



SQLBITS

10 tips for a successful migration from Databricks and ADF to Synapse

Insert coins to start



10 tips for a successful migration from Databricks and ADF to Synapse

1. Solution architecture differences
2. Spark version compatibility
3. Structured Streaming differences
4. Solution deployment differences
5. Platform/solution monitoring & tuning
6. The dbutils counterpart
7. Notebook management shortcuts
8. Orchestration pitfalls
9. Cost management
10. Best-practices



Are you planning a migration of your data platform to Azure Synapse Analytics? Is it currently based on Azure Databricks, Azure Data Factory, and Azure Data Lake Storage? Make sure to grab these 10 tips! We'll talk about Spark compatibilities, orchestration pitfalls and solution deployment differences.



BLUE ROCKET
DATA CONSULTING & SOLUTIONS

Dave Ruijter

Data Platform MVP | Azure Solution Architect Data & Analytics Consultant

Data & analytics all-rounder with a strong technical focus. Enjoys facilitating workshops and training sessions to share knowledge and build other people's skills.



dave@blue-rocket.it



@DaveRuijter



linkedin.com/in/DaveRuijter



ModernData.ai



Microsoft®
Most Valuable
Professional

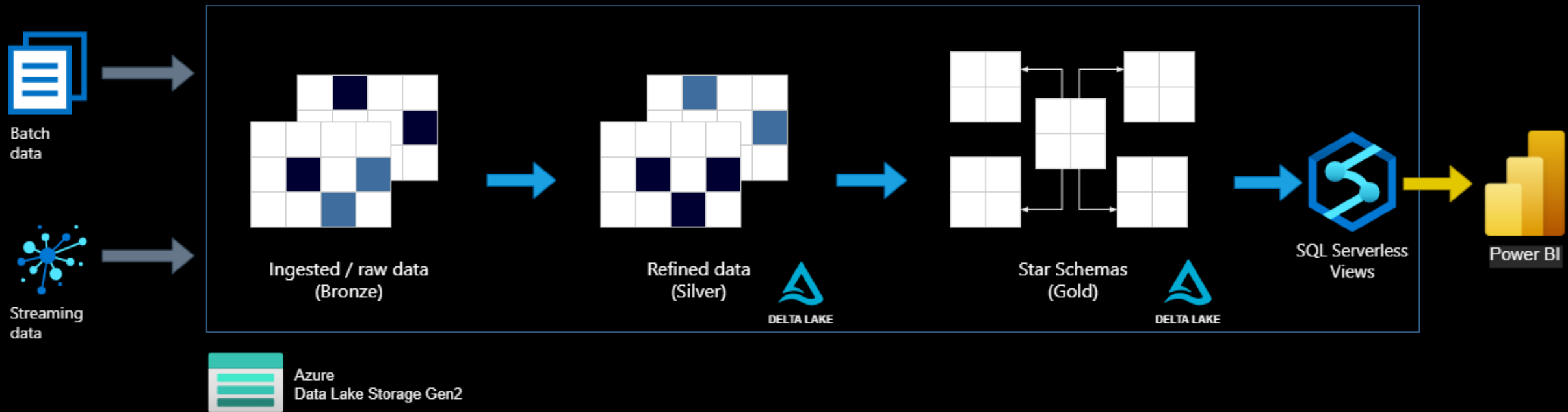


Feedback
Feedback
Feedback



<https://sqlb.it/?6953>

🧪 1. Solution architecture differences



2. Spark version compatibility

 Cluster config -> demo New Apache Spark pool

Size	vCore	Memory
Small	4	32 GB
Medium	8	64 GB
Large	16	128 GB
XLarge	32	256 GB
XXLarge	64	512 GB
XXX Large (Isolated Compute)	80	504 GB



2. Spark version comp



Cluster config



Runtime versions

Runtime name	Release date	Release stage
Azure Synapse Runtime for Apache Spark 2.4	December 15, 2020	GA
Azure Synapse Runtime for Apache Spark 3.1	May 26, 2021	GA

Version	Variant	Apache Spark version	Release date	End-of-support date
10.3		3.2.1	Feb 02, 2022	Aug 02, 2022
	Databricks Runtime 10.3 (includes Photon)			
	Databricks Runtime 10.3 for Machine Learning			
10.2		3.2.0	Dec 22, 2021	Jun 22, 2022
	Databricks Runtime 10.2 (includes Photon)			
	Databricks Runtime 10.2 for Machine Learning			
10.1		3.2.0	Nov 10, 2021	May 10, 2022
	Databricks Runtime 10.1 (includes Photon)			
	Databricks Runtime 10.1 for Machine Learning			
10.0		3.2.0	Oct 20, 2021	Apr 20, 2022
	Databricks Runtime 10.0 (includes Photon)			
	Databricks Runtime 10.0 for Machine Learning			
9.1 LTS		3.1.2	Sep 23, 2021	Sep 23, 2023
	Databricks Runtime 9.1 LTS (includes Photon)			
	Databricks Runtime 9.1 LTS for Machine Learning			
7.3 LTS		3.0.1	Sep 24, 2020	Sep 24, 2022
	Databricks Runtime 7.3 LTS			
	Databricks Runtime 7.3 LTS for Machine Learning			






2. Spark compatibility

-  Cluster config
-  Runtime versions
-  Cluster sharing ->
 -  Each notebook needs a separate Spark session!
 -  Consider having developer-specific clusters
 -  Stop notebook sessions immediately when you are finished




2. Spark version compatibility

-  Cluster config
-  Runtime versions
-  Cluster sharing
-  Languages support -> Synapse has built-in support for .NET!

2. Spark version compatibility

-  Cluster config
-  Runtime versions
-  Cluster sharing
-  Languages support
-  Data Lake available without mounting

3. Structured Streaming differences

-  No jobs
-  Unlike Databricks, the Synapse streaming data frames cannot be examined using a simple Display command.
-  With Databricks, the progress of the running streams and their essential stats can be tracked using nice charts. Unfortunately, that is not the case for Synapse streaming at this point.



4. Solution deployment differences

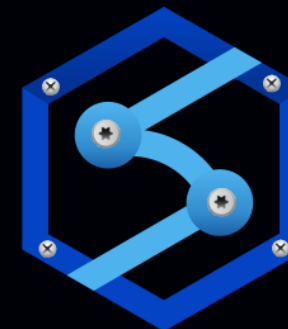
- SQL Notebook code linting
- SQL Automated testing
- SQL Notebook deployment
- SQL ADF vs Synapse Pipelines differences

SQLPlayer/azure.synapse.tools:

PowerShell module to deploy Synapse workspace (and more) in Microsoft Azure.
(github.com)

README.md

azure.synapse.tools



What is supported

The deployment of these objects:

- Workspace instance
- dataset
- dataflow
- integration runtime
- linked service
- pipeline
- KQL script *
- SQL script *
- notebook *





| * via RestAPI only

What is NOT yet supported

The deployment of these objects:

- credential
- Spark job definition
- 'AzResource' deployment method

5. Platform/solution monitoring & tuning

-  Azure Log Analytics integration
-  No ganglia!
-  Spark application monitoring:
 -  Azure Log Analytics integration!

6. The dbutils counterpart

mssparkutils!

Microsoft Spark Utilities (MSSparkUtils) is a builtin package to help you easily perform common tasks. You can use MSSparkUtils to work with file systems, to get environment variables, to chain notebooks together, and to work with secrets. MSSparkUtils are available in PySpark (Python), Scala, and .NET Spark (C#) notebooks and Synapse pipelines.

```
mssparkutils.fs provides utilities for working with various FileSystems.
```

```
Below is overview about the available methods:
```

```
cp(from: String, to: String, recurse: Boolean = false): Boolean -> Copies a file or directory
mv(from: String, to: String, recurse: Boolean = false): Boolean -> Moves a file or directory
ls(dir: String): Array -> Lists the contents of a directory
mkdirs(dir: String): Boolean -> Creates the given directory if it does not exist,
put(file: String, contents: String, overwrite: Boolean = false): Boolean -> Writes the contents of a file
head(file: String, maxBytes: int = 1024 * 100): String -> Returns up to the first maxBytes of the file
append(file: String, content: String, createFileIfNotExists: Boolean): Boolean -> Appends the content to the file
rm(dir: String, recurse: Boolean = false): Boolean -> Removes a file or directory
```

```
Use mssparkutils.fs.help("methodName") for more info about a method.
```

7. Notebook management shortcuts






Shortcut keys under command mode

Run the current cell and select below	Shift+Enter
Run the current cell and insert below	Alt+Enter
Run current cell	Ctrl+Enter
Select cell above	Up
Select cell below	Down
Select previous cell	K
Select next cell	J
Insert cell above	A
Insert cell below	B
Delete selected cells	Shift+D
Switch to edit mode	Enter






Shortcut keys under edit mode

Move cursor up	Up
Move cursor down	Down
Undo	Ctrl + Z
Redo	Ctrl + Y
Comment/Uncomment	Ctrl + /
Delete word before	Ctrl + Backspace
Delete word after	Ctrl + Delete
Go to cell start	Ctrl + Home
Go to cell end	Ctrl + End
Go one word left	Ctrl + Left
Go one word right	Ctrl + Right
Select all	Ctrl + A
Indent	Ctrl +]
Dedent	Ctrl + [
Switch to command mode	Esc



#8. Orchestration pitfalls

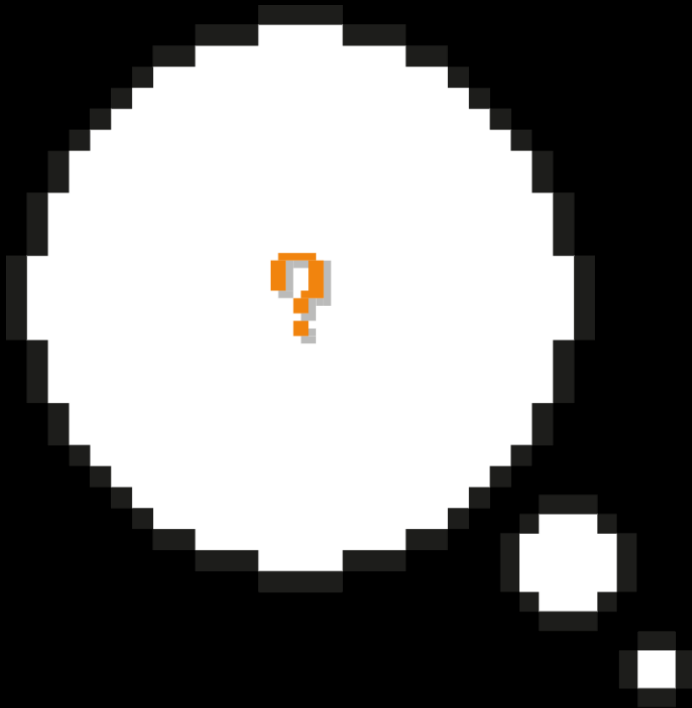
-  Calling notebooks from a pipeline: use parameters instead of widgets
-  For Spark notebooks: be aware of session/cluster sharing limitations
-  For Spark jobs: no support to call a notebook
-  No support yet for Power Query Activity in Pipelines
-  No support yet for IR sharing across different data factories

#9. Cost management

-  Cost Control for SQL pools
-  Azure Cost Calculator
-  Azure Cost analysis
-  Budgets
-  Pre-purchase

#10. Best-practices

-  Have multiple Spark pools available, also without auto-scale, as it can take 1 to 5 minutes for a scaling operation to complete
-  Really good Docs pages!



Any bonus questions?