

Dipping Your Toes In: Azure Data Lake for DBAs

Bob Pusateri

About Bob Pusateri



**Microsoft
CERTIFIED**

Master

SQL Server 2008



@sqlbob



bob@bobpusateri.com



bobpusateri.com



bobpusateri



Specialties / Focus Areas / Passions:

- Performance Tuning & Troubleshooting
- Very Large Databases
- SQL Server Storage Engine
- HA/DR
- Cloud

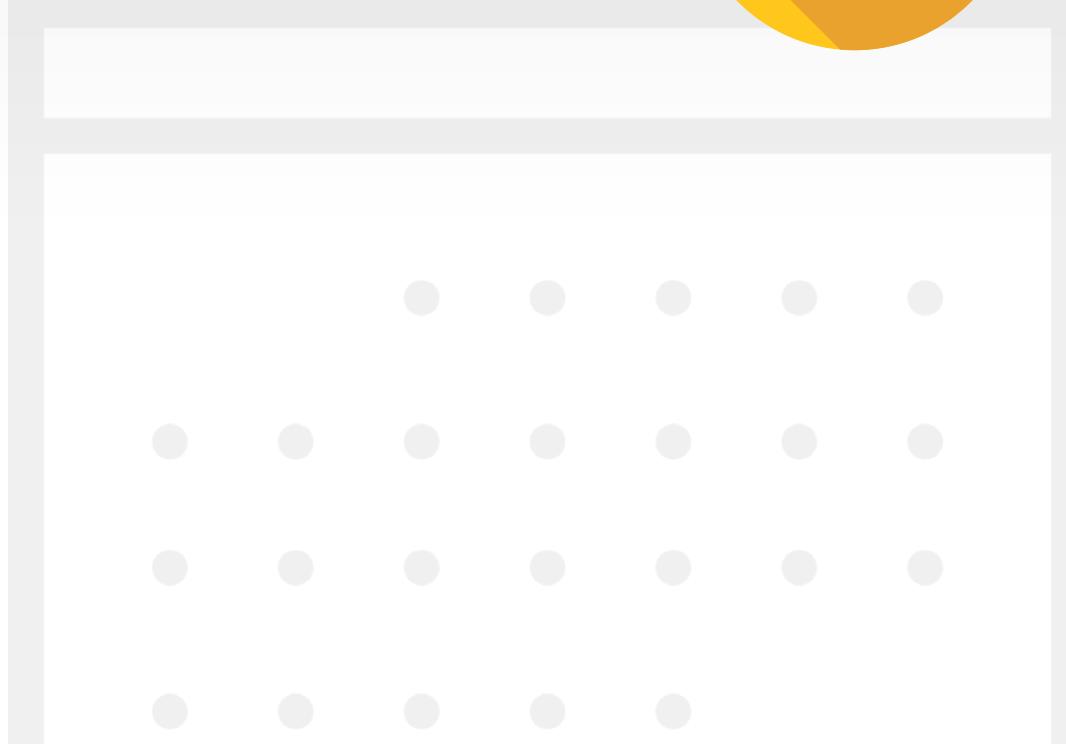
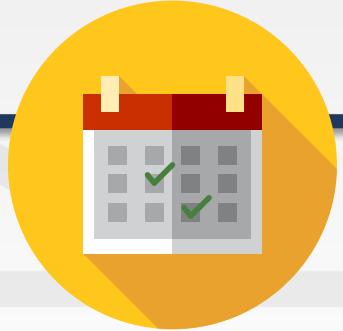




Your Feedback
Is Appreciated!

Agenda

- What Is Azure Data Lake?
- Data Lake Storage
- Data Processing
- Design Considerations
- Security
- DBA Topics
- Cost
- Demos



Data is...



About Your
Customers



Valuable
To
You



About Your
Business

Data is...



Valuable To
Others



YOUR
BUSINESS

THE NEW CURRENCY

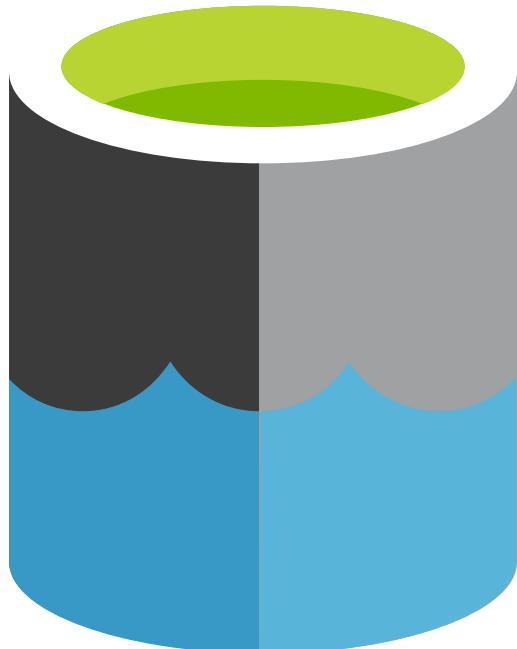




VAULT DOOR

WEIGHT: 22 1/2 Tons
THICKNESS: 22 Inches

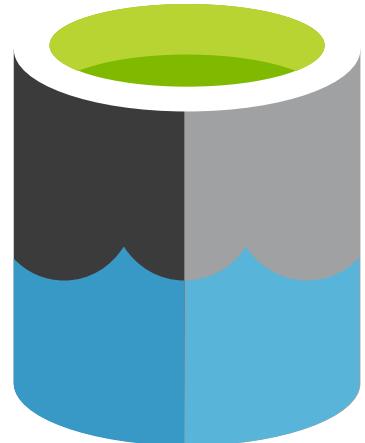
STEEL: 11 Layers of Special
Cutting and Drill Resistant
LOCKS: 4 Hamilton Watch
Movements for Time Locks



What is
Azure Data Lake?

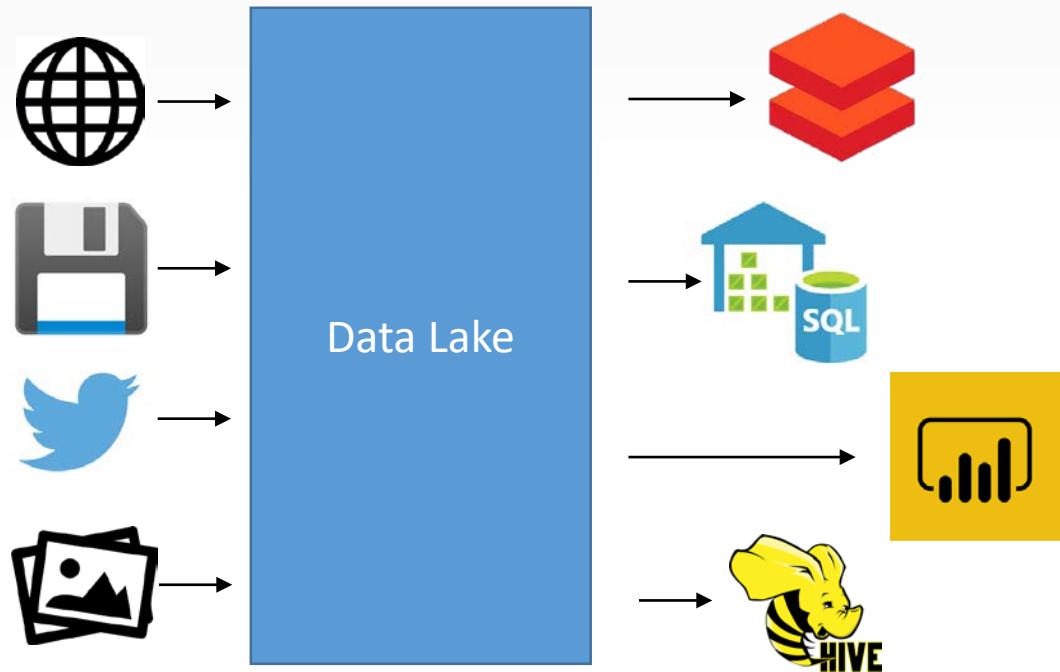
Azure Data Lake is...

- A scalable platform for housing all types of data for analysis
- A repository for analyzing many sources of data in its native format
- *TL;DR an extremely scalable storage system with helpful tools*



Azure Data Lake is...

- A tool that integrates with many services and sources
 - For both input and output



Azure Data Lake is...

- Performant – enables linear and infinite scale
- Scales on demand
- Supports varying access patterns
- Securable
 - RBAC – Groups and Principals
 - POSIX ACLs
- Manageable
 - Scriptable APIs
 - (Preferably REST APIs)

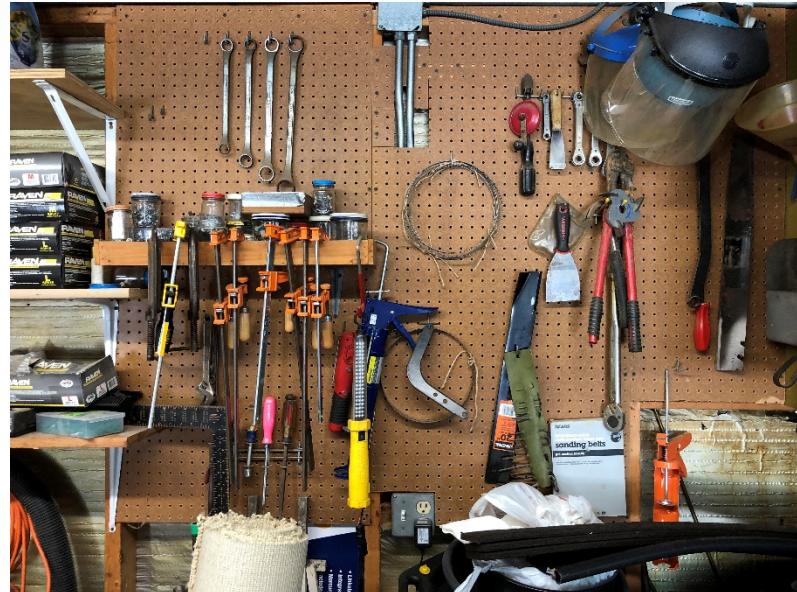


Azure Data Lake is...

- “Bring your own engine”
 - Hadoop
 - Apache Spark
 - Power BI
 - SQL Data Warehouse
 - Azure Data Lake Analytics
 - etc.
- There is no compute here

Azure Data Lake is...

- A great data tool to have in your toolbox
 - May be doing this work already with SSIS & Archiving
 - How much time do you spend writing packages?



OLTP vs. Data Warehouse

OLTP

- Focus
 - Operational Transactions
 - Writes
- Scope
 - Single database system
- Sample Objectives:
 - Recall a customer's orders
 - Generate a report

Data Warehouse

- Focus
 - Informational/Analytical
 - Reads
- Scope
 - Integrate multiple systems
- Sample Objectives:
 - Identify poorest-selling products
 - Compare margin per customer

What Goes In Azure Data Lake?

4 Vs of Big Data

- Volume
 - Growing exponentially every year!
- Variety
 - More than what fits neatly into a relational DB
 - Structured data being augmented by unstructured data
- Velocity
 - Frequency it needs to be processed. Is it a streaming source?
- Veracity
 - Is the data trustworthy?
- Value
 - Are we creating value from the above 4 things?

Structured vs. Unstructured Data

Structured: Relational



Unstructured: Binary or plain text



Semi-structured: Non-relational



Data Lake Storage

Azure Blob Storage (WASB)

- WASB = Windows Azure Storage Blob
- Optimized for Unstructured Data
 - Video, Images, Files
- Flat Structure
 - No folder tree*
- Lowest Cost (Cheaper than ADLS)
- Tiered
 - Can specify redundancy/performance/cost



Azure Storage Redundancy

- LRS – Locally Redundant Storage
 - Multiple copies in single data center
- ZRS – Zone-Redundant Storage
 - Multiple data centers within a zone
- GRS – Geo-Redundant Storage
 - Replicates data to a data center in another region
- RA-GRS – Read-access Geo-Redundant Storage
 - GRS, but can read from that remote region as well



Azure Storage Tiers

- Hot
 - Data that is accessed frequently
 - Highest storage cost. Lowest access cost.
- Cool
 - For less-frequently-accessed data. Minimum 30 day storage.
 - Lower storage cost. Higher access cost.
- Archive
 - Lowest storage cost. Highest access cost.
 - Slow to retrieve (hours). Minimum 180 day storage.
 - Comparable to Amazon Glacier

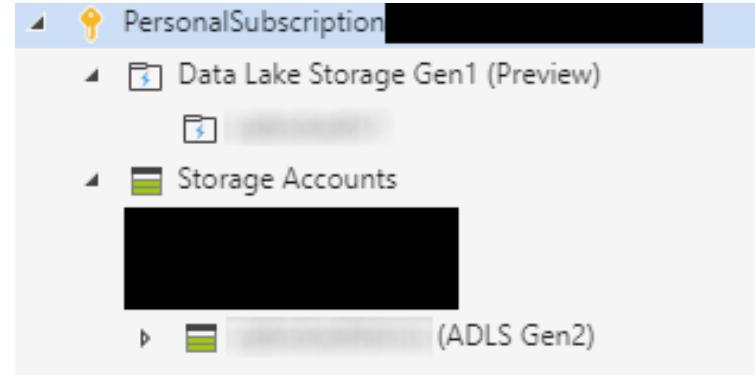


Azure Data Lake Gen 1 (ADLS)

- ADLS = Azure Data Lake Storage
- Interoperable
 - Works with all major MPP platforms
- Security
 - Posix ACLs & AAD integration
- Performance
 - Optimized for analytics workloads
 - No file size limit
- Folder Structure
 - A real hierarchical structure

Azure Data Lake Gen 2 (ABFS)

- ABFS = Azure Blob File System
 - AKA “ADLS Gen 2”
- Integration
 - Supports most* MPP & authentication schemes
- Hierarchical Namespace
 - Folder Structure & Atomic File Operations
- File System Tiering
- 30% price premium over ADLS



Comparing Storage Options

Object Storage (Azure Blob Storage)

- Flat storage space
- Data represented as blobs
- Highly scalable
- Cost-Effective

or

File System Storage Azure Data Lake Storage Gen 1

- Hierarchical storage space
- Data is blocks in a file system
- Security at folder/file levels
- Atomic & metadata operations

Azure Data Lake Storage Gen 2

- Get the features of file system storage
- Get the cost/scale features of blob storage

File Formats

- Text Files!

- Human readable
- Standardized
- Well supported
- .txt, .csv, etc.



- JSON

- Human readable
- Supports complex nesting
- Files can be large without compression



File Formats - Avro

- Two-part file format
 - Binary portion for data
 - JSON-based schema
- Supports schema evolution
- Requires a custom viewer utility



File Formats - Columnar

Parquet

- From Cloudera/Twitter



Orc

- From Hortonworks/Facebook



- Both support compression and predicate pushdown
- Writes are slower in both due to columnar format

File Formats - Columnar

- Generally better for analytical workloads
 - Test your workload!
- May be worth the effort if
 - Querying data multiple times
 - Need predicate pushdown or column elimination
- Can reduce/eliminate encoding issues with delimited formats

Zones

- Folders!
- Organization is key to success (with ADL and anything else..)
- Common Zones
 - archive, raw, reference, curated, temp, stage
- Other Zones
 - environment_dev, test, prod
- Sub-zones
 - Subsystems, users



Zones: Raw

- Data here is in its original state from source system
- Sorted by source system and schema
- Immutable – the key rule for this zone
 - Data is always appended here
- Can be partitioned by date

Zones: Curated

- Data here is in a transformed state
 - Similar to DW / Datamart
- End users can access here
 - Analytics is performed on this data
- Data is governed/blessed
- This zone is flexible based on usage & requirements
 - Data Science Sandbox
 - Datamarts
 - Outbound data for integration, Etc.

Zones: Temp

- Helpful when incoming data is mutable
- Full/Incremental loads arrive constantly
 - Data is batched together here for analytical purposes
 - Result can be more efficient for MPP
- Data can be compared against raw to calculate deltas
- Files can be archived/vacuumed/deleted per needs



Zones: Archive

- Data outside of analytical windows
- May need to be kept for compliance purposes
- Can restrict access to this via ACLs and ACEs
- Generally the last stop before removal or tiering out
- ABFS supports tiering within containers
 - Move to slower/cheaper storage and pay less!

Zones: Moving Data Between Them

- ADF is great at this
 - Lots of tools you can orchestrate with it
- Typically you don't move though, you transform & create new
- Databricks, Oozie
- MPP Engines
- AzCopy

Data Processing

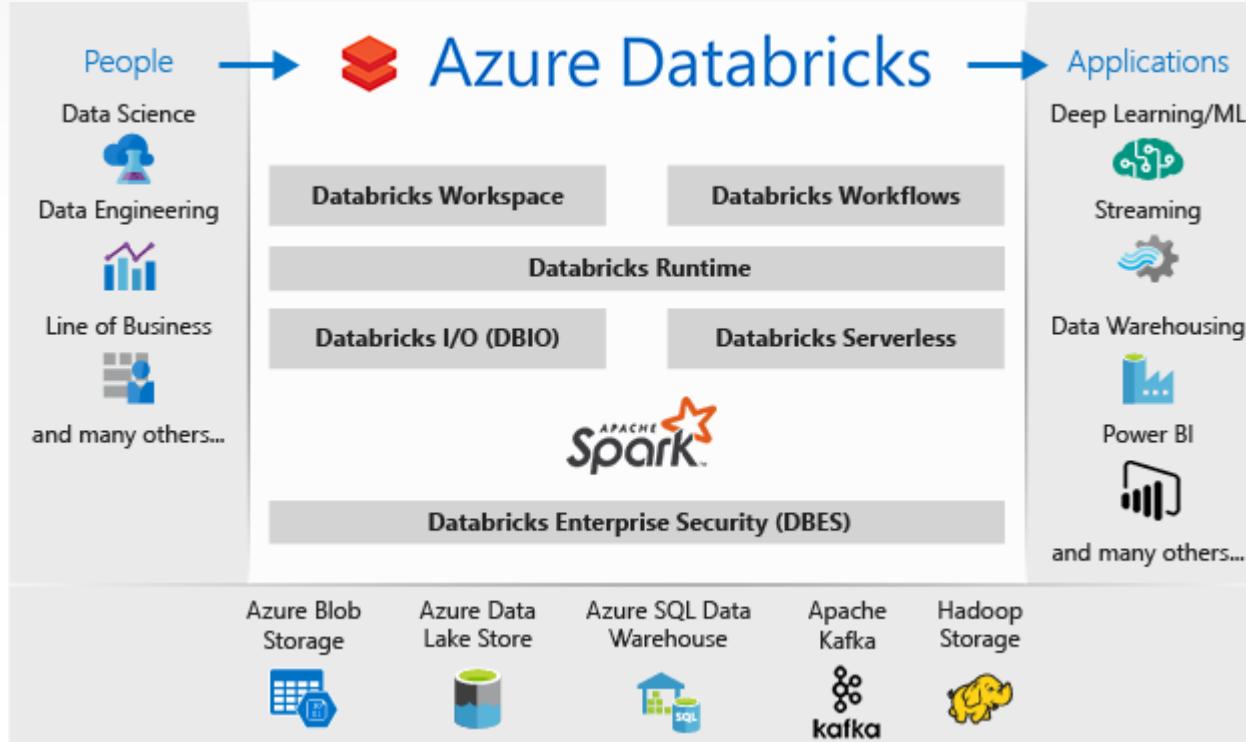
MPP Engines



U-SQL

Azure Databricks

- Cloud-based analytics platform based on Spark



Azure SQL Data Warehouse

- Cloud-based EDW that uses MPP
- Formerly SQL PDW
- Uses relational tables / columnar storage

U-SQL

- The native language of Azure Data Lake Analytics (ADLA)
- A combination of SQL and C#
- Can be executed in the cloud, or locally

U-SQL

```
@dataset =
    EXTRACT UserID      INT,
                StartDT     DATETIME,
                URL        STRING
    FROM @"/datafile.tsv"
    USING Extractors.Tsv();
```

```
OUTPUT @dataset
    TO @"/datafile_output.tsv"
    USING Outputters.Tsv();
```

Hive

- An ETL/Query tool that runs on top of Hadoop
- Has its own query language – HiveQL or HQL
 - Very similar to SQL

```
SELECT plate_number, COUNT(*)  
FROM Tickets;
```

Data Lake Design Considerations

File Naming

- Use Source and Date/Time into name
- Make Consistent
- This makes things easier, especially for self-service
- Example: ParkingTickets_20171031_2359.csv

Partitioning

- This helps make data “splitable”
- 1 Table per file becomes inefficient as size increases
 - Harder* have multiple nodes working against a single file
- Split file up by date within zone structure
- Thousands of files in same directory isn’t great either
- Example: raw/ParkingTickets/2017/10/ParkingTickets_20171031_2359.csv
- Avoid creating empty files/folders
- Avoid mixing types within folder hierarchy

Partitioning Example



raw

ParkingTickets

2017

10

ParkingTickets_20171031_2359.csv

Splitability

- Some file formats are splitable
 - MPP engine can split the file into segments for performance benefits
- Compression can render certain formats unsplitable
- This varies by MPP engine, check documentation
- If file format is unsplitable, you need to manage partitions manually
 - Balancing file sizes may matter here too for best performance
 - (Think of bad SQL Server statistics & CXPACKET waits)

Compressibility

- Compression can make formats unsplittable
- Some columnar formats have internal compression already
 - Sometimes you can layer these with GZIP et. al.
- TEST
 - Compression may not improve performance in all use cases



Compression Is A Tradeoff

- No Free Lunch!

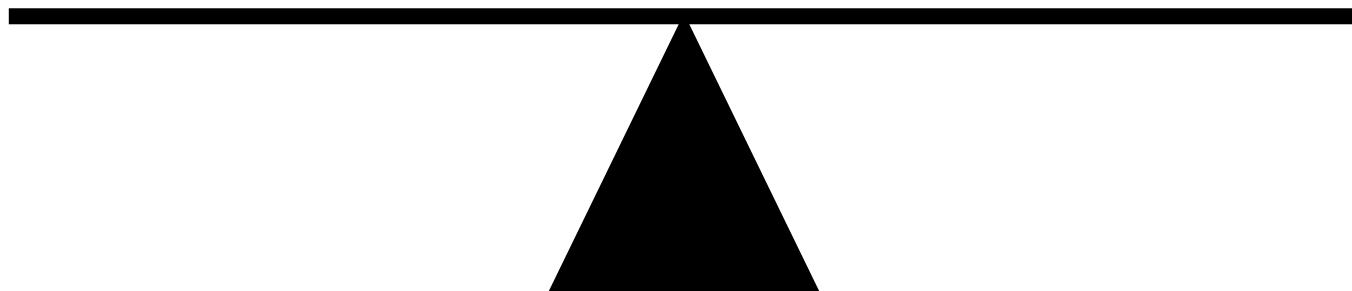
Storage Space

Memory

I/O

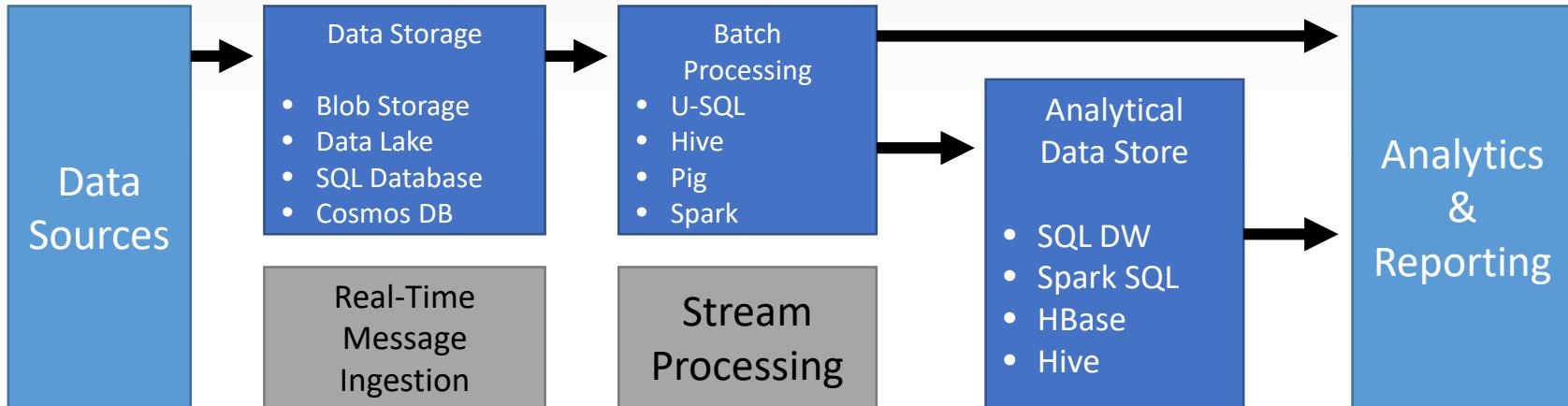
Network

CPU



Architectural Patterns: Batch

- Data streams in over time, is batched, and gets processed

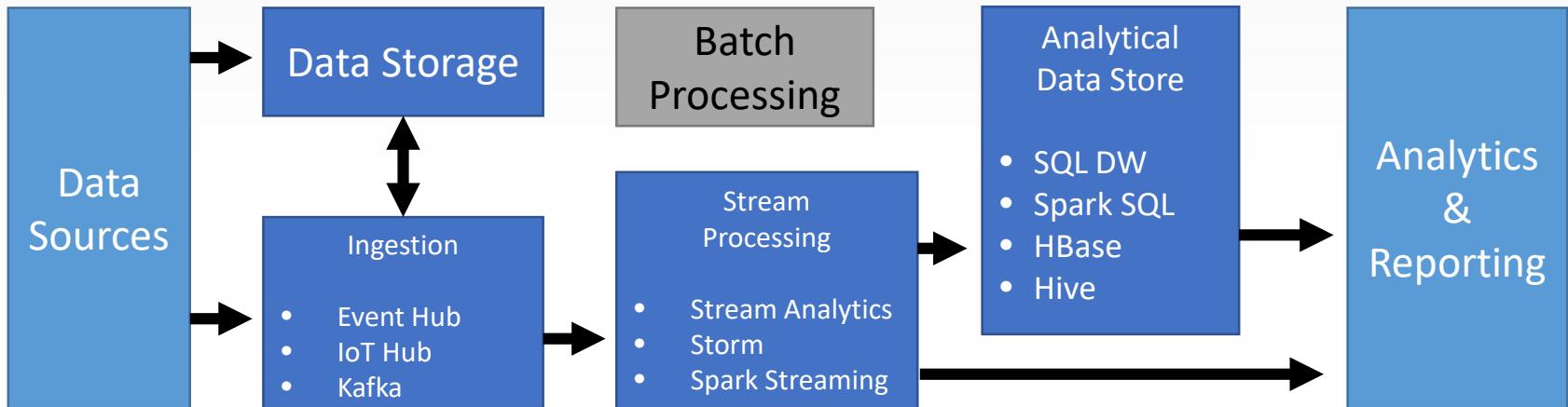


Orchestration

- Data Factory
- HDInsight

Architectural Patterns: Stream

- Data streams in and is processed in near real-time



Orchestration

Anti-Patterns: Data Swamp

- Multiple file types within a folder
- Poor Organization
- No standard naming
- No governance
- No data catalog
- **Makes self-service very difficult**

Anti-Patterns: Small Files

- Lots of small files can
 - Add overhead
 - Slow down MPP engines
- Optimal file size varies per compute engine
- Check your engine's documentation

Challenges

- By themselves, data lakes don't provide integrated views across org
- Hard to guarantee quality of data in a data lake
- Data lake may become a dumping ground for unused data
- Lack of descriptive metadata can make data hard to consume
- Lack of governance can lead to problems of varying types
- Data lakes are not the best way to integrate all types of data
 - Relational Data

Common Misconceptions

- Data lakes are for more than just IoT data!
- Can access a data lake with a relational database too
 - Polybase
 - Big Data Clusters

Security

Azure Data Lake Gen 1

- Authentication through Azure Active Directory
- Roles for account management
- POSIX ACL for data access
 - Unix-style read/write/execute permissions
 - ACLs can be enabled on folders or individual files
 - Permissions cannot be inherited from a parent object
 - Security groups also exist, can assign users to groups & grant that way

Azure Data Lake Gen 1

Access

Tickets_2003-03-01.csv (File)

Add Save Discard Advanced

Your permissions

bob@bobpusateri.com's effective permissions on this folder are: Read,Write,Execute.

Check access

You have superuser privileges on this account.

Manage access scope by creating a new access policy.

Owners

	Read	Write	Execute
Bob Pusateri bobpusateri_gmail.com#EXT#@bobpusateri.on...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
00000000-0000-0000-0000-000000000000	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

1 items (1 User)

NAME

Assigned permissions

No entries.

Everyone else

Users not covered above will be limited by these permissions

Azure Data Lake Gen 1



- Can enable a firewall
 - Only IP addresses in a defined range can access ADL
- Encryption
 - Can enable encryption before data is written on disk
 - TLS 1.2 for data in motion

Azure Data Lake Gen 2

- Builds on security available in ADL Gen 1
- RBAC has been extended into container resources
- Shared Key and Shared Access Signature Authentication
 - Shared Keys grant super user access
 - SAS tokens include permissions
- Inheritance (limited)
 - Objects can inherit permissions from parent objects if default permissions were set on parent objects ahead of time

Azure Data Lake Gen 2

Manage Access

Managing permissions for: bpadlfs/raw

Users and groups:

\$superuser (Owner)	
\$superuser (Owning Group)	
Other	
Mask	

Permissions for: \$superuser

	Read	Write	Execute
<input checked="" type="checkbox"/> Access	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/> Default *	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

* This will automatically add these permissions to all new children of this directory. Learn more about default ACLs.

Add user or group:

“DBA” Topics

What about backups?

- Not a great story for backups at the moment
- Use AzCopy to copy to a blob storage account
- Can manage redundancy within WASB/ABFS

What about indexes?

- Azure Data Lake itself provides no indexes
- U-SQL does have an CREATE INDEX statement
 - Only clustered indexes
 - Essentially re-orders the dataset
- Some file formats support indexing (Columnar)
- Generally that's what your DW platform is for

Development Environment

- Varies based on the tool you are using
- Azure Portal
- Azure Storage Explorer
- Visual Studio

How To Get Started With Azure Data Lake

- Offload archived/unused data from DW/OLTP systems into data lake
- Use data lake as a staging area for DW loads
- Have net-new projects ingest data through a data lake

What's It Cost?

Pricing – ADL Gen 1



- You Pay For:
 - Storage (\$0.039 per GB/month)
 - Operations
 - Reads (\$0.05 per 10,000)
 - Writes (\$0.004 per 10,000)
 - Outbound data transfers (varies by region)
 - North Central US: \$0.087 per GB
- *Additional storage savings with monthly commitments*
- *Check Azure Portal for most current pricing info*
- <https://azure.microsoft.com/en-us/pricing/details/data-lake-storage-gen1/>

Pricing – ADL Gen 2



- Prices very based on flat/hierarchical structure & storage tier
 - Storage (\$0.0208/0.0152/0.002 per GB/month)
 - Operations
 - Reads (\$0.0052/0.013/6.50 per 10,000)
 - Writes (\$0.065/0.13/0.13 per 10,000)
 - Outbound data transfers (varies by region)
 - North Central US: \$0.087 per GB
- *Check Azure Portal for most current pricing info*
- <https://azure.microsoft.com/en-us/pricing/details/storage/data-lake/>

Wanna Play For Free?

- Run U-SQL locally on your machine for free
- <https://docs.microsoft.com/en-us/azure/data-lake-analytics/data-lake-analytics-data-lake-tools-local-run>



Demo

Resources

- Chicago Parking Tickets Data:
<https://www.bobpusateri.com/archive/2018/09/new-data-set-chicago-parking-tickets/>
- Melissa Coates has lots of good blog posts: <https://www.sqlchick.com/entries/tag/Azure+Data+Lake>
- Run U-SQL Locally:
<https://docs.microsoft.com/en-us/azure/data-lake-analytics/data-lake-analytics-data-lake-tools-local-run>
- U-SQL Tutorial: <https://saveenr.gitbooks.io/usql-tutorial/content/>
- Querying Data in Azure Data Lake with Power BI:
<https://www.sqlchick.com/entries/2018/5/6/querying-data-in-azure-data-lake-store-with-power-bi>
- 10 Things to know about Azure Data Lake Storage Gen2:
<https://www.blue-granite.com/blog/10-things-to-know-about-azure-data-lake-storage-gen2>

Questions?



@sqlbob



bob@bobpusateri.com



bobpusateri.com



bobpusateri

THANK YOU FOR ATTENDING!