

Azure Data Lake. The Services. The U-SQL. The C-Sharp

With Paul Andrew





Paul Andrew

About Me



Microsoft Data Platform [MVP](#)



Business Intelligence Consultant at [Purple Frog Systems](#)



STEM Ambassador with [STEM Learning UK](#)



PASS Chapter Leader for [Microsoft Data Platform Group](#)



Speaker & part of the [SQL Relay](#) team



Speaker & helper at [SQL Bits](#)



SQLSATURDAY

Speaker & helper at various [SQL Saturday](#) 's



[Stack Overflow](#)

Unsung Hero

azure-data-factory

Many years' experience working with the complete on premises SQL Server stack in a variety of roles and industries. Now developing/consulting on hybrid business intelligence solutions using Microsoft Azure. Specialising in Data Factory, Data Lake Analytics, real-time data with the Internet of Things (IoT), Stream Analytics and Event Hubs.



*Best New Speaker
SQL Saturday
Dublin 2017*



[linkedin.com/in/mrpaulandrew](https://www.linkedin.com/in/mrpaulandrew)



[@mrpaulandrew](https://twitter.com/mrpaulandrew)



paul@purplefrogsystems.com



purplefrogsystems.com/paul

Gold Sponsors



ROBERT WALTERS



PYRAMID
ANALYTICS

@SQLSatMcr

Gold Sponsors



Silver Sponsors



@SQLSatMcr

Bronze Sponsors

Pragmatic
Works

 arapaima.uk

 redgate

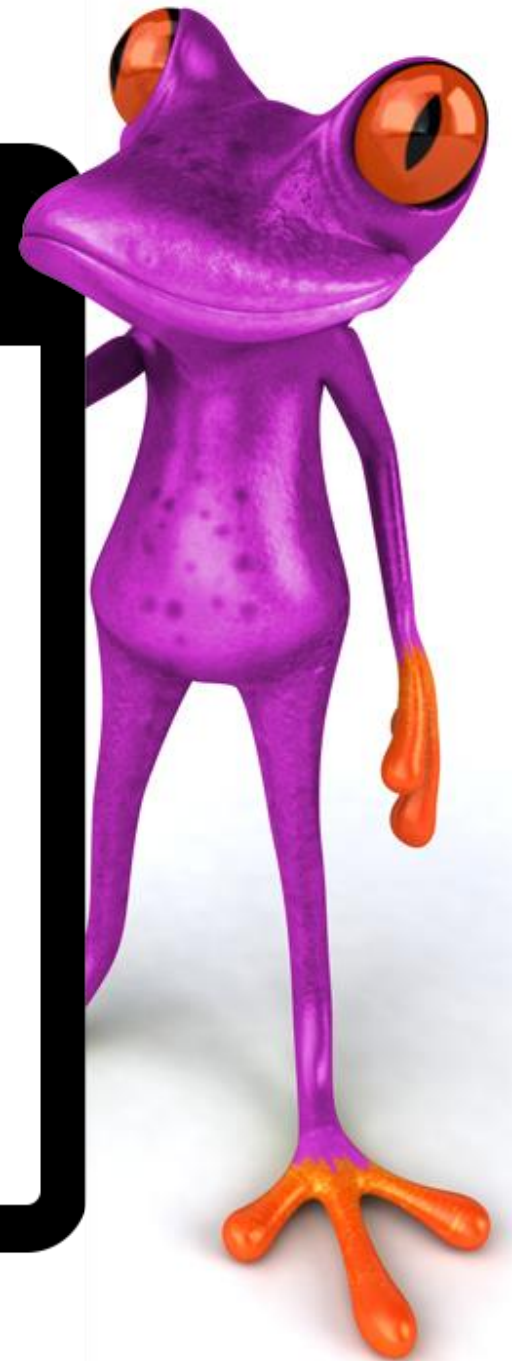


The Services

- What is Azure Data Lake?
Storage & compute
- Why use it?
Real world examples
- Working with each service

The SQL The Sharp

- U-SQL
- The language (T-SQL, C#)
- Database objects
- Advanced capabilities





What is Azure Data Lake?

Storage & compute

Why use it?

Real world examples

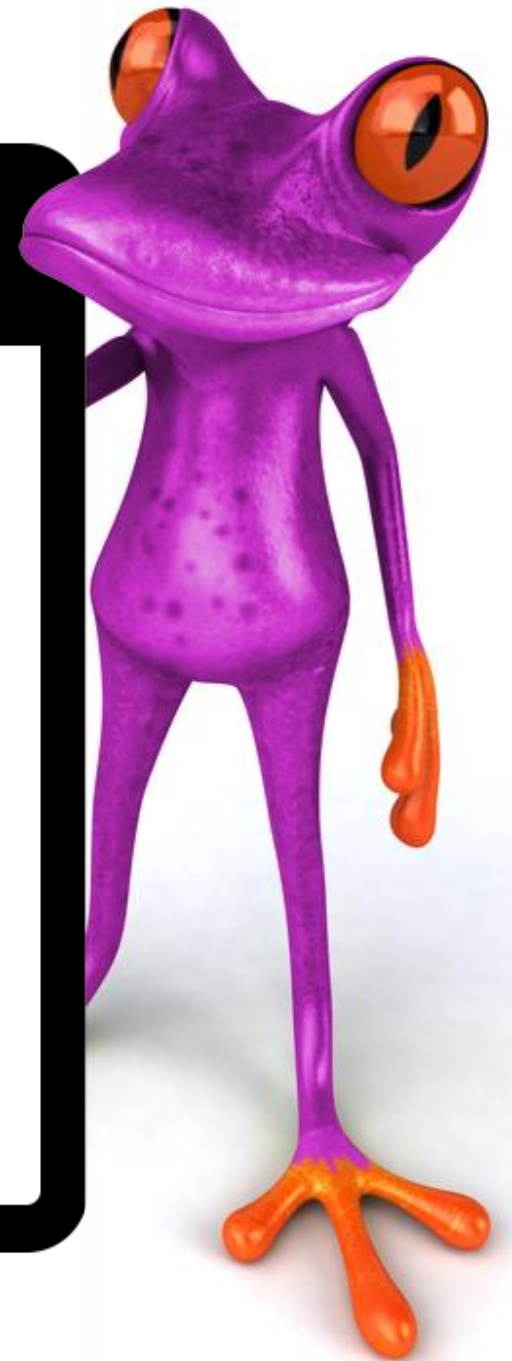
Working with each service

U-SQL

The language (T-SQL, C#)

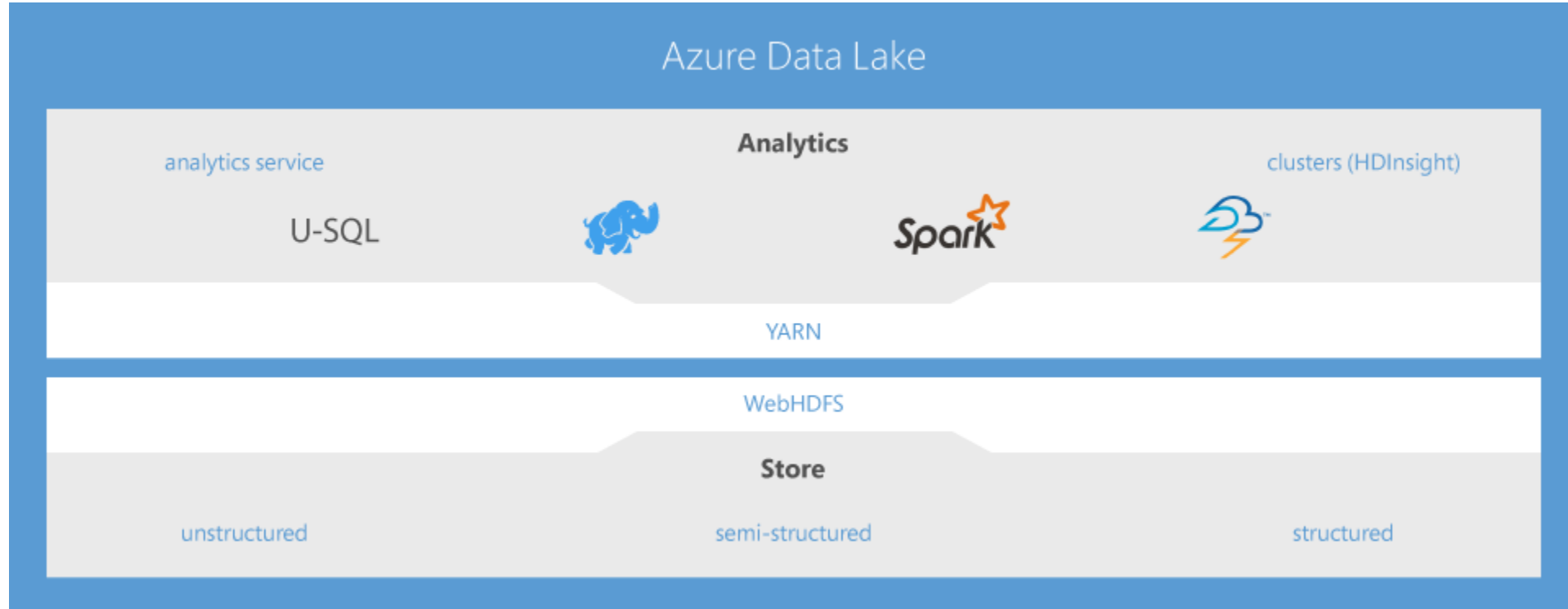
Database objects

Advanced capabilities



What is Azure Data Lake?

The Microsoft version...

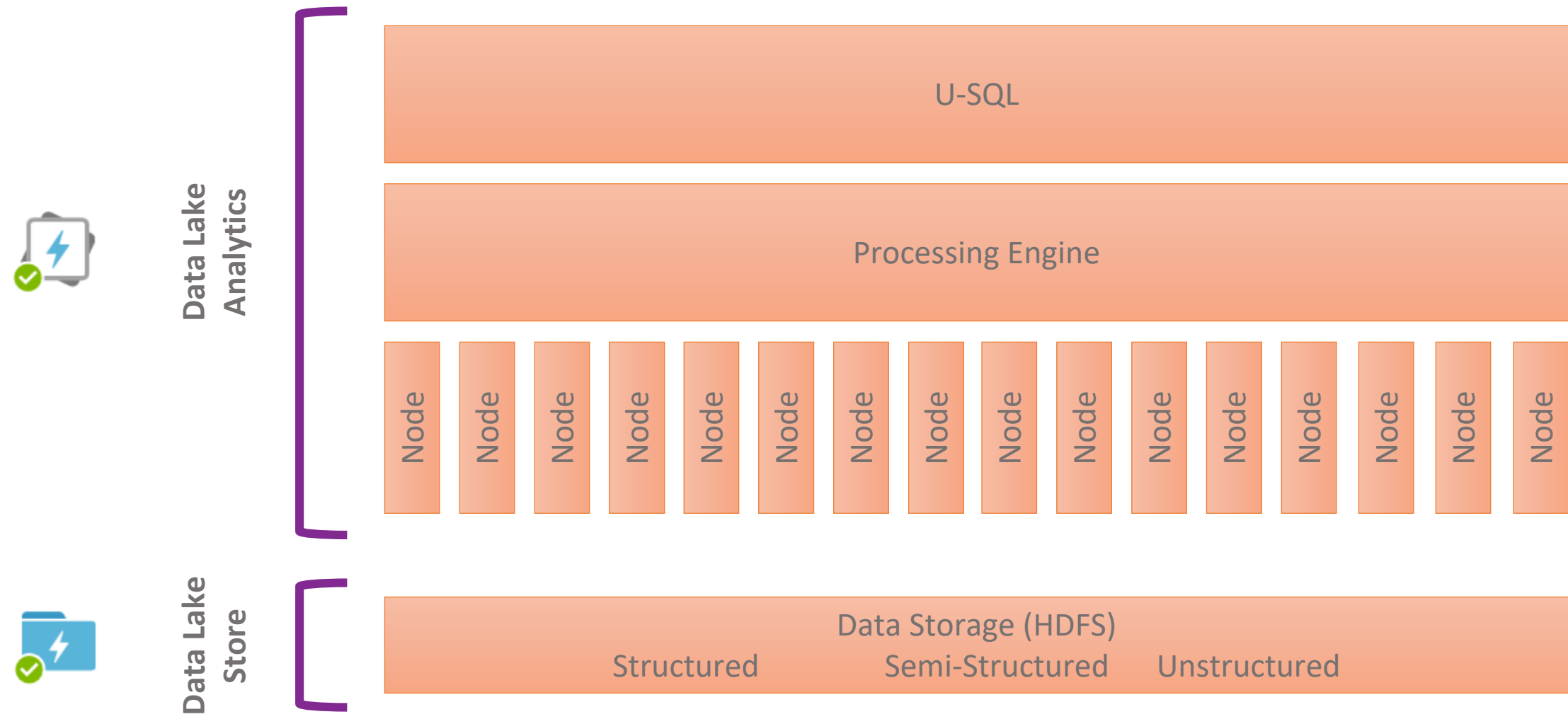


<https://azure.microsoft.com/en-gb/solutions/data-lake/>



What is Azure Data Lake?

The non product orientated version...



What is Azure Data Lake?
Storage & compute



Why use it?

Real world examples

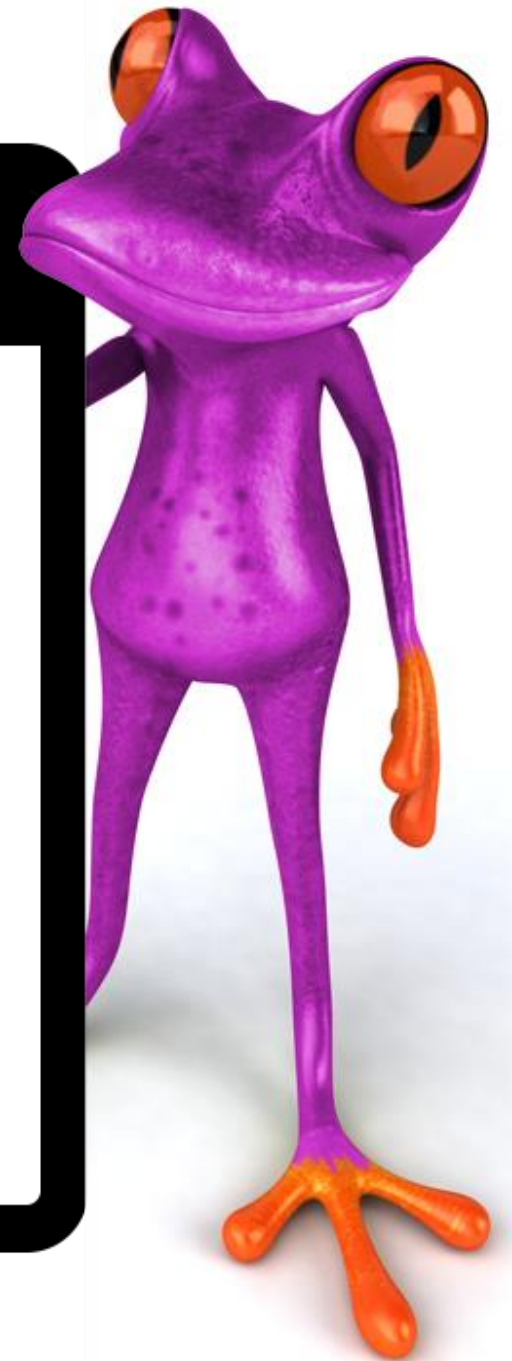
Working with each service

U-SQL

The language (T-SQL, C#)

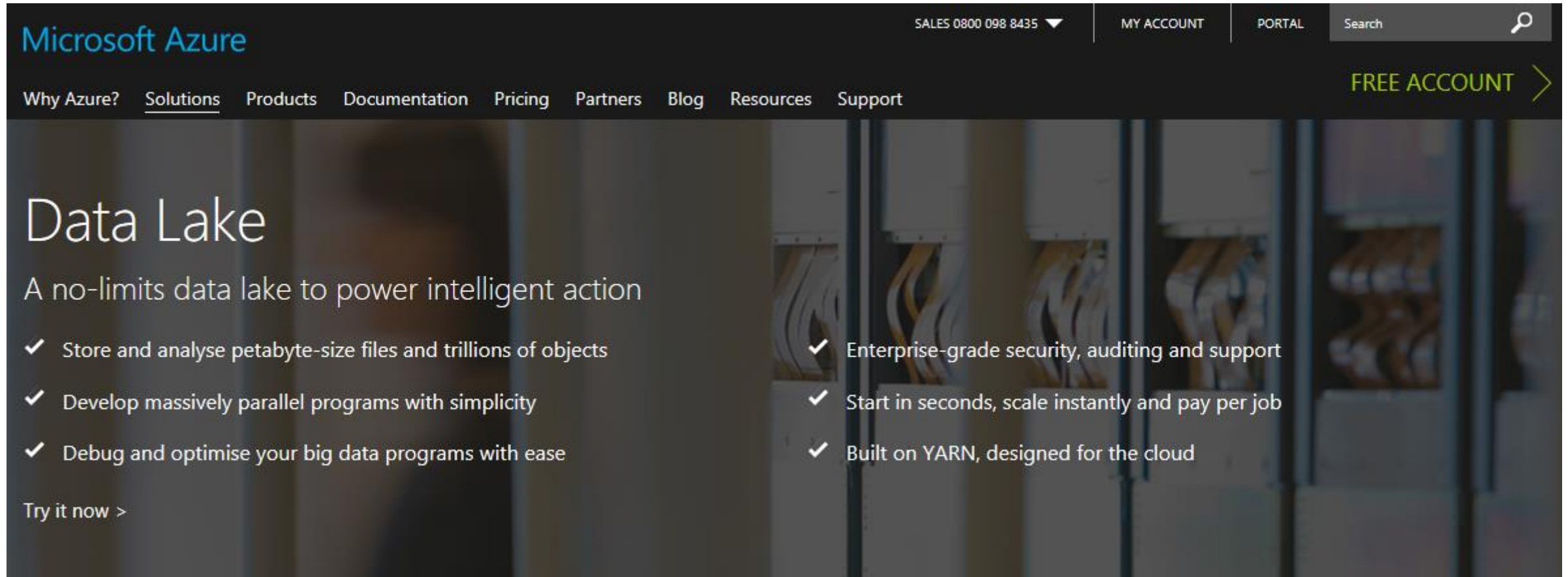
Database objects

Advanced capabilities



Why Use Azure Data Lake?

The Microsoft version...



The screenshot shows the Microsoft Azure website's landing page for Data Lake. The header includes the Microsoft Azure logo, contact information (SALES 0800 098 8435), and links for MY ACCOUNT, PORTAL, and a search bar. A navigation menu lists various sections like Why Azure?, Solutions, Products, Documentation, Pricing, Partners, Blog, Resources, and Support. A prominent 'FREE ACCOUNT' button with a right arrow is also visible. The main content area features the title 'Data Lake' and the tagline 'A no-limits data lake to power intelligent action'. Below this, there are two columns of bullet points, each preceded by a checkmark, highlighting key features of the service. At the bottom left of the main content, there is a 'Try it now >' link.

Microsoft Azure

SALES 0800 098 8435 ▼ MY ACCOUNT PORTAL Search

Why Azure? Solutions Products Documentation Pricing Partners Blog Resources Support

FREE ACCOUNT >

Data Lake

A no-limits data lake to power intelligent action

- ✓ Store and analyse petabyte-size files and trillions of objects
- ✓ Develop massively parallel programs with simplicity
- ✓ Debug and optimise your big data programs with ease
- ✓ Enterprise-grade security, auditing and support
- ✓ Start in seconds, scale instantly and pay per job
- ✓ Built on YARN, designed for the cloud

Try it now >

<https://azure.microsoft.com/en-gb/solutions/data-lake/>



purplefrogsystems.com/paul

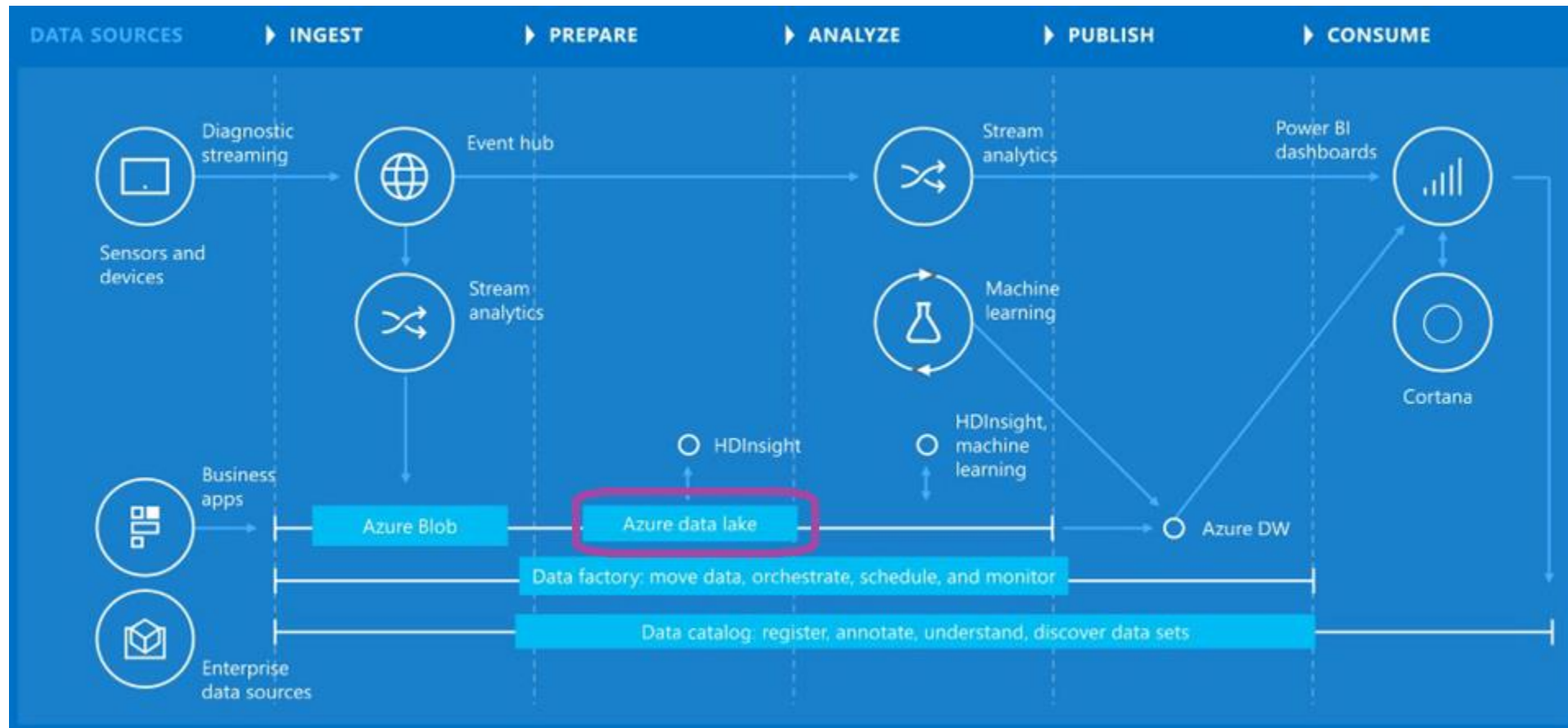


@mrpaulandrew



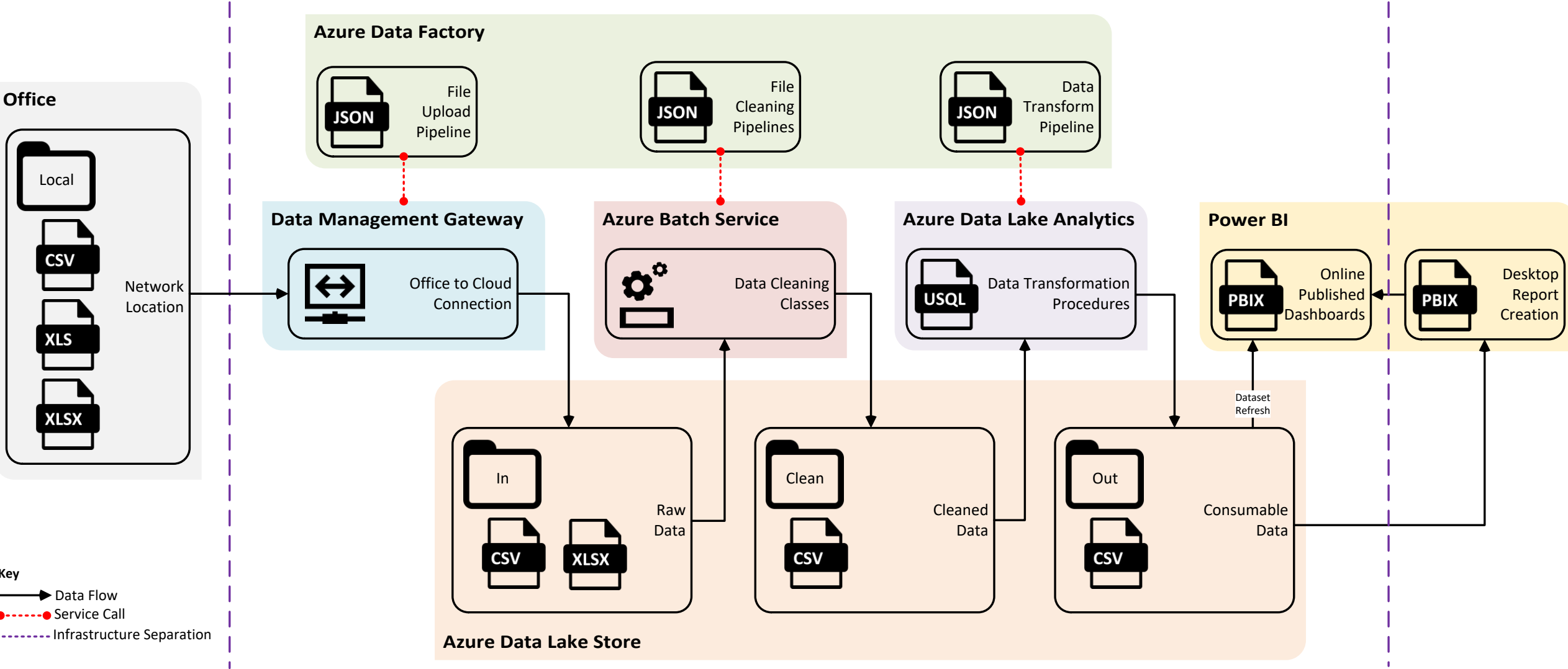
Azure Data Lake within the Cortana Intelligence Data Flow

- How does a lake fit into our data platform architecture?
- Is Data Lake going to run in isolation or be part of a larger pipeline?



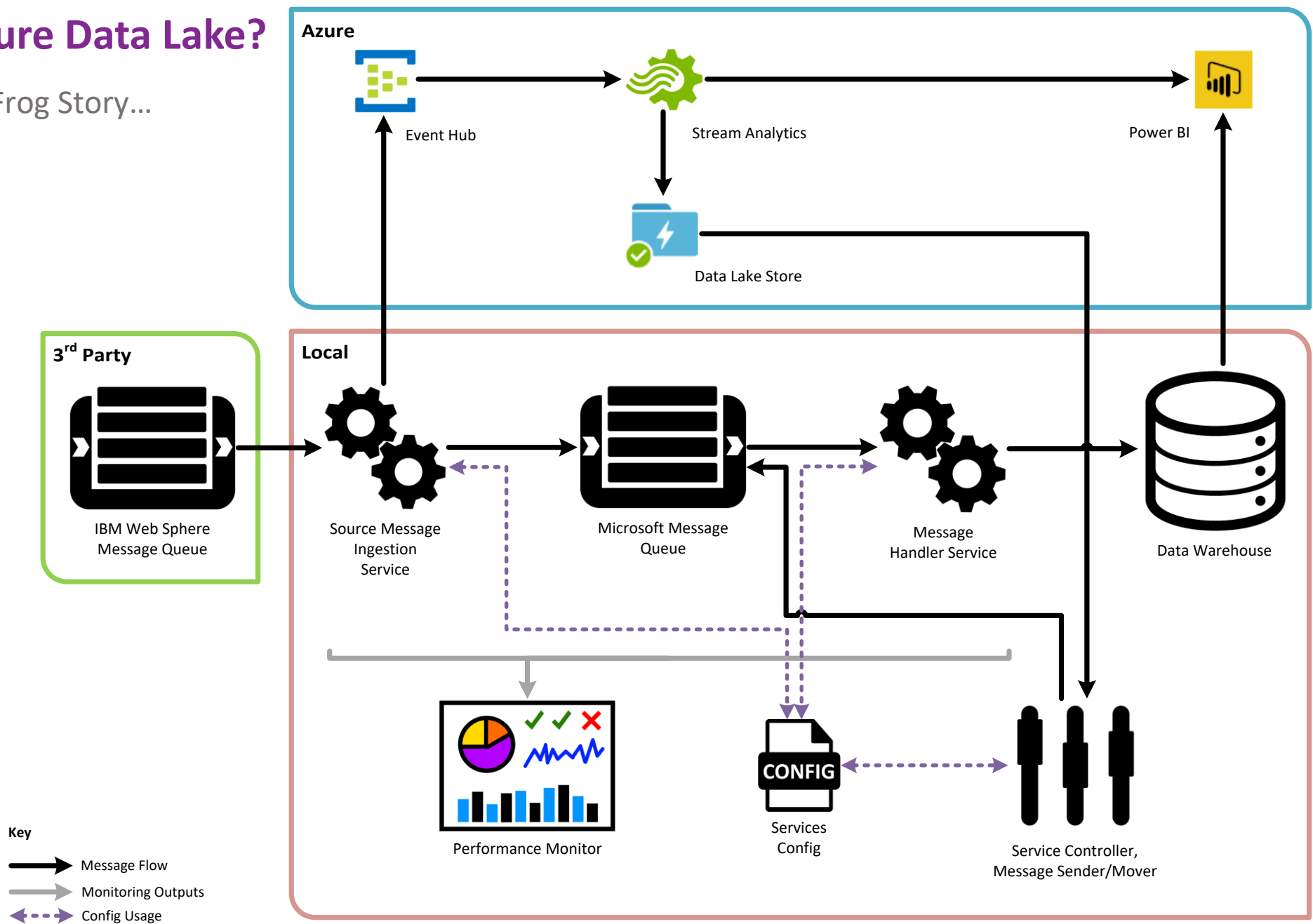
Why Use Azure Data Lake?

A Purple Frog Story...



Why Use Azure Data Lake?

Another Purple Frog Story...

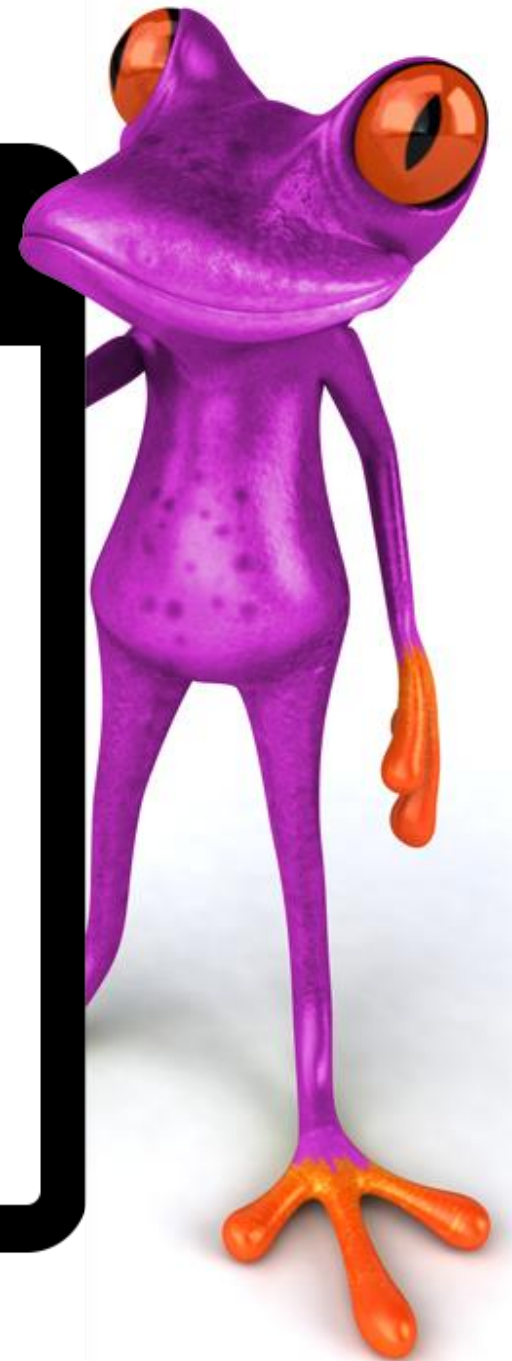


What is Azure Data Lake?
Storage & compute

Why use it?
Real world examples

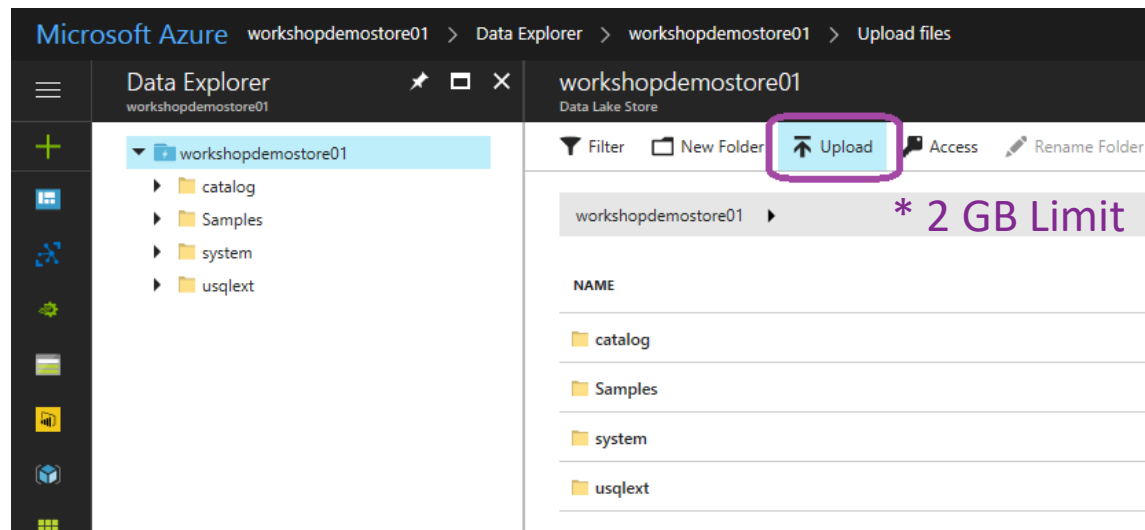
 **Working with each service**

U-SQL
The language (T-SQL, C#)
Database objects
Advanced capabilities

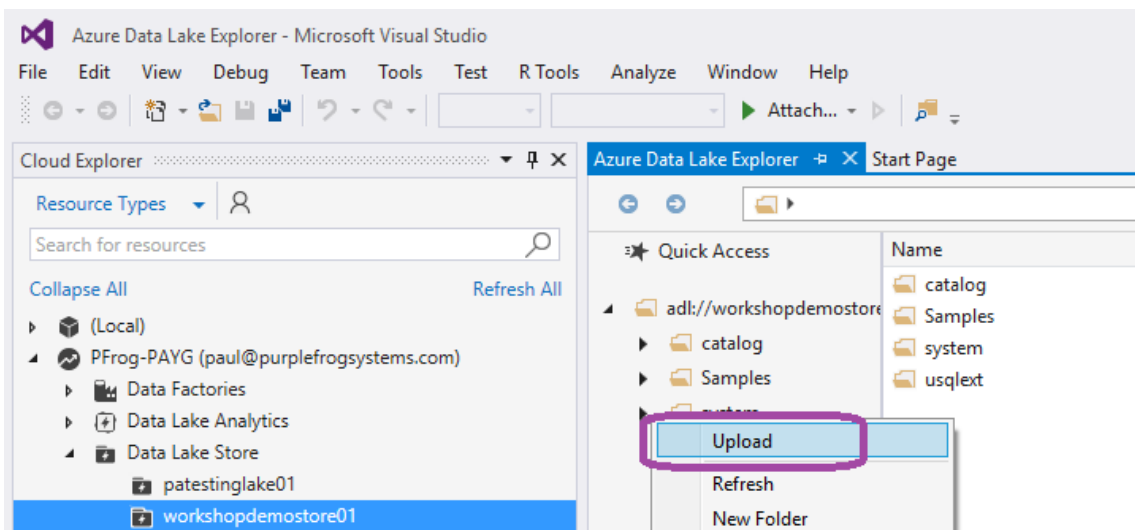


Uploading Data Into Azure Data Lake Store

Azure Portal UI



Visual Studio



Azure Data Factory

```
{
  "name": "DataLakeUploads",
  "properties": {
    "activities": [
      {
        "type": "Copy",
        "inputs": [
          {
            "name": "OnPremFileServer"
          }
        ],
        "outputs": [
          {
            "name": "DataLakeStore"
          }
        ]
      }
    ]
  }
}
```

//etc...

Power Shell

```
Import-AzureRmDataLakeStoreItem
-AccountName $dataLakeStoreName
-Path "C:\Temp\SampleData.txt"
-Destination $myrootdir\Samples\Data.txt
```

.Net SDK

```
var parameters = new UploadParameters(
    srcFilePath,
    destFilePath,
    accName, isOverwrite: force);

var frontend = new DataLakeStoreFrontEndAdapter(
    accName, adlsFileSstemClient);

var uploader = new DataLakeStoreUploader(parameters,
    frontend);

uploader.Execute();
```



Triggering Azure Data Lake Analytics Jobs

Azure Portal UI

Microsoft Azure workshopdemoanalytics01 - Jobs > SearchLog-1a-AdditionalOutputOptions > New U-SQL Job

New U-SQL Job

Submit Job Data Explorer Open File Save As

* Job Name SearchLog-1a-AdditionalOutputOptions Priority 1000 AUs 1 Estimated Cost 0.03 USD/minute

```
1 @searchlog =
2   EXTRACT UserId int,
3           Start DateTime,
4           Region string,
5           Query string,
6           Duration int?,
7           Urls string,
8           ClickedUrls string
9   FROM "/Samples/Data/SearchLog.tsv"
10  USING Extractors.Tsv(quoting : true, skipFirstNRows : 1);
```

Visual Studio

SearchLog-1a-Addi...utputOptions.usql X Azure Data Lake Explorer

Submit workshopdemoar master dbo

```
1 @searchlog =
2   EXTRACT UserId int,
3
4
5
6
7
8
9 FROM "
10 USING
11
12 OUTPUT @se
13 TO "/outpu
14 USING Outp
15
```

Submit Job

Azure User: paul@purplefrogsystems.com

Script Name: SearchLog-1a-AdditionalOutputOptions.usql

Job Name: SearchLog-1a-AdditionalOutputOptions

Analytics Account: workshopdemoanalytics01

Parallelism: 5 / 250

Advanced

Submit

Azure Data Factory

```
{
  "name": "DataLakeProcessing",
  "properties": {
    "activities": [
      {
        "type": "DataLakeAnalyticsU-SQL",
        "typeProperties": {
          "script": "@Data = SELECT * FROM @CSVFile",
          //OR "script": "[dbo].[usp_ProcessData](@TimeSlice)",
          //OR "scriptPath": "BlobStore\\ProcessData.USQL",
          "scriptLinkedService": "BlobStore",
          "degreeOfParallelism": 2,
          "priority": 7
        }
      }
    ]
  }
}
```

Power Shell

```
Submit-AzureRmDataLakeAnalyticsJob
-Name "ProcessData"
-AccountName $dataLakeAnalyticsName
-ScriptPath $usqlScript
```

.Net SDK

```
var script = File.ReadAllText(scriptPath);
var jobId = Guid.NewGuid();
var properties = new USqlJobProperties(script);
var parameters = new JobInformation(jobName, JobType.USql,
    properties, priority: 1, degreeOfParallelism: 1,
    jobId: jobId);

var jobInfo = _adlaJobClient.Job.Create(_adlaAccountName,
    jobId, parameters);

return jobId;
```



Job Management – Data Flow Graphs

Job Browser: [redacted] - Microsoft Visual Studio

File Edit View Debug Team Tools Test Tools Analyze Window Help

Cloud Explorer

Resource Types [icon] Search for resources

Collapse All Refresh All

(Local)

Data Factories

Data Lake Analytics

Databases

Jobs

Linked Storages

Data Lake Store

Network Interfaces

Network Security Groups

Public IP Addresses

SQL Databases

SQL Servers

Storage Accounts

Virtual Machines

Virtual Networks

Actions Properties

Open Job Explorer

Server Explorer Toolbox Test Explorer SQL Server Cloud...

Error List Output Find Results 1 Package Manager Console Azure Data Lake Explorer Task List Local Run Results

Job Browser [redacted]

My All Search [icon]

Name	Submitter	Priority	SubmitTime
ADF-90f3241f-d183-4098-bfc4-efdbt	alex@purplefrogsystems.com	9	7/13/2017 11:18:36 AM
ADF-90b37a1d-c6a8-4edd-b069-d36	alex@purplefrogsystems.com	9	7/13/2017 7:18:34 AM
ADF-596331fa-5724-455c-b3f8-7b9e	alex@purplefrogsystems.com	6	7/13/2017 10:06:28 AM
ADF-b08c0559-bfd9-4bc1-9909-67b	alex@purplefrogsystems.com	7	7/13/2017 10:05:19 AM
ADF-f8f09d63-9c6f-49ba-aefc-92cdt	alex@purplefrogsystems.com	7	7/13/2017 10:03:59 AM
ADF-f7675c0b-4b2a-4c73-bffc-11a9	alex@purplefrogsystems.com	7	7/13/2017 10:04:31 AM
ADF-162b65fd-ed2a-461c-b398-6b2	alex@purplefrogsystems.com	9	7/13/2017 6:23:30 AM
ADF-6f8a7d68-044d-4b7c-8607-790e	alex@purplefrogsystems.com	9	7/13/2017 3:18:27 AM
ADF-209dc118-a387-4892-946b-466e	alex@purplefrogsystems.com	10	7/13/2017 3:18:59 AM
ADF-9f2b0431-6268-4ce6-b36a-a647	alex@purplefrogsystems.com	10	7/13/2017 3:18:22 AM
ADF-f3ff589c-73d9-4a8f-be08-0e32a	alex@purplefrogsystems.com	10	7/13/2017 3:18:59 AM
ADF-6989f2dd-7fbd-4705-b7da-c87f	alex@purplefrogsystems.com	10	7/13/2017 3:07:51 AM
ADF-7d2c681a-c295-42fe-a224-4560	alex@purplefrogsystems.com	7	7/13/2017 1:10:44 AM
ADF-d161fa68-bb68-4e65-8690-5c6t	alex@purplefrogsystems.com	9	7/12/2017 6:33:48 PM
ADF-dd861374-33ca-4864-8539-cda	alex@purplefrogsystems.com	10	7/12/2017 3:04:25 PM
InstallCheck_with_installname		1000	7/12/2017 5:18:11 PM
InstallCheck_with_installname		1000	7/12/2017 5:14:39 PM
InstallCheck_with_installname		1000	7/12/2017 5:13:28 PM
InstallCheck_with_installname		1000	7/12/2017 5:11:18 PM
InstallCheck_with_installname		1000	7/12/2017 5:10:29 PM
InstallCheck_with_installname		1000	7/12/2017 5:05:03 PM
ADF-892e0d06-c4c2-44a2-ba01-318	alex@purplefrogsystems.com	9	7/12/2017 7:10:57 AM
ADF-824386e8-ab1b-4b80-aaf3-e02f	alex@purplefrogsystems.com	6	7/12/2017 10:06:08 AM
ADF-edd6cfff-06a1-4aba-9271-6833	alex@purplefrogsystems.com	7	7/12/2017 10:05:21 AM
ADF-afd845df-d501-4d0f-9b66-8741	alex@purplefrogsystems.com	7	7/12/2017 10:04:44 AM
ADF-2c2f794b-d43d-455a-bfc3-409c	alex@purplefrogsystems.com	7	7/12/2017 10:02:35 AM
ADF-3e5f1ce5-8cd3-4b79-b022-a2e	alex@purplefrogsystems.com	9	7/12/2017 3:10:56 AM
ADF-05eb8776-dc42-4aac-bf91-7d3f	alex@purplefrogsystems.com	10	7/12/2017 3:23:39 AM
ADF-f7bc35c4-8162-4a38-a51b-7d8c	alex@purplefrogsystems.com	10	7/12/2017 3:24:42 AM
ADF-64db4c0b-65fb-43d1-84ac-a57f	alex@purplefrogsystems.com	10	7/12/2017 3:17:33 AM
ADF-bc5f5f10-0305-438a-a7e4-bd3e	alex@purplefrogsystems.com	10	7/12/2017 3:10:50 AM
ADF-c16937c9-5777-4e7f-a501-4e74f	alex@purplefrogsystems.com	10	7/12/2017 3:14:36 AM
ADF-e0048803-b951-4598-8b3f-aa2c	alex@purplefrogsystems.com	10	7/12/2017 3:11:30 AM
ADF-703f57fa-58f1-4071-9b65-a8cf7	alex@purplefrogsystems.com	7	7/12/2017 1:11:10 AM
ADF-1e5aad76-9c42-4cd1-b585-3cb	alex@purplefrogsystems.com	6	7/11/2017 10:08:00 AM
ADF-83335f57-2a49-45b5-a6fe-7a1a	alex@purplefrogsystems.com	7	7/11/2017 10:04:13 AM
ADF-30b120b2-a6e1-4781-8660-a18	alex@purplefrogsystems.com	7	7/11/2017 10:05:59 AM

Showing 301 results [More...](#)

Job Name: ADF-90f3241f-d183-4098-bfc4-efdb6d689

Job Summary

Preparing Queued Executing Finalizing

1.9 minutes 12 seconds 48.56% N/A

Job Result

Total Duration

N/A

Submit Time

Start Time

End Time

13/07/2017 11:18:36 13/07/2017 11:20:41 N/A

Compilation

Queued

Running

1.9 minutes 12 seconds 25.8 minutes

Account

Author

alex@purplefrogsystems.com

Priority

Parallelism

9 2

Bytes Left

Bytes Read

Bytes Written

2,185,327 655,737,372 696,965,085

Total Vertices

Completed

Running

Failed

7,334 688 2 0

Job Detail

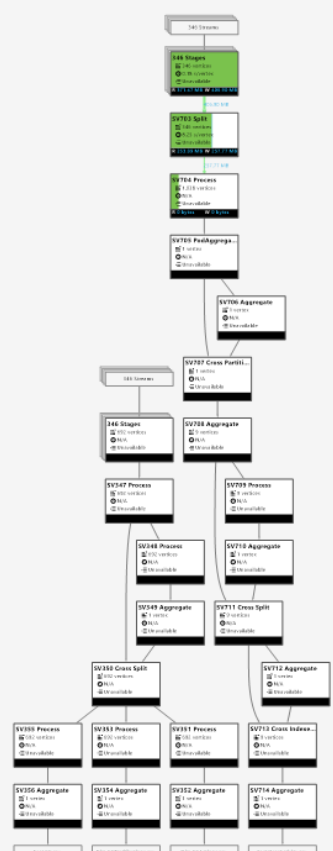
[Script](#) [Resources](#) [Vertex Execution View](#)

Job Graph Metadata Operations Data State History Diagnostics

Search stage

Display: Progress Succeeded Retried Failed Running Waiting

Expand All



Job Playback 00:00:00



Job Management – AU Modelling

Visual Studio

File Edit View Debug Team Tools Test R Tools Analyze Window Help

Cloud Explorer

Resource Types

Search for resources

Collapse All Refresh All

(Local)

Data Factories

Data Lake Analytics

Databases

Jobs

Linked Storages

Data Lake Store

Network Interfaces

Network Security Groups

Public IP Addresses

SQL Databases

SQL Servers

Storage Accounts

Virtual Machines

Virtual Networks

Actions Properties

Open Job Explorer

Server Explorer Toolbox Test Explorer SQL Server Cloud...

Job Browser: ADF-596331fa-5724-455c-b3f8-7... AU Usage View

AU Usage Stage AU Usage AU Usage Modeler

This view models the job execution for a given resource allocation.

(DISCLAIMER: This calculation is a coarse estimation based on vertex dependencies and previous vertex execution time.)

Estimated best time (sec): 176.83, max useful AUs: 2124

Number to model: 35 (enter an allocation and press enter to model)

AUs in use

Time (sec)

Legend

AU Usage

Ready

Job Management – AU Polices & Usage Costs

Microsoft Azure

Data Lake Analytics

Save

Discard

Name	Status	Location	Created
	Running	eastus2	09/01/2017 09:33:34 AM

Subscription Name

Piriform Pay-As-You-Go

Change subscription

Subscription Id

Debug Data Access Levels

Who has permission to view debug vertex data in Visual Studio

Users of this account and Microsoft Support Engineers

Maximum AUs

Set the maximum AUs for this account.

250

AU

Maximum Number of Running Jobs

Set the maximum number of jobs that you can run at the same time.

10

Jobs

Days to Retain Job Queries

Set the number of days to keep U-SQL job resources (scripts, DLLs, graphs) in your Data Lake Store account.

30

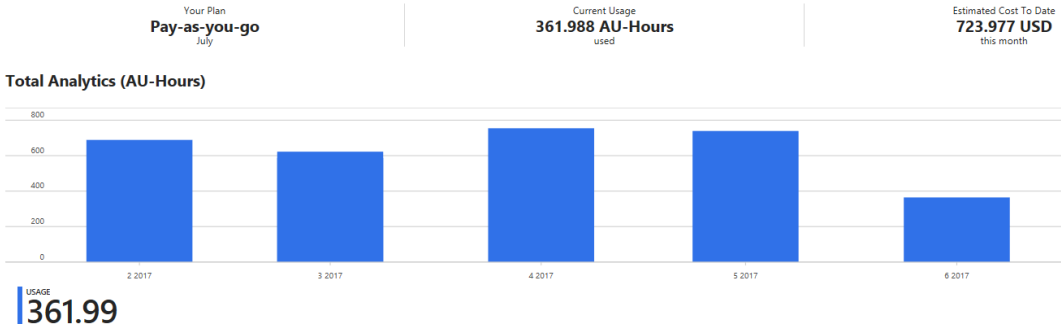
Days

Job Submission Limits

Set the maximum amount of AUs and the highest priority each user has access to when they submit a job. You can override the default by adding policies for different users, and groups. If a submitter has multiple policies that apply to them, the most permissive limit will take effect.

Add Policy

Bulk buy your compute!



Change Your Plan

Select your monthly plan

Pay-as-you-go

Pay-as-you-go

Monthly Plans

100 AU-Hourss for 100 USD

500 AU-Hourss for 450 USD

1,000 AU-Hourss for 800 USD

5,000 AU-Hourss for 3,600 USD

10,000 AU-Hourss for 6,500 USD

50,000 AU-Hourss for 29,000 USD

100,000 AU-Hourss for 52,000 USD

500,000 AU-Hourss for 234,000 USD

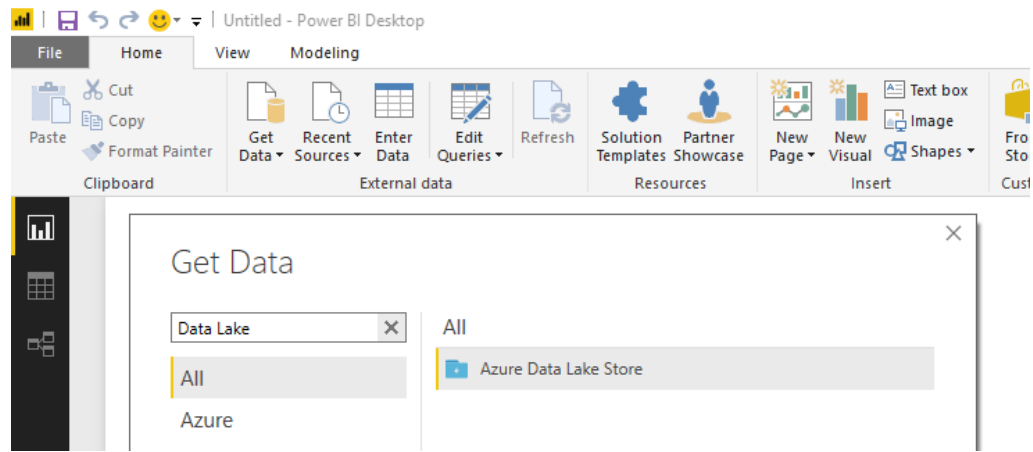
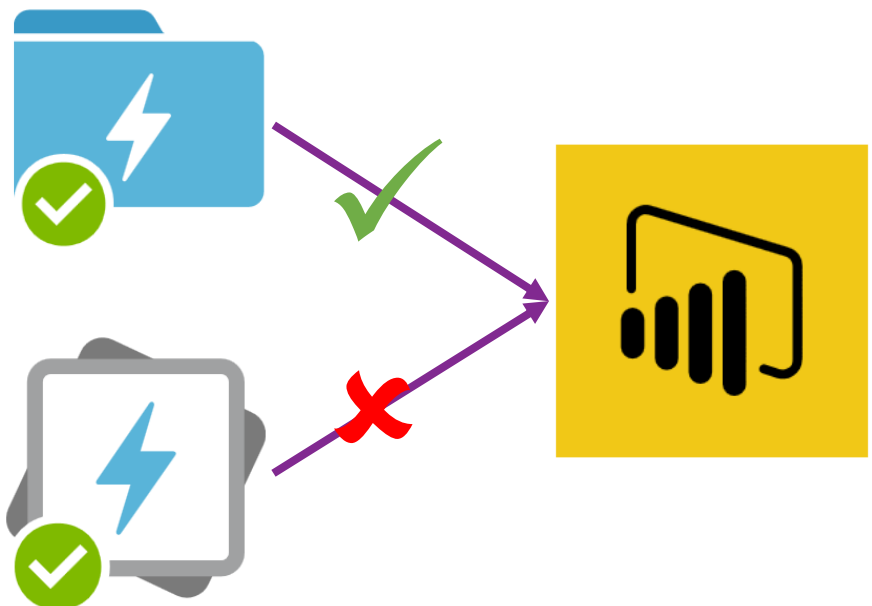
Pay-as-you-go
400 AU's
800 USD

Pay Monthly
500 AU's
450 USD

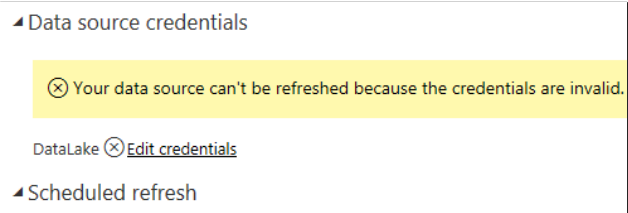
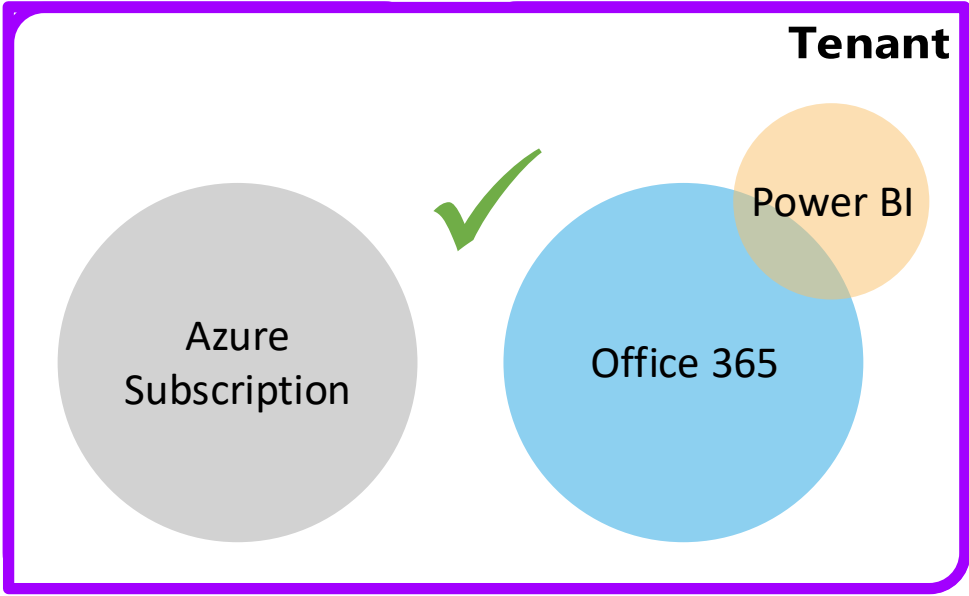


Consuming Data Lake with Power BI

Only storage connector available at present.



Published PowerBI.com dataset refresh across tenants.



*“Failed to update data source credentials:
The credentials provided for the DataLake source are invalid.”*

<https://www.purplefrogsystems.com/paul/2017/06/connecting-power-bi-to-azure-data-lake-store-across-tenants/>



What is Azure Data Lake?
Storage & compute

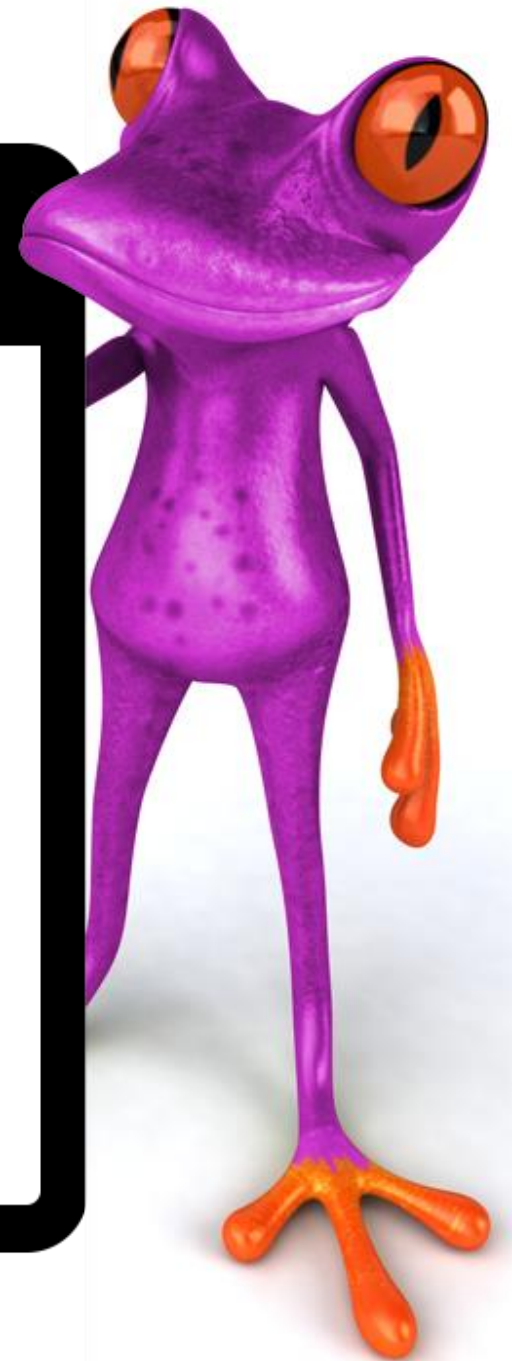
Why use it?
Real world examples

Working with each service



U-SQL

The language (T-SQL, C#)
Database objects
Advanced capabilities



What is U-SQL?

- Hybrid of T-SQL and C#
- Created by Michael Rys

```
@FileSummary= SELECT CountryCode,
                  COUNT(*) AS RecordCount,
                  SUM(SMSIn) AS SMSIn,
                  SUM(SMSOut) AS SMSOut,
                  SUM(CallIn) AS CallIn,
                  SUM(CallOut) AS CallOut,
                  SUM(Data) AS Data
FROM @RawData
GROUP BY CountryCode;
```

```
@DataPrep01 =
SELECT *
    , DateTime.Parse("1970-01-01").AddSeconds(TimeMS/1000) AS StartWindowDT
    , DateTime.Parse("1970-01-01").AddSeconds(TimeMS/1000).AddMinutes(10) AS EndWindowDT
FROM @RawData;
```



@MikeDoesBigData
#USQL

Why 'U'?

- Unifies T-SQL and C#
- Dive into the data lake like a U-Boat.
- U is the next letter after T-SQL.



@PurpleFrogSys

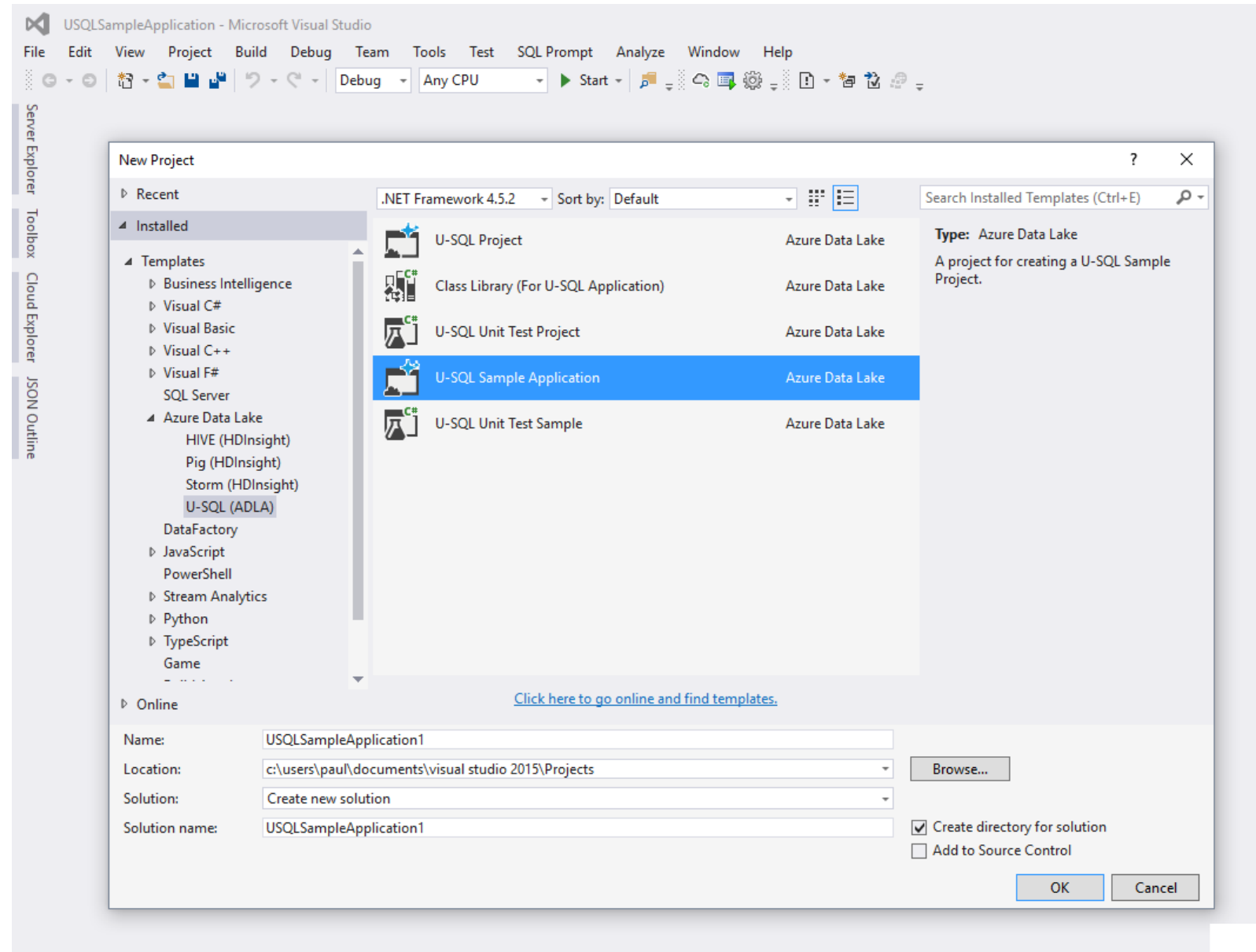


www.purplefrogsystems.com



Developing U-SQL – The Tools

- Visual Studio 2013
 - Azure SDK – ADL Tools
- Visual Studio 2015
 - Azure SDK – ADL Tools
- Visual Studio 2017
 - Out of the box extensions
- Azure Portal
 - Directly with blades
- Visual Studio Code
 - Extension now GA



U-SQL, like T-SQL

- Based on T-SQL
SELECT
DISTINCT
SUM()
FROM
GROUP BY
ORDER BY
Windowing Functions – ROW_NUMBER() etc.

```
@FileSummary= SELECT CountryCode,  
                COUNT(*) AS RecordCount,  
                SUM(SMSIn) AS SMSIn,  
                SUM(SMSOut) AS SMSOut,  
                SUM(CallIn) AS CallIn,  
                SUM(CallOut) AS CallOut,  
                SUM(Data) AS Data  
FROM @RawData  
GROUP BY CountryCode;
```

U-SQL, but with C#

- Uses .Net Data Types

bool	decimal	int	short	ushort
byte	double	long	uint	string
char	float	sbyte	ulong	

- Embeds C# functions

```
DateTime.Parse("1970-01-01").AddSeconds(TimeMS/1000) AS
```

- Access C# Assemblies

```
USQLApplication2.Udfs.ValidateDate(CustomerDoB) AS CustomerDoB
```



T-SQL vs U-SQL

T-SQL

CASE WHEN Gender IN ('M', 'F') THEN Gender ELSE '-' END

SubString(Field,0,10)

CharIndex('?',Field)

IsNull(Field,'Unknown')

Convert(Varchar, OrderDateTime, 112)

Replace(Field,'?','Unknown')

U-SQL

(Gender=="M" || Gender=="F") ? Gender : "-" AS Gender

Field.SubString(0,10)

Field.IndexOf("?")

String.IsNullOrEmpty(Field) ? Field : "Unknown"

OrderDateTime.ToString("yyyyMMdd")

Field.Replace("?", "Unknown")



A Couple of Limitations

T-SQL

MERGE/UPDATE

WHILE

SELECT

 @Check = COUNT(0)

FROM

 [SomeWhere]

IF @Check > 5

Date Dimension

U-SQL

<https://www.purplefrogsystems.com/paul/2016/12/writing-a-u-sql-merge-statement/>

<https://www.purplefrogsystems.com/paul/2017/05/recursive-u-sql-with-powershell-u-sql-looping/>

<https://www.purplefrogsystems.com/paul/2017/02/creating-a-u-sql-date-dimension-numbers-table-in-azure-data-lake/>



Azure Data Lake Analytics Database Object (DDL and DML)

- Assemblies
 - Stored Procedures
 - Schemas
 - Tables
 - Functions
- Live in the ‘_catalog_’ directory of your storage.

```
CREATE DATABASE IF NOT EXISTS [Demo];
```

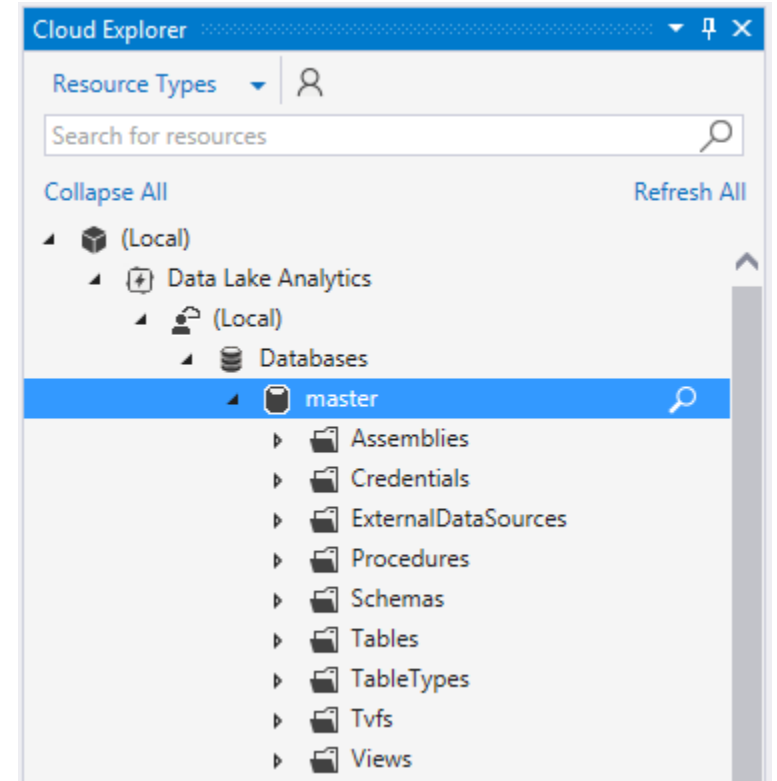
```
CREATE SCHEMA IF NOT EXISTS [fact];
```

```
DROP TABLE IF EXISTS [fact].[Sales];
```

```
CREATE TABLE IF NOT EXISTS [fact].[Sales];
```

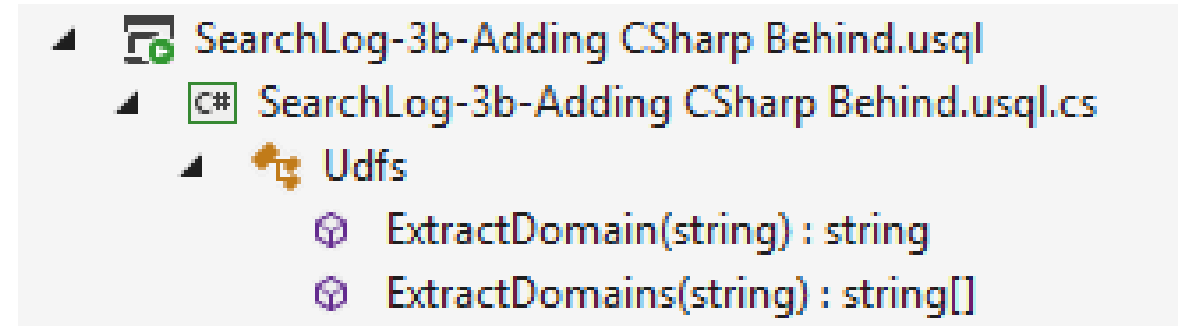
```
DROP PROCEDURE IF EXISTS [Demo].[fact].[usp_LoadSales];
```

```
CREATE PROCEDURE [Demo].[fact].[usp_LoadSales](@TimeSliceStart string)  
AS  
BEGIN /* do stuff */ END;
```



C# Code Behind

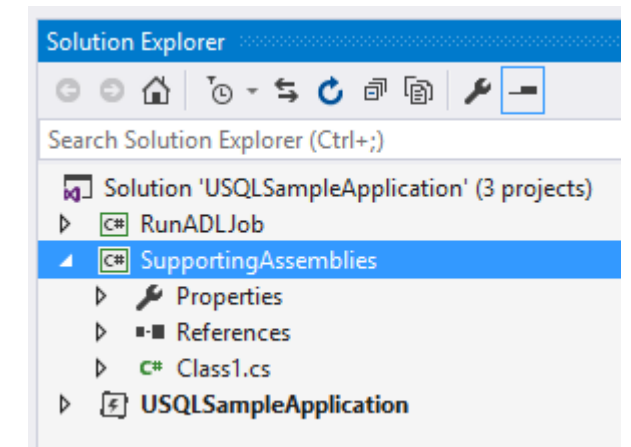
- Handled by Visual Studio when summiting jobs.
- U-SQL file does not carry C# around with it.
- Calling from other services it needs help compiling the C#.
- Calling U-SQL Stored procedures needs help.
- Assemblies are your friend 😊



```
CREATE ASSEMBLY IF NOT EXISTS [Demo].[USQLHelperClasses]  
FROM "Assemblies/USQLHelperClasses.dll";
```

Compiled C# Code in Assemblies

- Inline binary in U-SQL.
- Create the assembly from Azure Data Lake Store.
- Create the assembly from Azure Blob Storage.



Lets Try It



@PurpleFrogSys



www.purplefrogsystems.com



U-SQL Further Reading

Microsoft Blog – An Introduction to U-SQL in Azure Data Lake

<https://blogs.msdn.microsoft.com/robinlester/2016/01/04/an-introduction-to-u-sql-in-azure-data-lake/>

Microsoft Documentation – U-SQL Programmability Guide

<https://docs.microsoft.com/en-us/azure/data-lake-analytics/data-lake-analytics-u-sql-programmability-guide>

Microsoft MSDN – U-SQL Language Reference

[https://msdn.microsoft.com/en-US/library/azure/mt591959\(Azure.100\).aspx](https://msdn.microsoft.com/en-US/library/azure/mt591959(Azure.100).aspx)

SQL Server Central – Stairway to U-SQL

<http://www.sqlservercentral.com/stairway/142480/>

Purple Frog Blog

<https://www.purplefrogsystems.com/>

Stack Overflow – U-SQL Tag

<http://stackoverflow.com/questions/tagged/u-sql>



Thank You

