DATA( ) {
EXPOSED;

DATA EXPOSED SPECIAL

Around the Clock with Azure SQL and Azure Data Factory

Americas
February 3, 2021
09:00 - 17:00 PT

Asia
February 4, 2021
09:00 - 17:00 SGT

HOSTED BY
Wee Hyong Tok & Anna Hoffman

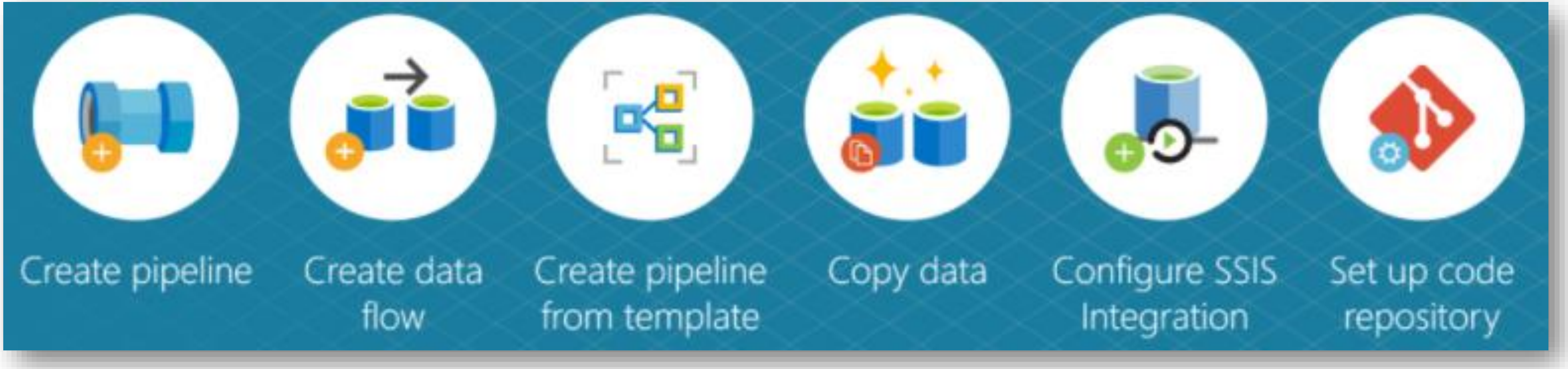
16 Sessions | 2 Ask the Expert Panels | 1 Hackathon

DATA( ) {
EXPOSED;

Designing a Data Architecture with Azure Data Factory

Buck Woody - Microsoft

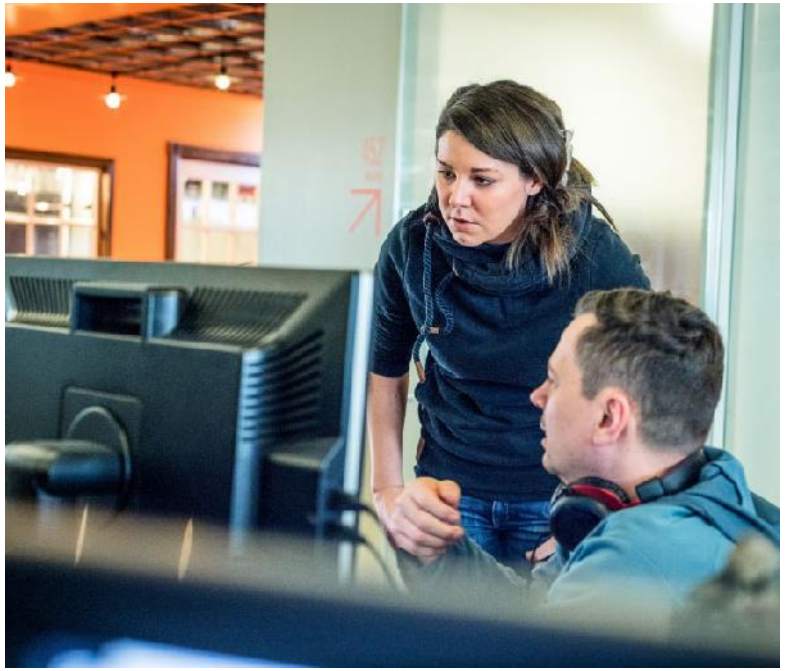
Centering on the Pipeline



Designing your Architecture

Solution Thinking

Good design starts with an understanding of the opportunity, the requirements and constraints, and your options



Sources and Sinks

Designing the architecture requires a large vocabulary of your options

What to Use When

With an understanding of the opportunity and a plan for the design, you can select your choices and finalize the design with options

Solution Thinking

Designing a Data Architecture involves considerations for Storage, Compute , and Data Paths and Movement (Pipelines)

General Architecture

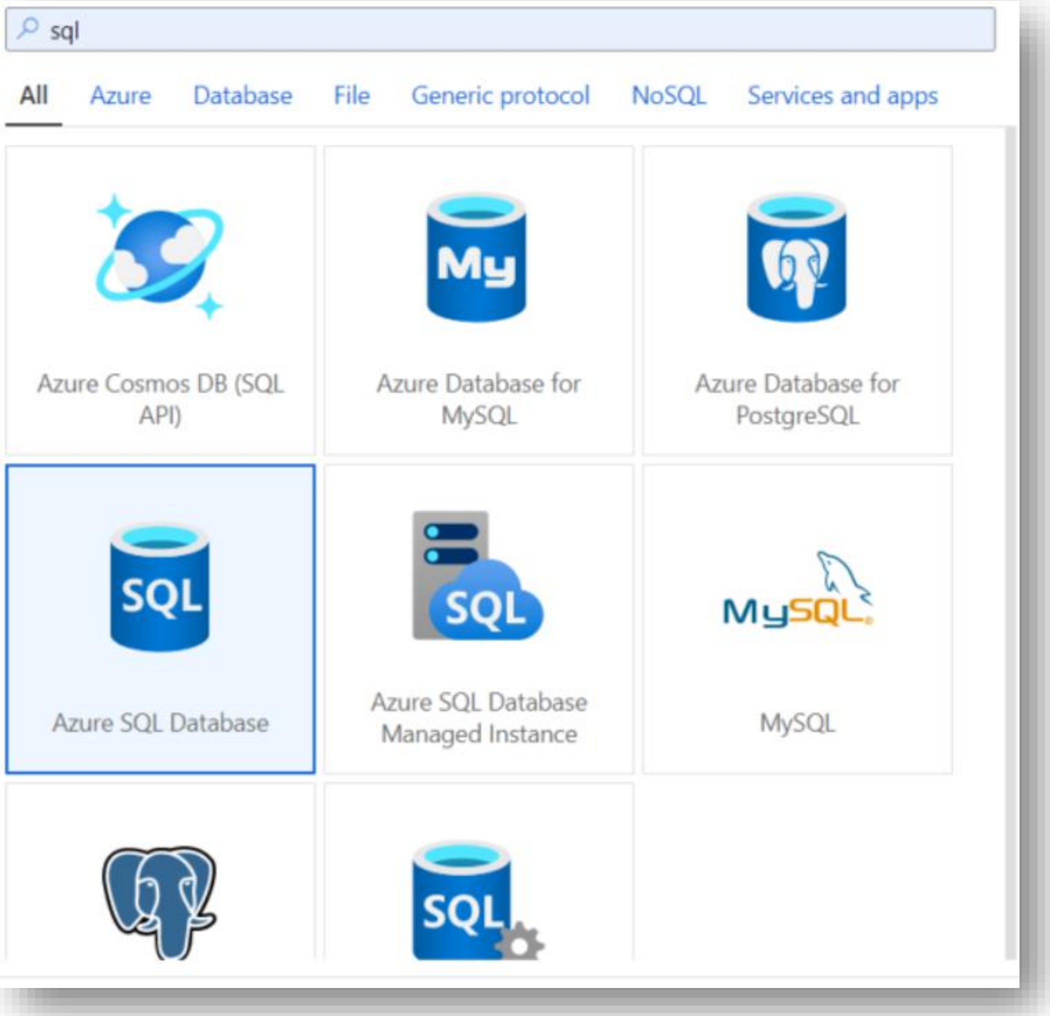
- Architecture Center
- Well-Architected Center
- DevOps

Data Patterns and Specifics

- aka.ms/dag

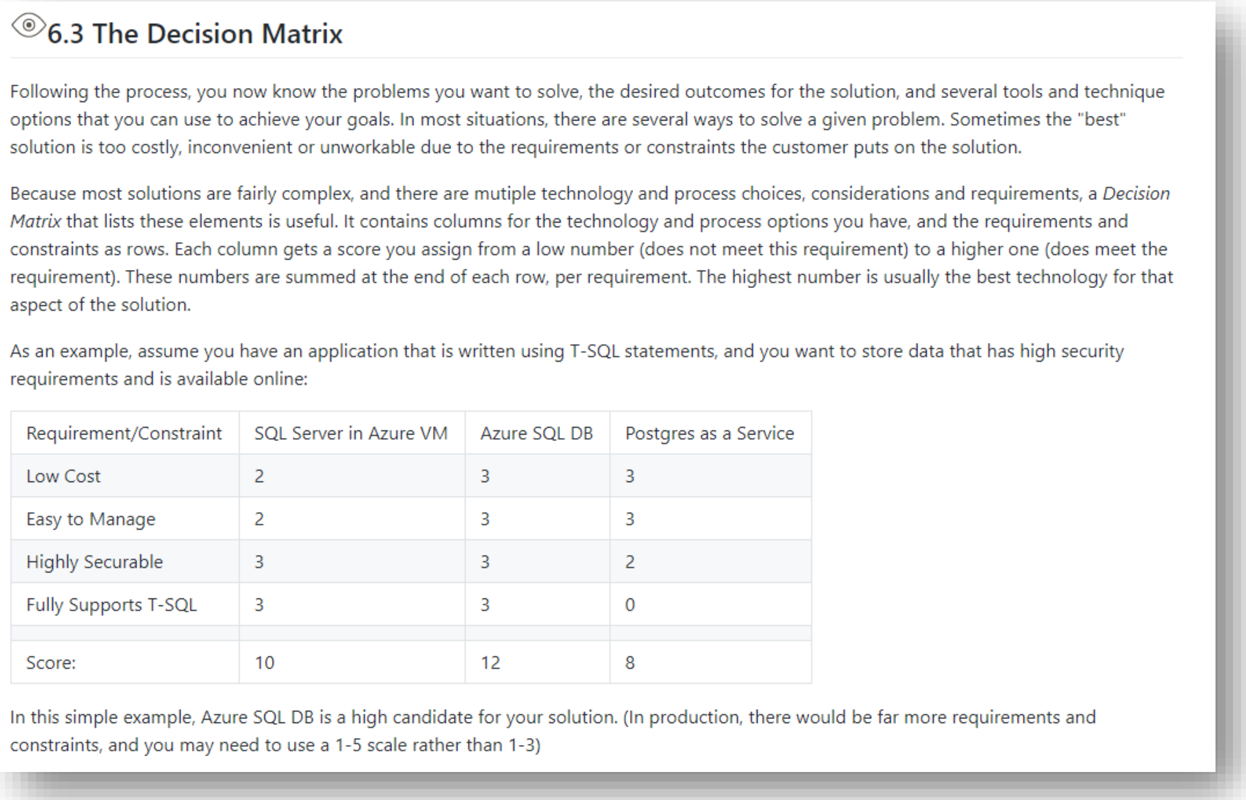
Microsoft Azure

Knowing your options – Sources and Sinks



Microsoft Azure

What to Use When – pairing up the options with the requirements



6.3 The Decision Matrix

Following the process, you now know the problems you want to solve, the desired outcomes for the solution, and several tools and technique options that you can use to achieve your goals. In most situations, there are several ways to solve a given problem. Sometimes the "best" solution is too costly, inconvenient or unworkable due to the requirements or constraints the customer puts on the solution.

Because most solutions are fairly complex, and there are mutiple technology and process choices, considerations and requirements, a *Decision Matrix* that lists these elements is useful. It contains columns for the technology and process options you have, and the requirements and constraints as rows. Each column gets a score you assign from a low number (does not meet this requirement) to a higher one (does meet the requirement). These numbers are summed at the end of each row, per requirement. The highest number is usually the best technology for that aspect of the solution.

As an example, assume you have an application that is written using T-SQL statements, and you want to store data that has high security requirements and is available online:

Requirement/Constraint	SQL Server in Azure VM	Azure SQL DB	Postgres as a Service
Low Cost	2	3	3
Easy to Manage	2	3	3
Highly Securable	3	3	2
Fully Supports T-SQL	3	3	0
Score:	10	12	8

In this simple example, Azure SQL DB is a high candidate for your solution. (In production, there would be far more requirements and constraints, and you may need to use a 1-5 scale rather than 1-3)

DATA( ) {
EXPOSED;

Learn with us!

View our on-demand playlist:
aka.ms/azuresqlandadf

@AzureSQL
@AzDataFactory

Microsoft Azure