynapseScriptsAndAccelerators · GitHub



Notes:

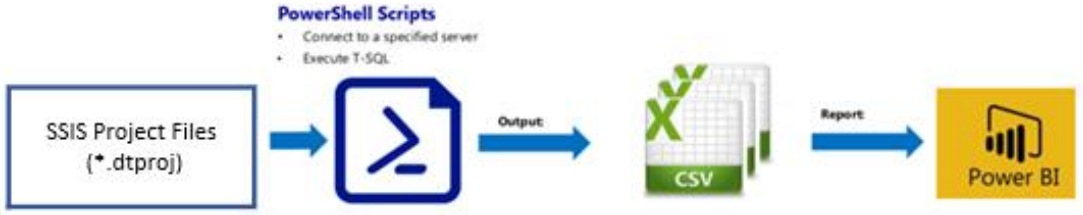
- This assessment tool works with multiple data sources. Customers mostly need some guidance to set up the tool and run it.

- In situations where customer is not comfortable with PowerShell scripts, we can ask customer to run 'Light weight' simple T-SQL Script (see this in later slide)

Walk through - GitHub Accelerators

2. SSIS Assessment Scripts (PowerShell Scripts -> .csv files -> PowerBI Reports):

AzureSynapseScriptsAndAccelerators/Assessment/SSIS at main · microsoft/AzureSynapseScriptsAndAccelerators · GitHub



SSIS (SQL Server Integration Services) is ETL tool which is widely used with data warehouses based on Microsoft SQL Server or APS. When migrating/modernizing these data warehouses in Azure cloud, ETL migration/modernization is frequently the hottest topic.

This toolkit collects SSIS packages inventory at scale and assesses overall complexity. It will help to get insights on the questions such as (but not limited):

- How many SSIS packages are there?
- What is target SQL Server version? What is deployment model (package vs project)?
- How many tasks / data flows / event handlers / connection managers are there?
- What kind of control flow tasks are in use?
- What kind of data flow transformations are in use?
- What kind of connection managers and providers are in use?

Walk through - GitHub Accelerators

4. SQL Server T-SQL Assessment Scripts (Light Weight) AzureSynapseScriptsAndAccelerators/SQL_Server_DB_ObjCounts_and_Sizes.sql at main · microsoft/AzureSynapseScriptsAndAccelerators · GitHub

DatabaseName	Tables	TblsTrkdByCDC	ExtTbls	Columns	IdentityColu	XmlColums	Procs	Views	Triggers	PKeyConstra	FKeyConstra	ScalarFcns	TblValueFcn
Database_name_1	21	0	0	1057	5	0	0	0	0	19	0	0	0
Database_name_2	7	0	0	1002	5	0	0	0	0	4	0	0	0
Database_name_3	12	0	0	1242	11	0	3	11	0	4	0	0	0
Database_Name_X	0	0	0	941	5	0	0	0	0	0	0	0	0
Database_name_5	2	0	0	957	7	0	0	0	0	0	0	0	0
Database_name_6	37	0	0	1852	6	0	6	5	0	5	0	0	0
Database_name_7	49	0	0	1587	45	0	46	37	1	31	27	9	0
Database_name_8	52	0	0	1431	10	0	161	8	6	44	16	15	0
Database_Name_A	55	0	0	3096	45	0	136	19	0	33	0	6	1
Database_Name_B	53	0	0	1591	13	0	164	5	7	42	16	16	0
Database_Name_C	713	0	0	20625	611	0	8	193	4	45	0	6	8
Sandbox	20	0	0	1205	5	0	0	7	0	16	0	2	0
SalesDB	23	0	0	1471	5	0	0	2	0	0	0	0	0
ServiceBus	4	0	0	1097	12	0	6	5	0	3	0	0	0
SSISDB	32	0	0	1579	46	0	114	35	1	32	29	10	2
Database_name_Y	3	0	0	960	6	0	0	0	0	1	0	0	0
Database_name_Z	14	0	0	1592	21	0	9	52	0	12	0	1	0

ServerName	Ndatabases	NTables	NProcedures	NViews	NTriggers	SizeMB	SizeGB	SizeTB	Notes
pr-db-1	31	5754	3164	3755	0	57814392.89	56459.36	60.67	
pr-db-2	39	6092	2412	4643	147	44596418.16	43551.19	46.79	
pr-db-3	11	2145	969	2366	0	2380542.88	2324.75	2.5	
pr-db-4	10	1313	574	821	0	2563077.63	2503.01	2.7	
pr-db-5	11	1408	882	905	0	4854099.13	4740.34	5.09	
Totals	102	16712	8001	12490	147	112208530.7	109578.65	117.75	

Walk through - GitHub Accelerators

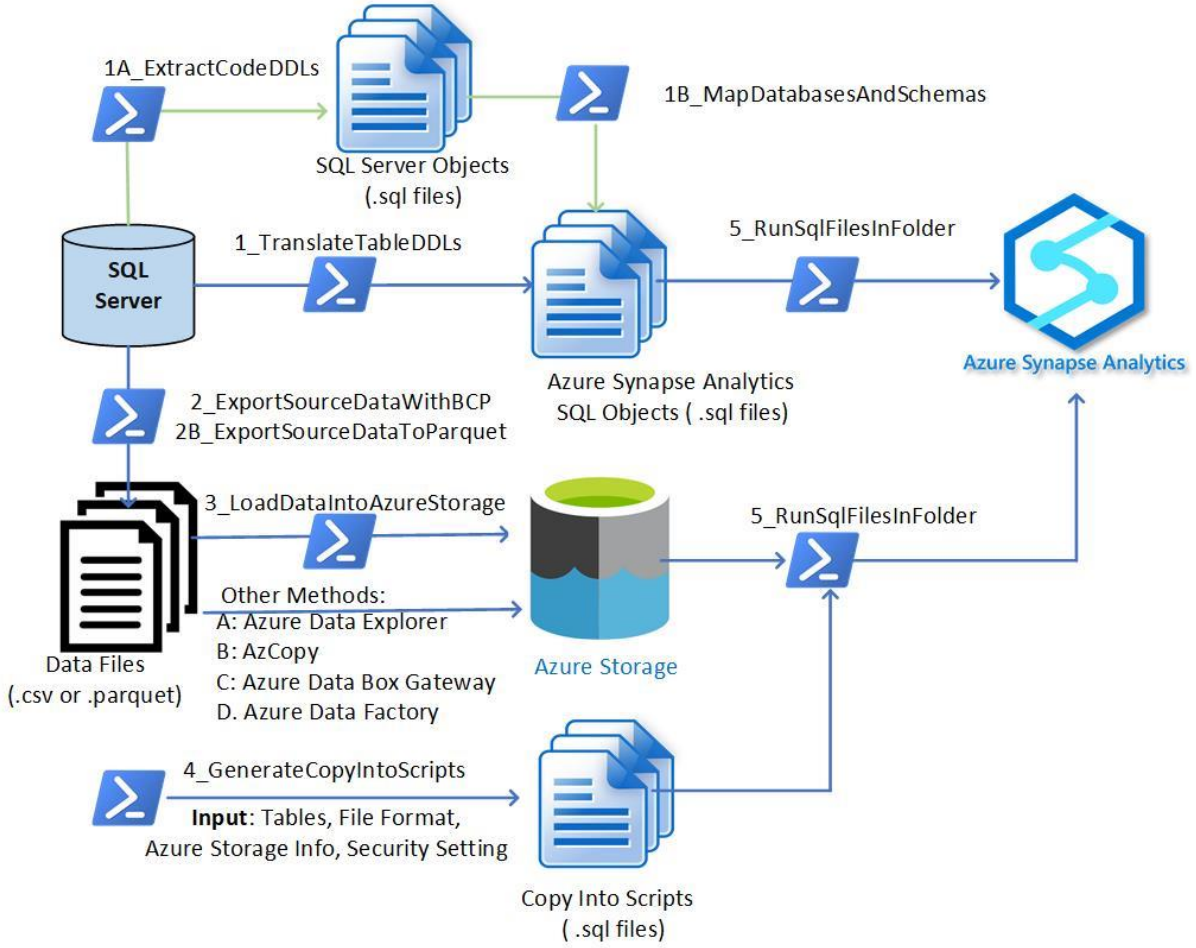
5. SQL Server Code (Stored Procedures or Views) Dependency Analysis – T-SQL Script

AzureSynapseScriptsAndAccelerators/SQL_Server_Check_SP_View_Dependency.sql at main · microsoft/AzureSynapseScriptsAndAccelerators · GitHub

```
/*****************************************************************************/
/*           T-SQL Script to test important and relevant database information          */
/*                          Gaiye "Gail" Zhou, Architect                                 */
/*                               March 2020                                              */
/*****************************************************************************/
--
-- Run this script against each DB in SQL Server
--
--use AdventureWorks2017
SELECT distinct OBJECT_NAME(referencing_id) AS referencing_entity_name,
    o.type_desc AS referencing_desciption,
@@SERVERNAME as referenced_server_name,
ISNULL(referenced_database_name, db_name()) AS referenced_database_name,
s.name as referencing_schema,
    COALESCE(COL_NAME(referencing_id, referencing_minor_id), '(n/a)') AS referencing_minor_id,
    referencing_class_desc,
ISNULL(referenced_schema_name, 'dbo') AS referenced_schema_name,
    referenced_entity_name
    --COALESCE(COL_NAME(referenced_id, referenced_minor_id), '(n/a)') AS referenced_column_name,
    --is_caller_dependent, is_ambiguous
FROM sys.sql_expression_dependencies AS sed  WITH(NOLOCK)
INNER JOIN sys.objects AS o WITH(NOLOCK) ON sed.referencing_id = o.object_id
inner join sys.schemas s on o.schema_id = s.schema_id
where o.type_desc = 'SQL_STORED_PROCEDURE' or o.type_desc = 'VIEW'
```

Walk through - GitHub Accelerators

6. SQL Server to Azure Synapse Migration Automation Process & Scripts [AzureSynapseScriptsAndAccelerators/Migration/SQLServer at main · microsoft/AzureSynapseScriptsAndAccelerators · GitHub]



Training Video Play list

Demos of All Modules


Migration Planning Time
Accelerators

Database Sizes (larger DB, higher Risks), Unsupported data types and relationships

SQL Server T-SQL Assessment Scripts (Light Weigh) [AzureSynapseScriptsAndAccelerators/SQL_Server_DB_ObjCounts_and_Sizes.sql at main · microsoft/AzureSynapseScriptsAndAccelerators · GitHub]

DatabaseName	Tables	TblsTrkdByCDC	ExtTbls	Columns	IdentityColu	XmlColums	Procs	Views	Triggers	PKeyConstra	FKeyConstra	ScalarFcns	TblValueFcn
Database_name_1	21	0	0	1057	5	0	0	0	0	19	0	0	0
Database_name_2	7	0	0	1002	5	0	0	0	0	4	0	0	0
Database_name_3	12	0	0	1242	11	0	3	11	0	4	0	0	0
Database_Name_X	0	0	0	941	5	0	0	0	0	0	0	0	0
Database_name_5	2	0	0	957	7	0	0	0	0	0	0	0	0
Database_name_6	37	0	0	1852	6	0	6	5	0	5	0	0	0
Database_name_7	49	0	0	1587	45	0	46	37	1	31	27	9	0
Database_name_8	52	0	0	1431	10	0	161	8	6	44	16	15	0
Database_Name_A	55	0	0	3096	45	0	136	19	0	33	0	6	1
Database_Name_B	53	0	0	1591	13	0	164	5	7	42	16	16	0
Database_Name_C	713	0	0	20625	611	0	8	193	4	45	0	6	8
Sandbox	20	0	0	1205	5	0	0	7	0	16	0	2	0
SalesDB	23	0	0	1471	5	0	0	2	0	0	0	0	0
ServiceBus	4	0	0	1097	12	0	6	5	0	3	0	0	0
SSISDB	32	0	0	1579	46	0	114	35	1	32	29	10	2
Database_name_Y	3	0	0	960	6	0	0	0	0	1	0	0	0
Database_name_Z	14	0	0	1592	21	0	9	52	0	12	0	1	0

ServerName	Ndatabases	NTables	NProcedures	NViews	NTriggers	SizeMB	SizeGB	SizeTB	Notes
pr-db-1	31	5754	3164	3755	0	57814392.89	56459.36	60.67	
pr-db-2	39	6092	2412	4643	147	44596418.16	43551.19	46.79	
pr-db-3	11	2145	969	2366	0	2380542.88	2324.75	2.5	
pr-db-4	10	1313	574	821	0	2563077.63	2503.01	2.7	
pr-db-5	11	1408	882	905	0	4854099.13	4740.34	5.09	
Totals	102	16712	8001	12490	147	112208530.7	109578.65	117.75	

Check List for High-Risk Items

Use Link: AzureSynapseScriptsAndAccelerators/SQL_Server_DB_ObjCounts_and_Sizes.sql at main · microsoft/AzureSynapseScriptsAndAccelerators · GitHub

- How many SQL Servers? (More SQL Servers, higher Risks)
- How many databases in each server? (More databases, higher risks)
- What are the sizes of the databases (Larger sizes, higher risks, > 30 TB is large, > 100TB is huge.)
- How many tables in each Database (Higher the number, higher risks, 5000+ tables in one DB is higher risk)
- Do you have foreign keys that are enforced? (If yes, high risks, the workload may not be good fit for Synapse)
- Do you have data types that fall into below type? (More of the unsupported data types, higher the risks)
    1. 'geometry',
    2. 'geography',
    3. 'hierarchyid',
    4. 'image',
    5. 'text',
    6. 'ntext',
    7. 'sql_variant',
    8. 'table',
    9. 'timestamp',
    10. 'xml',
    11. 'sequence'

Higher Code Dependency, Higher Risks

SQL Server Code (Stored Procedures or Views) Dependency Analysis – T-SQL Script

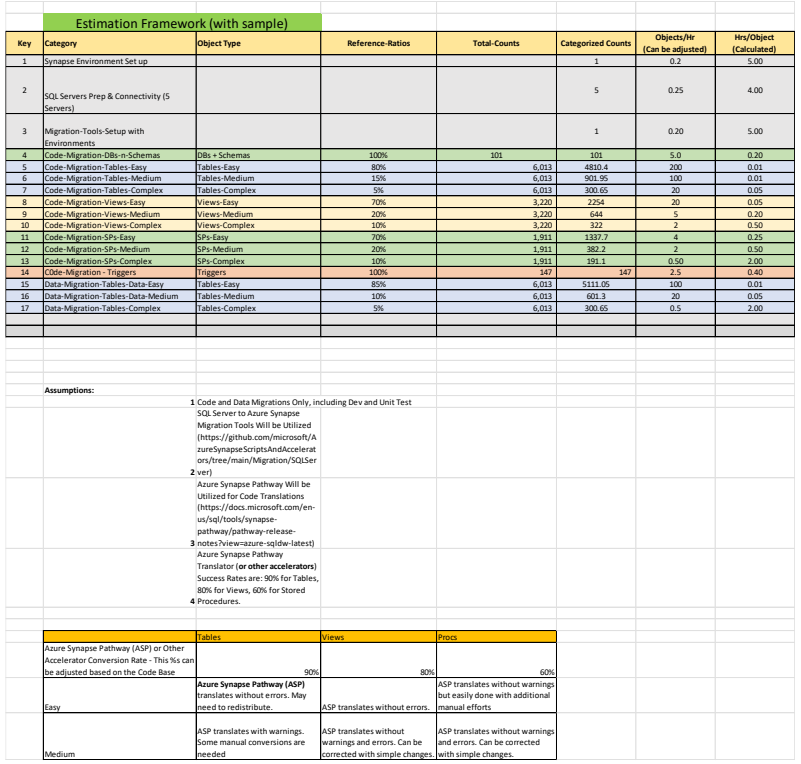
AzureSynapseScriptsAndAccelerators/SQL_Server_Check_SP_View_Dependency.sql at main · microsoft/AzureSynapseScriptsAndAccelerators · GitHub

```
/*****************************************************************************/
/*           T-SQL Script to test important and relevant database information          */
/*                          Gaiye "Gail" Zhou, Architect                                 */
/*                               March 2020                                              */
/*****************************************************************************/
--
-- Run this script against each DB in SQL Server
--
--use AdventureWorks2017
SELECT distinct OBJECT_NAME(referencing_id) AS referencing_entity_name,
    o.type_desc AS referencing_desciption,
@@SERVERNAME as referenced_server_name,
ISNULL(referenced_database_name, db_name()) AS referenced_database_name,
s.name as referencing_schema,
    COALESCE(COL_NAME(referencing_id, referencing_minor_id), '(n/a)') AS referencing_minor_id,
    referencing_class_desc,
ISNULL(referenced_schema_name, 'dbo') AS referenced_schema_name,
    referenced_entity_name
    --COALESCE(COL_NAME(referenced_id, referenced_minor_id), '(n/a)') AS referenced_column_name,
    --is_caller_dependent, is_ambiguous
FROM sys.sql_expression_dependencies AS sed  WITH(NOLOCK)
INNER JOIN sys.objects AS o WITH(NOLOCK) ON sed.referencing_id = o.object_id
inner join sys.schemas s on o.schema_id = s.schema_id
where o.type_desc = 'SQL_STORED_PROCEDURE' or o.type_desc = 'VIEW'
```

Estimation Framework to help with Migration Hours (SQL Server)

[AzureSynapseScriptsAndAccelerators/Migration/SQLServer/6_Bonus/EstimationFramework at main · microsoft/AzureSynapseScriptsAndAccelerators · GitHub]

Use the GitHub Link to download the .xlsx file framework to your desktop to use. The input numbers can be plugged in from the results of the 'light weight' SQL Server Assessment T-SQL Script Results. You can use this estimation framework for production migration projects.



Estimation Framework (with sample)

Key	Category	Object Type	Reference-Ratios	Total-Counts	Categorized Counts	Objects/Hr (Can be adjusted)	Hrs/Object (Calculated)
1	Synapse Environment Set up				1	0.2	5.00
2	SQL Servers Prep & Connectivity (5 Servers)				5	0.25	4.00
3	Migration-Tools-Setup with Environments				1	0.20	5.00
4	Code-Migration-DBs-n-Schemas	DBs + Schemas	100%	101	101	5.0	0.20
5	Code-Migration-Tables-Easy	Tables-Easy	80%	6,013	4810.4	200	0.01
6	Code-Migration-Tables-Medium	Tables-Medium	15%	6,013	901.95	100	0.01
7	Code-Migration-Tables-Complex	Tables-Complex	5%	6,013	300.65	20	0.05
8	Code-Migration-Views-Easy	Views-Easy	70%	3,220	2254	20	0.05
9	Code-Migration-Views-Medium	Views-Medium	20%	3,220	644	5	0.20
10	Code-Migration-Views-Complex	Views-Complex	10%	3,220	322	2	0.50
11	Code-Migration-SPs-Easy	SPs-Easy	70%	1,911	1337.7	4	0.25
12	Code-Migration-SPs-Medium	SPs-Medium	20%	1,911	382.2	2	0.50
13	Code-Migration-SPs-Complex	SPs-Complex	10%	1,911	191.1	0.50	2.00
14	COde-Migration - Triggers	Triggers	100%	147	147	2.5	0.40
15	Data-Migration-Tables-Data-Easy	Tables-Easy	85%	6,013	5111.05	100	0.01
16	Data-Migration-Tables-Data-Medium	Tables-Medium	10%	6,013	601.3	20	0.05
17	Data-Migration-Tables-Complex	Tables-Complex	5%	6,013	300.65	0.5	2.00

Assumptions:

1. Code and Data Migrations Only, including Dev and Unit Test
2. SQL Server to Azure Synapse Migration Tools Will be Utilized (https://github.com/microsoft/AzureSynapseScriptsAndAccelerators/tree/main/Migration/SQLServer)
3. Azure Synapse Pathway Will be Utilized for Code Translations (https://docs.microsoft.com/en-us/sql/tools/synapse-pathway/pathway-release-notes?view=azure-sqldw-latest)
4. Azure Synapse Pathway Translator (**or other accelerators**) Success Rates are: 90% for Tables, 80% for Views, 60% for Stored Procedures.

	Tables	Views	Procs
Azure Synapse Pathway (ASP) or Other Accelerator Conversion Rate - This %s can be adjusted based on the Code Base	90%	80%	60%
Easy	**Azure Synapse Pathway (ASP)** translates without errors. May need to redistribute.	ASP translates without errors.	ASP translates without warnings but easily done with additional manual efforts
Medium	ASP translates with warnings. Some manual conversions are needed	ASP translates without warnings and errors. Can be corrected with simple changes.	ASP translates without warnings and errors. Can be corrected with simple changes.

MVP/POC Migration Task List & Staffing Plan*

AzureSynapseScriptsAndAccelerators/POC-or-MVP-Scoping-Project-Plan-Tasks-Starter.xlsx at main · microsoft/AzureSynapseScriptsAndAccelerators · GitHub

*Please note this sample project plan is for MVP or POC. It should not be used as an authoritative source for SOW scope coverages. It can be used as starter plan during initial SOW writing.

Use the GitHub Link to download the .xlsx file framework to your desktop to use. Compare this with the SOW scope items. Bridge gaps where exist.

SQL Server DW to Azure Synapse Migration

#	Item	Description	Notes
1	Portal Access	POC (MVP) team members should all be Contributors or Owners within the Resource Group created for the POC (MVP)	Customer user to set up from Azure portal: portal.azure.com
2	Azure Synapse Admin	A customer AAD member will need to be the Azure Synapse Admin, the same person can be the Azure Active Directory Admin for Azure Synapse (see item below).	Customer user. Access to the Administrator SQL Login for Azure Synapse
3A	Azure Active Directory (AAD) Admin	A customer AAD member will need to be the Azure Active Directory Admin for Azure Synapse. This user can create AAD users in Azure Synapse (allowing single sign-on using AAD).	Customer user
3B	Azure Active Directory (AAD) User(s)	POC (MVP) Team members should be made Guest users within AAD. If this is not possible then all Portal activities will become the responsibility of the customer POC (MVP) team members.	POC (MVP) team
4A	Azure Synapse Login Access	POC (MVP) team members should each have their own SQL Login or AAD login	POC (MVP) team
4B	Azure Synapse Privileges	POC (MVP) team member should be members of the db_owner database role (i.e. dbo) of the Azure Synapse	POC (MVP) team
4C	Azure Synapse Load User(s)	Load User SQL login with resource class as largerc.	
4C	Azure Synapse Performance Tester(s)	Load User SQL login with resource class as staticrc50 - staticrc80	
5	Azure Data Lake Store (ADLS)	POC (MVP) team members will need full access to ADLS storage for the POC (MVP)	POC (MVP) team
6	Azure Data Factory (ADF)	Synapse Load User(s) need full access to ADF	POC (MVP) data loader
7	Network Connectivity	POC (MVP) Users will need to connect to Azure Synapse database from Azure Data Factory (within Azure), SQL Server Management Studio (from laptop remote)	POC (MVP) team

Migration Implementation Time Best Practices and Tips

Migration Project Best Practices

- Reengineering code during migration is not recommended (how do you test new code? What if you broke the system that are mission critical? How much longer will the migration be if you add additional work into it?)
- Use Agile Process
- Set up network and access in first sprint
- First Sprint to Migrate small number of Objects to create and smooth out migration process
- Assess and mange Code Dependencies to determine the right migration sequence
- Assess data types that are not compatible with target environment and determine mapping methods
- Work with network management group to secure bandwidth for large volumes of data transfer
- Perform some data cleaning before migration will be beneficial, such as rid of unused tables and code to reduce migration efforts (but be careful to make sure they are no longer needed)