Microsoft

Reference Lambda Architecture for Big Data Platform in Azure

Big Data & Advanced Analytics Platform (BDAAP)

Prepared by

Data SQL Ninja Engineering Team (datasqlninja@microsoft.com)

Disclaimer

The High-Level Architecture, Migration Dispositions and guidelines in this document is developed in consultation and collaboration with Microsoft Corporation technical architects. Because Microsoft must respond to changing market conditions, this document should not be interpreted as an invitation to contract or a commitment on the part of Microsoft.

Microsoft has provided generic high-level guidance in this document with the understanding that MICROSOFT MAKES NO WARRANTIES, EXPRESS OR IMPLIED, WITH RESPECT TO THE INFORMATION CONTAINED HEREIN.

This document is provided "as-is". Information and views expressed in this document, including URL and other Internet Web site references, may change without notice.

Some examples depicted herein are provided for illustration only and are fictitious. No real association or connection is intended or should be inferred.

This document does not provide you with any legal rights to any intellectual property in any Microsoft product. You may copy and use this document for your internal, reference purposes.

Note: The detail provided in this document has been harvested as part of a customer engagement sponsored through the Data SQL Ninja Engineering.

Table of Contents

1 Introduction

This paper outlines approaches to build Cloud Data & Analytics Platform (CDAP) on Azure for organizations looking to leverage benefits of big data on Microsoft Cloud Platform offerings such as Azure HDInsight, Azure Cosmos DB, Azure Databricks, Azure SQL Data Warehouse, Azure SQL MI & Azure SQL DB. This paper demonstrates the recommended approach for effective usage of Azure Platform offerings to leverage best practices for Business Analyst, Data Scientist and Developers to solve real world problems with their enterprise data.

This paper is going to cover the following different areas:

a) Lambda Architecture Overview
b) CDAP Architecture depicting high level representation of setup for both batch and stream processing
c) Query Layer using Azure Databricks and delta tables
d) Query Layer using Azure SQL DW or SQL DB/MI
e) Operational Reporting with SQL DB/MI/Cosmos DB

2 Customer Scenario

This whitepaper proposes reference architecture for cloud data and analytics platform based on the real customer implementation scenario which enabled them to leverage Azure Data Services for solving their business problems. This solution helped them by enabling and empowering different functions in their business to interact with data in various ways for their decision making process, be it reporting their key KPIs for management using virtualization tools like Power BI\Tableau or *enabling* their data scientists to focus more on solving business problem to predict\forecast business critical information for future using their vast historical data. We are going to talk in detail on how this big data and advanced analytics platform can be prepared and leveraged in coming sections. You can use this reference to prepare your own Big Data and Advanced Analytics Platform on Microsoft Azure.

3 What is a Lambda Architecture?

Lambda architectures enable efficient data processing of massive data sets. Lambda architectures use batch-processing, stream-processing, and a serving layer to minimize the latency involved in querying big data.

To implement a lambda architecture on Azure, you can combine the different technologies to accelerate real-time big data analytics.

To read and understand more on Lambda Architecture, please refer to this article.

4 BDAAP Architecture

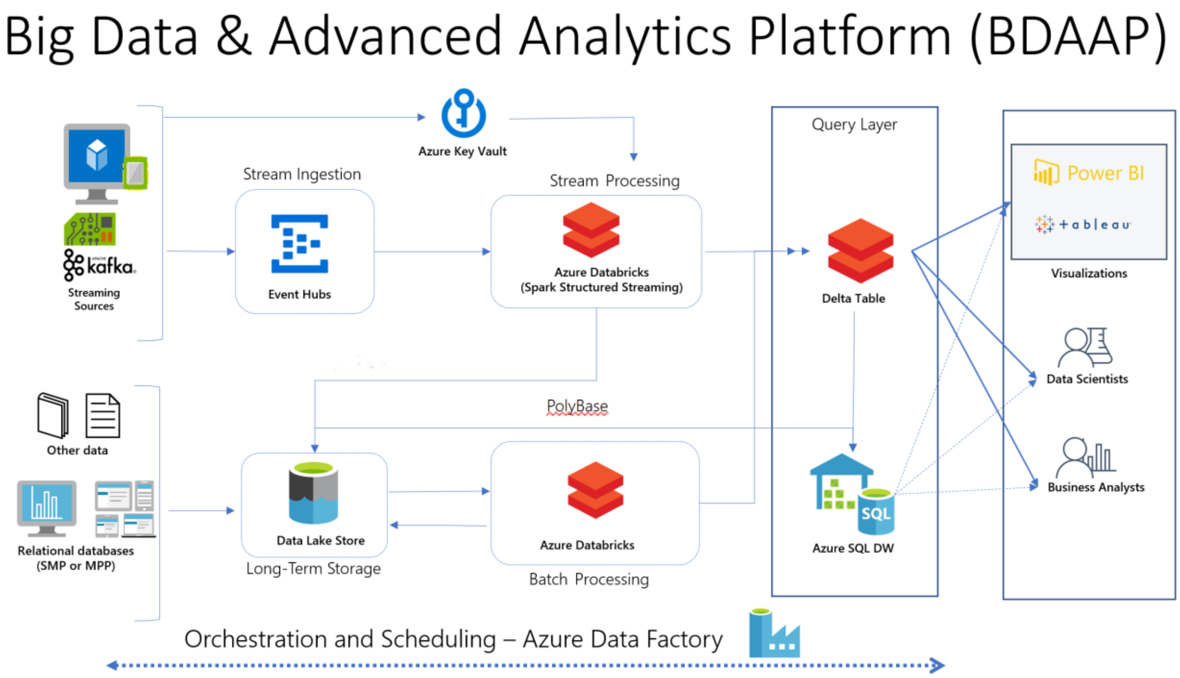


Figure 1

Big Data Architecture is premised on a skill set for developing reliable, scalable, completely automated data pipelines for the huge data volume of different types (structured/semi-structured/unstructured) which is practically not possible to process with traditional technologies. Your data resides on multiple source systems and in different formats. This data could be coming from different RDBMS sources such as Oracle/SQL Server/DB2 or flat files and could be from real time streaming from various sources through Event Hub, Kafka, Amazon Kinesis. The frequency of data could be anything - hourly, daily or even weekly. The challenge for any organization is how to ingest, transform and organize this data to derive meaningful information to drive their business. The proposed architecture suggests two different ways of organizing or curating the data.

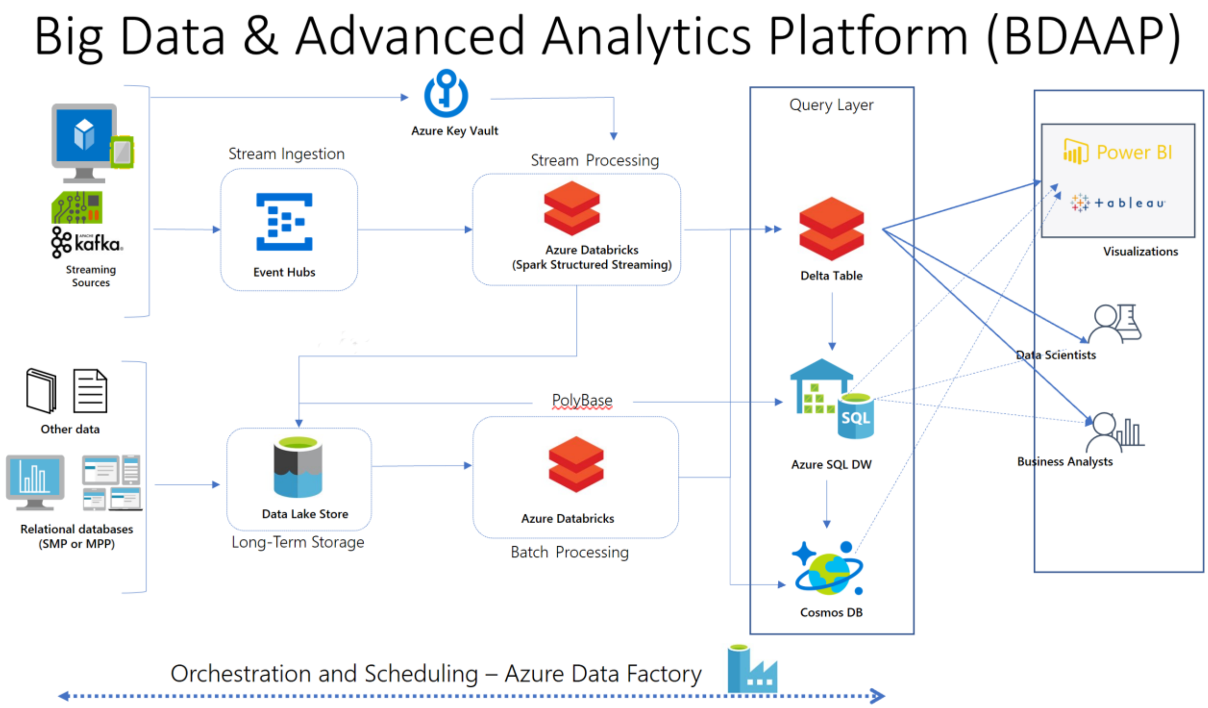


Figure 2

The difference you see in this architecture from the earlier architecture is inclusion of [Azure Cosmos DB], which is covered under non-relational section of this document. This would be desirable if you want to meet the need of operational reporting.

5 Big Data Setup / Cluster Planning

While designing Big Data solution, it is important to have the answers of following basic questions, to better understand the current workload, use case, requirements and what would be meet the expectations:

Sr No	Area	Question
1	Architecture	Which kind of technology has been used in systems?
2	Architecture	What is the average complexity of systems? For example, in terms of simple, medium, complex or very complex.
3	Architecture	What kind of data model has been applied?
4	Architecture	What kind of data management technology has been used?
5	Architecture	What analytical or semantic platform has been in use currently?
6	Architecture	What streaming technologies has been used on platform, if any?
7	Architecture	What is current solution set up for on premises big data?
8	Architecture	What framework has been used for storage?
9	Architecture	How data has been ingested into databases?

10	Architecture	How data has been processed in infrastructure?
11	Architecture	How data has ingestion has done?
12	Architecture	Which stream processing platform has been used to screen out the data not required for processing?
13	Architecture	How data has been handled prior to ingestion?
14	Architecture	What tool has been used to obtaining data? For example, data ingestion, transformation, scheduling and orchestration of data flows.
15	Architecture	what kind of architecture used for processing data?
16	Architecture	what kind of platform has been used to integrating data?
17	Architecture	What protocols in used for messaging?
18	Architecture	What tool has been used for cleaning data, if any?
19	Architecture	How data has been transformed and normalized while doing integration?
20	Architecture	What algorithms or libraries have been used as part of ML data model?
21	Architecture	In which language Algorithm has been written?
22	Architecture	What kind of business value has been considered while deciding algorithm to use?
23	Architecture	What are the challenges /blockers faced during first execution of ML architecture?
24	Architecture	If more numbers of complex features has been processed, what execution platform has been used to run ML algorithm?
25	Architecture	How deployment has happened currently?
26	Architecture	How much time taking to deploy the algorithm to Prod environment?
27	Architecture	What is the timeline and duration for which the cluster is required?
28	Architecture	What type of workload do you have? Analytical or Operational Reporting?
29	Architecture	What kind of data do you have? Relational or non-relational?
30	Architecture	Do you also have data stream as one of the core inputs?
31	Data	What is average size of data, source wise and dataset wise?
32	Data	What is target volume of data stored, with year over year growth?
33	Data	What are the different sources from where data is collected?
34	Infrastructure	What are the current network configurations?
35	Infrastructure	What all hardware requirements has been fulfilled in current set up?
36	Infrastructure	What software has been used for analysis?
37	Infrastructure	How the current infrastructure has been set up?
38	Infrastructure	What kind of storage has been used?
39	Infrastructure	Where is the infrastructure has been hosted for execution?
40	Infrastructure	What kind of different environments has currently in used?
41	Infrastructure	What geographical location should the cluster be created? This could be driven by the legal restrictions and/or availability of the cloud service
42	Infrastructure	What cluster type should you chose for your workload?

43	Infrastructure	What is the desired storage limit?
44	Infrastructure	What type of nodes required?
45	Infrastructure	What is the availability of the work force (data scientists, data engineers) for any of the supported languages (Hive, Python, Java etc.)?
46	Infrastructure	Any cost restriction?
47	Security	What is security set up established for current network?
48	Security	How data stored and managed considering security?
49	Security	What security measures taken for infrastructure security?
50	Azure Infrastructure	Preferred Region
51	Azure Infrastructure	VNet preferred?
52	Azure Infrastructure	HA / DR Needed?
53	Azure Infrastructure	Integration with other cloud services?
54	Data Movement	Initial load preference
55	Data Movement	Data transfer delta
56	Data Movement	Ongoing incremental data transfer
57	Monitoring & Alerting	Use Azure Monitoring & Alerting Vs Integrate third-party monitoring
58	Security Preferences	Private and protected data pipeline?
59	Security Preferences	Domain Joined cluster (ESP)?
60	Security Preferences	On-Premises AD Sync to Cloud?
61	Security Preferences	No. of AD users to sync?
62	Security Preferences	Ok to sync passwords to cloud?
63	Security Preferences	Cloud only Users?
64	Security Preferences	MFA needed?
65	Security Preferences	Data authorization requirements?
66	Security Preferences	Role-Based Access Control?
67	Security Preferences	Auditing needed?
68	Security Preferences	Data encryption at rest?
69	Security Preferences	Data encryption in transit?
70	Architecture Preferences	Single cluster vs Specific cluster types
71	Architecture Preferences	Colocated Storage Vs Remote Storage?
72	Architecture Preferences	Smaller cluster size as data is stored remotely?
73	Architecture Preferences	Use multiple smaller clusters rather than a single large cluster?
74	Architecture Preferences	Use a remote metastore?
75	Architecture Preferences	Share metastores between different clusters?
76	Architecture Preferences	Deconstruct workloads?
77	Architecture Preferences	Use ADF for data orchestration?

Based on the answers of above questionnaire and the need a setup can be done for the cluster (either using Azure Databricks or Azure HDInsight) that can support the architecture. E.g. A cluster that can take streaming, run the concurrent jobs (written using SQL or Python etc.) using API or UI.

6 Ingress / Data Loading

Once your cluster is up and running, you must decide how to load the data. In practice there are two widely used approaches and they are batch and event driven, streaming data. Batch load are usually for structure data and file, event driven approach is mostly used for near-real-time events such as social network data, logs or transactional data.

6.1 Batch Ingest

The data from structured data sources namely RDBMS, flat files (on-premise or cloud) can be ingested to Azure Data Lake Store by using Azure Data Factory in the batches. Using Azure Data Factory this process could be orchestrated and scheduled depending on the frequency of data availability in source systems. Alternatively, one can use any other open source (like Apache Airflow), non-proprietary technology if there is need.

6.2 Event Ingest

Not all data arrives to your solution in an orderly fashion at a reasonable speed. Some data sources will send data in real-time with hundreds or thousands of messages per second that will need to be queued, processed and then eventually stored. To handle this faster variant of data sources, your solution may need a component that can ingest messages in real-time.

The data from your streaming systems aka streaming data which includes wide variety of data such as log files generated by customers using mobile web applications, ecommerce purchases, information from social networks, financial trading floors, or geospatial services, and telemetry from connected devices or instrumentation in data centers. This data needs to be processed sequentially and incrementally on a record-by-record basis or over sliding time windows and used for a wide variety of analytics including correlations, aggregations, filtering, and sampling. Azure Event Hubs can be used as message queue to ingest the data which can then be processed using Azure Databricks Spark Structured Streaming in to Azure Data Lake Store, enhanced data protection and compliance can be achieved by using encryption\decryption of data and maintaining key in Azure Key Vault to protect your online data. We will talk more about security architecture in greater details in our next whitepaper.

Note – Sometimes, you might have event sources for which you might not have native support to consume data from in Structured Streaming or Spark Streaming, for example HTTP endpoint – in this case you can either:

- Write a custom application to read from HTTP endpoint and write it to Event Hub or Kafka from where Structured Streaming or Spark Streaming can consume it for further processing
- Create Custom Data Source (Structured Streaming) or Custom Receiver (Spark Streaming) to consume data directly HTTP endpoint.

Both of these approaches have its own pros and cons, for example with first one you need to ensure you are handling failure scenario in your custom application, likewise with second option you have to write your very own Custom Data Source (Structured Streaming) or Custom Receiver (Spark Streaming).

6.3 Staging

Azure Data Lake Store is used to stage data in the cloud. A data lake is a storage repository that holds a large amount of data in its native, raw format. Data lake stores are optimized for scaling to terabytes and petabytes of data. The data typically comes from multiple heterogeneous sources, and may be structured, semi-structured, or unstructured. The idea with a data lake is to store everything in its original, untransformed state. This approach differs from a traditional data warehouse, which transforms and processes the data at the time of ingestion.

ADLS can be classified in three different zones data storage:

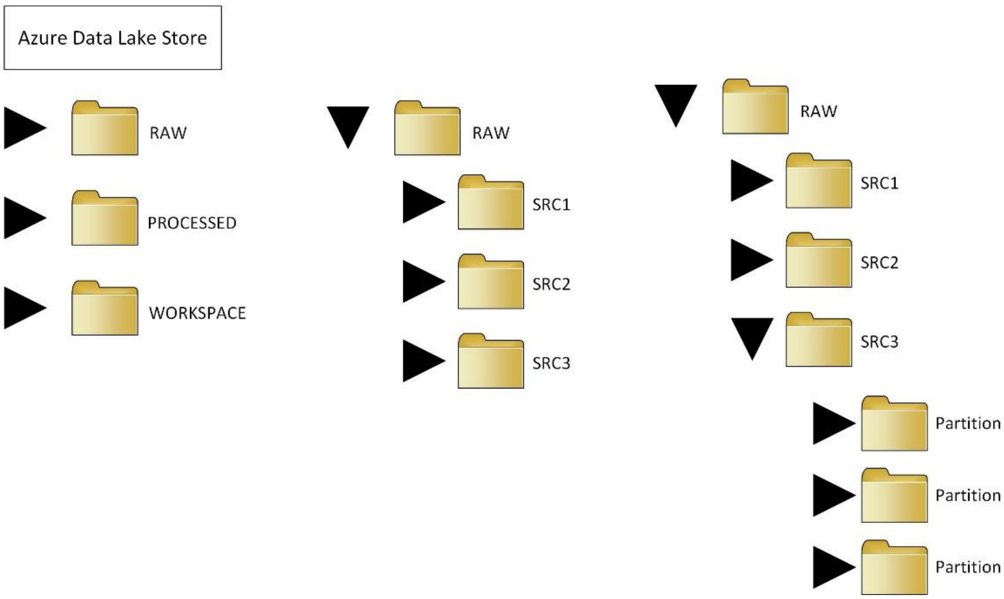
1) Raw or Landing Zone: This is where the data is ingested directly from source in its original formats and the underlying data will be organized by source/domain. It can be by domain for example Sales. Furthermore, this can be segregated into datasets by functions/entity and then partitioned. Only data ingestion jobs should have read-write access to this zone where as data processing jobs and data scientists (needed access to raw data) should have read-only access.

2) Processed: This section is usually organized keeping destination data model in mind. You use this zone for storing processed data intended for analytical queries and normally would be Parquet format (you can choose other format as well based on your need like ORC, AVRO etc.). Only data processing jobs should have read-write access to this zone where as individuals from your organizations and teams should have read-only access.

3) Workspaces: This is the zone where individuals from your organizations and teams will have read-write access to their respective folders. This section is organized keeping access

masking done for data based on individuals or team. In many cases, it has been observed that the users mostly data scientist has access to Raw data directly so as they can curate the data for different kind of analysis. We will talk more about security architecture in greater details in our next whitepaper.

By Source



By Domain

Raw

Domain

Dataset

Partitions

Processed

Domain

Dataset

Partitions

Workspaces

Individuals

Employee A

Employee B

Teams

Team A

Team B

7 Data Processing

Once the data has been loaded into the staging area or landing zone (Azure Data Lake Store), the next step is to automatically process this data using either or combination for these compute engines – Azure Databricks, Azure HDInsight and Azure SQL Data Warehouse. Based on choice of your compute engines, your language of choices to write data processing language would vary.

The data processing to target systems can be orchestrated and scheduled using Azure Data Factory. The processed data in the target systems can be consumed by end users for varied purpose, data analysis, data engineering, training machine learning models, forecasting.

7.1 Non-relational Engines

7.1.1 Azure Databricks

You can use Azure Databricks to process and prepare your big data using underlying parallel and distributed Spark engine.

Databricks builds on top of Spark and adds:

- Highly reliable and performant data pipelines
- Productive data science at scale
- Import data using APIs or connectors
- Clean malformed data
- Aggregate data to create a data warehouse

Also, based on the answers of the questionnaire, it would be clear that when there is a need of a team of engineers experienced in Python or Scala to work closely on data science project, Azure Databricks is the right choice. Because of the available features of Azure Databricks, it's very easy to manage clusters by non-IT personnel. They you can start with a basic cluster which can later be scaled up based on work load. Once the required load is completed and the queries are executed, the cluster can be scaled down. The Azure Databricks can also work with the data streaming. You can learn more about it here: Azure Databricks.

7.1.2 HDInsight (Hive or Spark – using Interactive Query)

Based on the answers of the questionnaire, it would be clear that when you have people experienced with SQL programming and you want to make use of open source technology without the need of frequent pause on cluster, HDInsight with HQL is the right choice. The

Interactive Query feature make it more convenient for the developers to work with multi structured data.

In case the developers are not familiar with SQL or HQL then still HDInsight can be a choice. HDInsight Spark cluster supports other languages like Python, Scala or R. This spark cluster could also help you in case you have streaming data sources. However, the frequent scale-up and down is not best supported. To achieve such functionality, the whole cluster need to be spin up and killed once work is over. You can learn more about it here: Azure HDInsight.

7.1.3 Azure Cosmos DB

Azure Cosmos DB is a global distributed, multi-model database that is used in a wide range of applications and use cases. It is a good choice for any serverless application that needs low order-of-millisecond response times, and needs to scale rapidly and globally. It supports multiple data models (key-value, documents, graphs and columnar) and many APIs for data access including Azure Cosmos DB's API for MongoDB, SQL API, Gremlin API, and Tables API natively, and in an extensible manner. Power BI can connect to Azure Cosmos DB data by using the Power BI connector for visualization/reporting needs. You can learn more about it here: Azure Cosmos DB.

7.2 Relational Engines

7.2.1 Azure SQL Data Warehouse

It’s clear from the questionnaire that when you have the Relational/Structured data, SQL DW would be the right choice. SQL Data Warehouse is a cloud-based Enterprise Data Warehouse (EDW) that leverages Massively Parallel Processing (MPP) to quickly run complex queries across petabytes of data. Use SQL Data Warehouse as a key component of a big data solution. Import big data into SQL Data Warehouse with simple PolyBase T-SQL queries, and then use the power of MPP to run high-performance analytics. As you integrate and analyze, the data warehouse will become the single version of truth your business can count on for insights. You can learn more about it here: Azure SQL Data Warehouse.

8 Visualization

Visualization of the data is the ability to present a massive amount of data in a pictorial or graphical format to enable decision-makers interpret or identity new patterns easily. Big data visualization need displaying of huge volume of data that can be stored in cloud or dealing with continuously flowing real-time data. This can be achieved using Power BI, Tableau or any other visualization tool of your choice. Business Groups having access to their workspaces can query Azure SQL Data Warehouse using standard SQL complaint queries to develop reports, dashboards. On the other hand, Data scientists, data engineers and business analysts can collaborate on shared projects in an interactive workspace of Azure Databricks. Apply their existing skills with support for Python, Scala, R and SQL, as well as deep learning frameworks and libraries like TensorFlow, Pytorch and Scikit-learn.

Power BI is our preferred and recommended tool for your big data visualization. With Power BI, you can

- Self-service preparation for big data
- Prep your data using the familiar Power Query experience
- Make use of rich visualization available to present your data
- Get started quickly with a common data model
- Take advantage of a standardized and extensible collection of data schemas
- Enrich your models with other data sources from Microsoft and third parties
- Unify access to data between Power BI and Azure Data Lake Storage
- Leverage Advanced analytics and AI with Azure by using ML models.
- Comply with industry standard data security

You can learn more about it here: Power BI

9 Query Layer

9.1 Query Layer – Azure Databricks via Delta Tables

The recommended approach for this layer is to have a separate Azure Databricks workspace for each individual business groups (all the data scientists from that group). Each business group users will have separate workspace defined for them under Azure Databricks layer which will have access to Azure Data Lake Store and Shared Metastore. The BG users in the workspace will have access to all the databases, tables within the workspace. This ensure scalability by reducing the load as each group will have access to only what they need and helps track consumption for each individual business group within the subscription. The implementation can be done by creating ARM template and using the same to replicate as and how the scenario or business case is.

Query Layer – Azure Databricks via Delta Tables

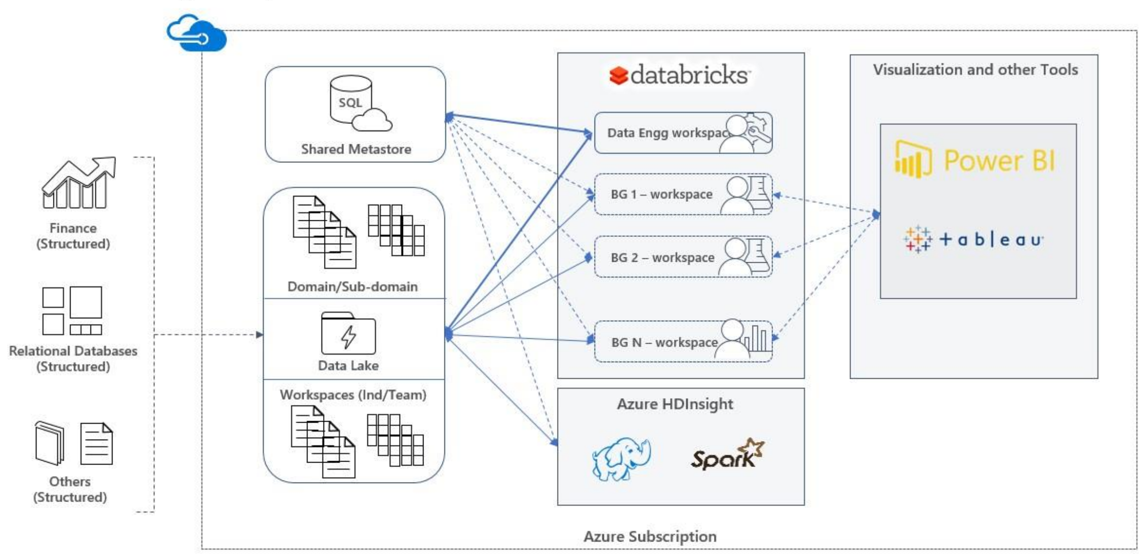


Figure 3

9.2 Query Layer – Azure SQL DW

The recommended approach for this layer is to have a SQL Compliant Query Layer created using [Azure SQL DW](#) which can be separate for each business user group. Azure SQL DW has features that are designed for working with Big Data and serving it for further analysis and visualization.

You can have one single Azure SQL Data Warehouse database, serving as one single query and processing layer for all the business users or can scaler out have multiple Azure SQL Data Warehouse databases, one for each business groups or combination of both.

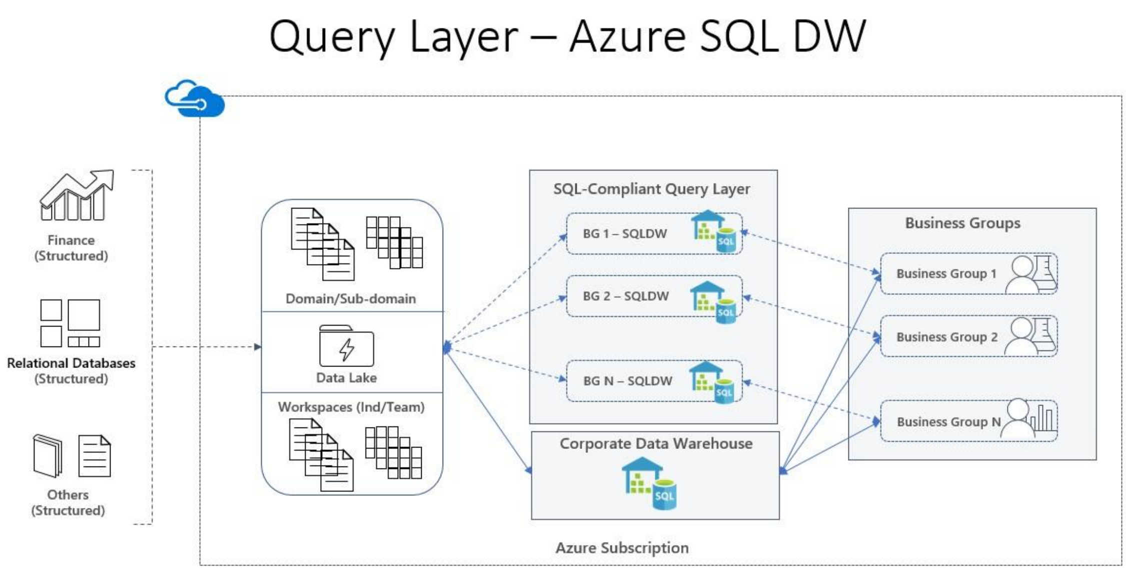


Figure 4

The Azure SQL DW will be connected to Azure Data Lake Store to pull the data, there are two different ways to access the data

1) By using PolyBase queries to access data directly from Data Lake Store
2) By creating internal table and loading data into it

Depending on your scenario, you might have Azure Databricks to prepare and train the data then you can use Azure SQL Data Warehouse database to store processed data for efficient querying, if not then you can land raw data into staging area of Azure SQL Data Warehouse database, applying required transformation using stored procedures and load processed data into your data warehouse models.

9.3 Query Layer – SQL DB/Managed Instance

You would prefer to use Azure SQL Data Warehouse if your data volume is huge and you want massively parallel processing engine to serve your data processing and querying needs. In case if your data volume is not huge you can either use [Azure SQL DB] or [Azure SQL Managed Instance (MI).]

The recommended approach for this layer is to have a SQL Compliant Query Layer created using [Azure SQL DB] or [Azure SQL MI] (Figure 5) which can be separate for each business user group. A corporate data warehouse can be created and subsequent SQL DB/MI instances to serve the needs of different business groups within the organization. Each business group will have their own workspaces, tables to interact with data for their needs. Access control must be implemented for individual users and business groups/teams as desired.

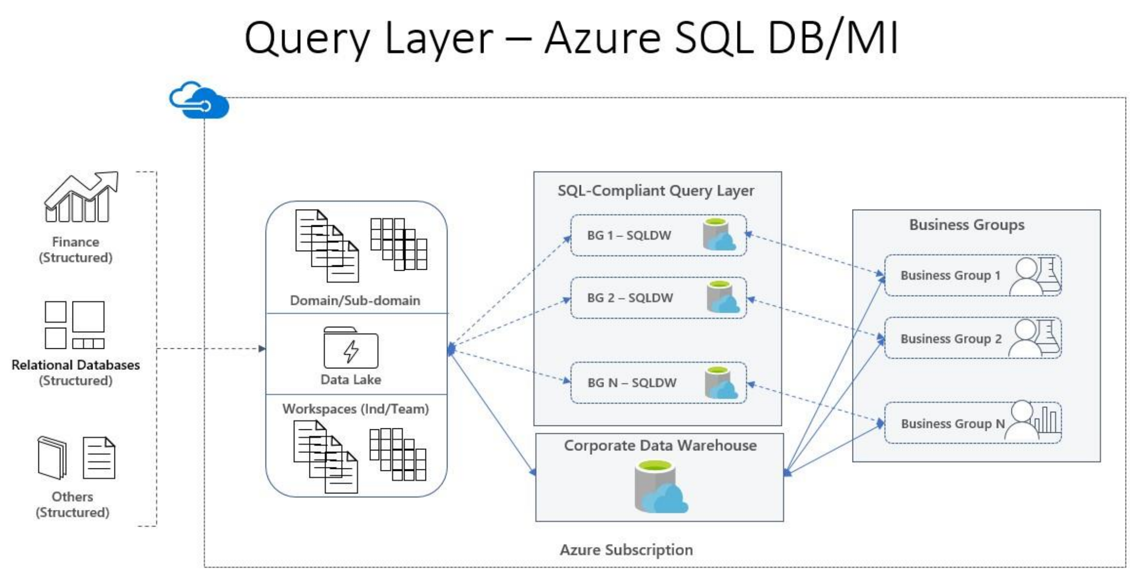


Figure 5

You can have one single Azure SQL Data Warehouse database, serving as one single query and processing layer for all the business users or can scaler out have multiple Azure SQL Data Warehouse databases, one for each business groups or combination of both.

10 Reference

Azure Data Lake Storage Gen2: https://docs.microsoft.com/en-us/azure/storage/blobs/data-lake-storage-best-practices

Azure HDInsight: https://azure.microsoft.com/en-in/blog/migrating-on-premises-hadoop-infrastructure-to-azure-hdinsight/

Azure SQL Data Warehouse: https://docs.microsoft.com/en-us/azure/sql-data-warehouse/sql-data-warehouse-best-practices

Azure Databases: https://azure.microsoft.com/en-us/product-categories/databases/

Lambda Architecture: https://docs.microsoft.com/en-us/azure/cosmos-db/lambda-architecture

Azure Databricks:

https://docs.azuredatabricks.net/getting-started/index.html

https://docs.azuredatabricks.net/user-guide/clusters/create.html

https://databricks.com/product/azure

https://docs.microsoft.com/en-us/azure/architecture/reference-architectures/data/stream-processing-databricks

https://docs.azuredatabricks.net/user-guide/bi/index.html

https://docs.azuredatabricks.net/delta/delta-intro.html

https://docs.microsoft.com/en-us/azure/azure-databricks/

Azure Event Hubs:

https://docs.microsoft.com/en-us/azure/event-hubs/

https://docs.microsoft.com/en-us/azure/event-hubs/event-hubs-for-kafka-ecosystem-overview

Suggested articles:

https://azure.microsoft.com/en-gb/solutions/architecture/modern-data-warehouse/

https://azure.microsoft.com/en-gb/solutions/architecture/advanced-analytics-on-big-data/

https://blogs.msdn.microsoft.com/azurecat/2018/12/24/azure-data-architecture-guide-blog-8-data-warehousing/

https://azure.microsoft.com/en-gb/blog/azure-stream-analytics-now-supports-azure-sql-database-as-reference-data-input/

https://azure.microsoft.com/en-in/resources/videos/azure-friday-hdinsight-fast-interactive-queries-with-hive-on-llap/

https://docs.microsoft.com/en-us/azure/hdinsight/spark/apache-spark-overview

https://azure.microsoft.com/en-in/blog/the-emerging-big-data-architectural-pattern/

https://www.blue-granite.com/data-warehouse-azure-cloud-webinar-jan-2018

11 Feedback and suggestions

If you have feedback or suggestions for improving this data migration asset, please contact the Data SQL Ninja Team (datasqlninja@microsoft.com). Thanks for your support!

Note: For additional information about migrating various source databases to Azure, see the Azure Database Migration Guide.