Azure Data Architectures

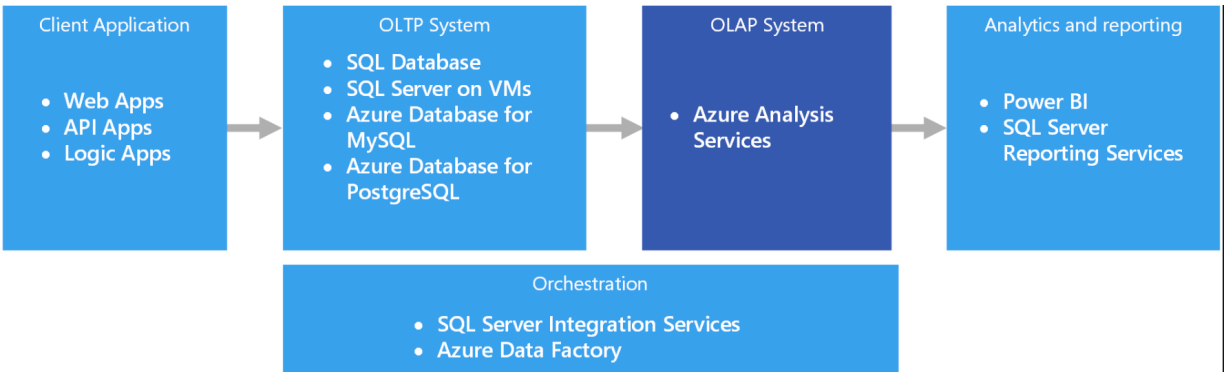
OLAP in Azure

https://docs.microsoft.com/en-us/azure/architecture/data-guide/relational-data/online-analytical-processing

In Azure, data held in OLTP systems such as Azure SQL Database is copied into the OLAP system, such as Azure Analysis Services. Data exploration and visualization tools like Power BI, Excel, and third-party options connect to Analysis Services servers and provide users with highly interactive and visually rich insights into the modeled data. The flow of data from OLTP data to OLAP is typically orchestrated using SQL Server Integration Services, which can be executed using Azure Data Factory.
In Azure, all of the following data stores will meet the core requirements for OLAP:

- SQL Server with Columnstore indexes
- Azure Analysis Services
- SQL Server Analysis Services (SSAS)

SQL Server Analysis Services (SSAS) offers OLAP and data mining functionality for business intelligence applications. You can either install SSAS on local servers, or host within a virtual machine in Azure. Azure Analysis Services is a fully managed service that provides the same major features as SSAS. Azure Analysis Services supports connecting to various data sources in the cloud and on-premises in your organization.
Clustered Columnstore indexes are available in SQL Server 2014 and above, as well as Azure SQL Database, and are ideal for OLAP workloads. However, beginning with SQL Server 2016 (including Azure SQL Database), you can take advantage of hybrid transactional/analytics processing (HTAP) through the use of updateable nonclustered columnstore indexes. HTAP enables you to perform OLTP and OLAP processing on the same platform, which removes the need to store multiple copies of your data, and eliminates the need for distinct OLTP and OLAP systems.



Capability matrix

The following tables summarize the key differences in capabilities.

General capabilities

Capability	Azure Analysis Services	SQL Server Analysis Services	SQL Server with Columnstore Indexes	Azure SQL Database with Columnstore Indexes
Is managed service	Yes	No	No	Yes
Supports multidimensional cubes	No	Yes	No	No
Supports tabular semantic models	Yes	Yes	No	No
Easily integrate multiple data sources	Yes	Yes	No 1	No 1
Supports real-time analytics	No	No	Yes	Yes
Requires process to copy data from source(s)	Yes	Yes	No	No

Azure AD integration	Yes	No	No 2	Yes

Scalability Capabilities

Capability	Azure Analysis Services	SQL Server Analysis Services	SQL Server with Columnstore Indexes	Azure SQL Database with Columnstore Indexes
Redundant regional servers for high availability	Yes	No	Yes	Yes
Supports query scale out	Yes	No	Yes	Yes
Dynamic scalability (scale up)	Yes	No	Yes	Yes

OLTP in Azure

https://docs.microsoft.com/en-us/azure/architecture/data-guide/relational-data/online-transaction-processing

Applications such as websites hosted in App Service Web Apps, REST APIs running in App Service, or mobile or desktop applications communicate with the OLTP system, typically via a REST API intermediary.
In practice, most workloads are not purely OLTP. There tends to be an analytical component as well. In addition, there is an increasing demand for real-time reporting, such as running reports against the operational system. This is also referred to as HTAP (Hybrid Transactional and Analytical Processing). For more information, see Online Analytical Processing (OLAP).

Capability matrix

The following tables summarize the key differences in capabilities.

General capabilities

Capability	Azure SQL Database	SQL Server in an Azure virtual machine	Azure Database for MySQL	Azure Database for PostgreSQL
Is Managed Service	Yes	No	Yes	Yes
Runs on Platform	N/A	Windows, Linux, Docker	N/A	N/A
Programmability 1	T-SQL, .NET, R	T-SQL, .NET, R, Python	SQL	SQL, PL/pgSQL

Scalability capabilities

Capability	Azure SQL Database	SQL Server in an Azure virtual machine	Azure Database for MySQL	Azure Database for PostgreSQL
Maximum database instance size	4 TB	256 TB	16 TB	16 TB
Supports capacity pools	Yes	Yes	No	No
Supports clusters scale out	No	Yes	No	No
Dynamic scalability (scale up)	Yes	No	Yes	Yes

Analytic workload capabilities

Capability	Azure SQL Database	SQL Server in an Azure virtual machine	Azure Database for MySQL	Azure Database for PostgreSQL
Temporal tables	Yes	Yes	No	No
In-memory (memory-optimized) tables	Yes	Yes	No	No
Columnstore support	Yes	Yes	No	No

Adaptive query processing	Yes	Yes	No	No

Availability capabilities

Capability	Azure SQL Database	SQL Server in an Azure virtual machine	Azure Database for MySQL	Azure Database for PostgreSQL
Readable secondaries	Yes	Yes	Yes	Yes
Geographic replication	Yes	Yes	Yes	Yes
Automatic failover to secondary	Yes	No	No	No
Point-in-time restore	Yes	Yes	Yes	Yes

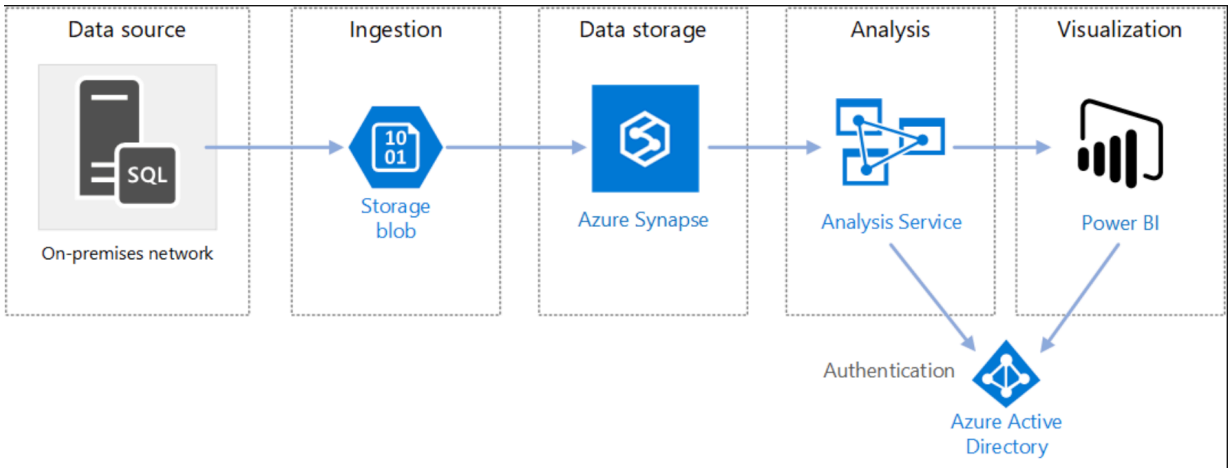
Security capabilities

Capability	Azure SQL Database	SQL Server in an Azure virtual machine	Azure Database for MySQL	Azure Database for PostgreSQL
Row level security	Yes	Yes	Yes	Yes
Data masking	Yes	Yes	No	No
Transparent data encryption	Yes	Yes	Yes	Yes
Restrict access to specific IP addresses	Yes	Yes	Yes	Yes
Restrict access to allow VNet access only	Yes	Yes	Yes	Yes
Azure Active Directory authentication	Yes	Yes	No	No

Active Directory authentication	No	Yes	No	No
Multi-factor authentication	Yes	Yes	No	No
Supports Always Encrypted	Yes	Yes	No	No
Private IP	No	Yes	No	No

Data warehouse architectures

A data warehouse is a centralized repository of integrated data from one or more disparate sources. Data warehouses store current and historical data and are used for reporting and analysis of the data.



To move data into a data warehouse, data is periodically extracted from various sources that contain important business information. As the data is moved, it can be formatted, cleaned, validated, summarized, and reorganized. Alternatively, the data can be stored in the lowest level of detail, with aggregated views provided in the warehouse for reporting. In either case, the data warehouse becomes a permanent data store for reporting, analysis, and business intelligence (BI).

The following reference architectures show end-to-end data warehouse architectures on Azure:

- Enterprise BI in Azure with Azure Synapse Analytics. This reference architecture implements an extract, load, and transform (ELT) pipeline that moves data from an on-premises SQL Server database into Azure Synapse.
- Automated enterprise BI with Azure Synapse and Azure Data Factory. This reference architecture shows an ELT pipeline with incremental loading, automated using Azure Data Factory.

Data warehousing in Azure

You may have one or more sources of data, whether from customer transactions or business applications. This data is traditionally stored in one or more OLTP databases. The data could be persisted in other storage mediums such as network shares, Azure Storage Blobs, or a data lake. The data could also be stored by the data warehouse itself or in a relational database such as Azure SQL Database. The purpose of the analytical data store layer is to satisfy queries issued by analytics and reporting tools against the data warehouse. In Azure, this analytical store capability can be met with Azure Synapse, or with Azure HDInsight using Hive or Interactive Query. In addition, you will need some level of orchestration to move or copy data from data storage to the data warehouse, which can be done using Azure Data Factory or Oozie on Azure HDInsight.
There are several options for implementing a data warehouse in Azure, depending on your needs. The following lists are broken into two categories, symmetric multiprocessing (SMP) and massively parallel processing (MPP).

SMP:

- Azure SQL Database
- SQL Server in a virtual machine

MPP:

- Azure Data Warehouse
- Apache Hive on HDInsight
- Interactive Query (Hive LLAP) on HDInsight

As a general rule, SMP-based warehouses are best suited for small to medium data sets (up to 4-100 TB), while MPP is often used for big data. The delineation between small/medium and big data partly has to do with your organization's definition and supporting infrastructure. (See Choosing an OLTP data store.)
Beyond data sizes, the type of workload pattern is likely to be a greater determining factor. For example, complex queries may be too slow for an SMP solution, and require an MPP solution instead. MPP-based systems usually have a performance penalty with small data sizes, because of how jobs are distributed and consolidated across nodes. If your data sizes already exceed 1 TB and are expected to continually grow, consider selecting an MPP solution. However, if your data sizes are smaller, but your workloads are exceeding the available resources of your SMP solution, then MPP may be your best option as well.

The data accessed or stored by your data warehouse could come from a number of data sources, including a data lake, such as Azure Data Lake Storage. For a video session that compares the different strengths of MPP services that can use Azure Data Lake, see Azure Data Lake and Azure Data Warehouse: Applying

Modern Practices to Your App.
SMP systems are characterized by a single instance of a relational database management system sharing all resources (CPU/Memory/Disk). You can scale up an SMP system. For SQL Server running on a VM, you can scale up the VM size. For Azure SQL Database, you can scale up by selecting a different service tier. MPP systems can be scaled out by adding more compute nodes (which have their own CPU, memory, and I/O subsystems). There are physical limitations to scaling up a server, at which point scaling out is more desirable, depending on the workload. However, the differences in querying, modeling, and data partitioning mean that MPP solutions require a different skillset.

Capability Matrix

The following tables summarize the key differences in capabilities.

General capabilities

Capability	Azure SQL Database	SQL Server (VM)	Azure Synapse	Apache Hive on HDInsight	Hive LLAP on HDInsight
Is managed service	Yes	No	Yes	Yes 1	Yes 1
Requires data orchestration (holds copy of data/ historical data)	No	No	Yes	Yes	Yes
Easily integrate multiple data sources	No	No	Yes	Yes	Yes
Supports pausing compute	No	No	Yes	No 2	No 2
Relational data store	Yes	Yes	Yes	No	No
Real-time reporting	Yes	Yes	No	No	Yes

Flexible backup restore points	Yes	Yes	No 3	Yes 4	Yes 4
SMP/MPP	SMP	SMP	MPP	MPP	MPP

Scalability capabilities

Capability	Azure SQL Database	SQL Server (VM)	Azure Synapse	Apache Hive on HDInsight	Hive LLAP on HDInsight
Redundant regional servers for high availability	Yes	Yes	Yes	No	No
Supports query scale out (distributed queries)	No	No	Yes	Yes	Yes
Dynamic scalability	Yes	No	Yes 1	No	No
Supports in-memory caching of data	Yes	Yes	Yes	Yes	Yes

Security capabilities

Capability	Azure SQL Database	SQL Server in a virtual machine	Azure Synapse	Apache Hive on HDInsight	Hive LLAP on HDInsight
Authentication	SQL / Azure Active Directory (Azure AD)	SQL / Azure AD / Active Directory	SQL / Azure AD	local / Azure AD 1	local / Azure AD 1
Authorization	Yes	Yes	Yes	Yes	Yes 1
Auditing	Yes	Yes	Yes	Yes	Yes 1

Data encryption at rest	Yes 2	Yes 2	Yes 2	Yes 2	Yes 1
Row-level security	Yes	Yes	Yes	No	Yes 1
Supports firewalls	Yes	Yes	Yes	Yes	Yes 3
Dynamic data masking	Yes	Yes	Yes	No	Yes 1

Big Data Architecture

A big data architecture is designed to handle the ingestion, processing, and analysis of data that is too large or complex for traditional database systems. The threshold at which organizations enter into the big data realm differs, depending on the capabilities of the users and their tools. For some, it can mean hundreds of gigabytes of data, while for others it means hundreds of terabytes. As tools for working with big data sets advance, so does the meaning of big data. More and more, this term relates to the value you can extract from your data sets through advanced analytics, rather than strictly the size of the data, although in these cases they tend to be quite large.

Big data solutions typically involve one or more of the following types of workload:

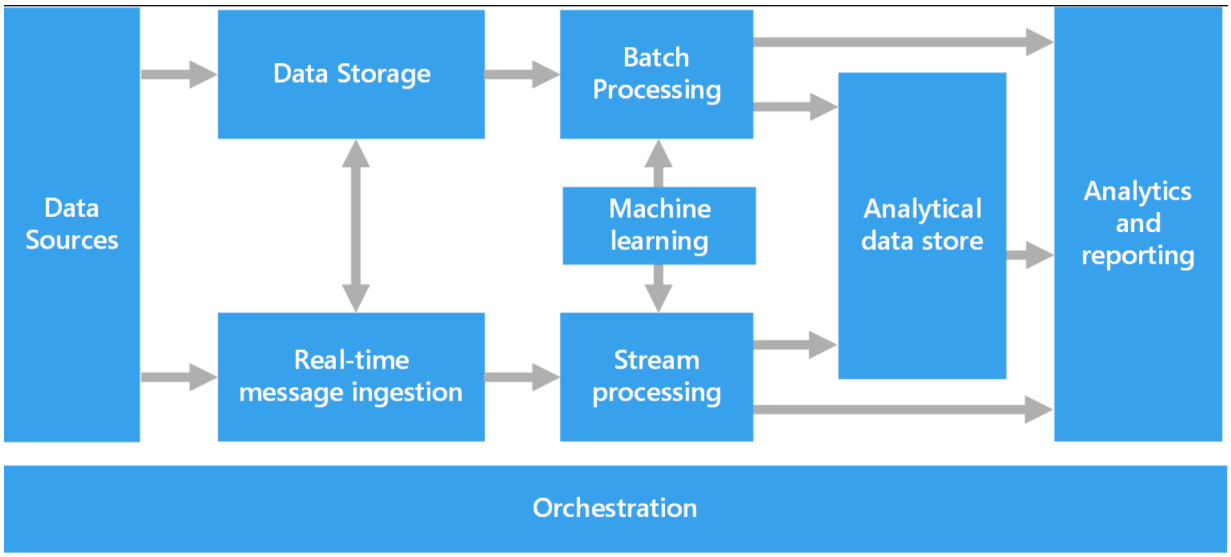
- Batch processing of big data sources at rest.
- Real-time processing of big data in motion.
- Interactive exploration of big data.
- Predictive analytics and machine learning.

Consider big data architectures when you need to:

- Store and process data in volumes too large for a traditional database.
- Transform unstructured data for analysis and reporting.
- Capture, process, and analyze unbounded streams of data in real time, or with low latency.

Components of a big data architecture

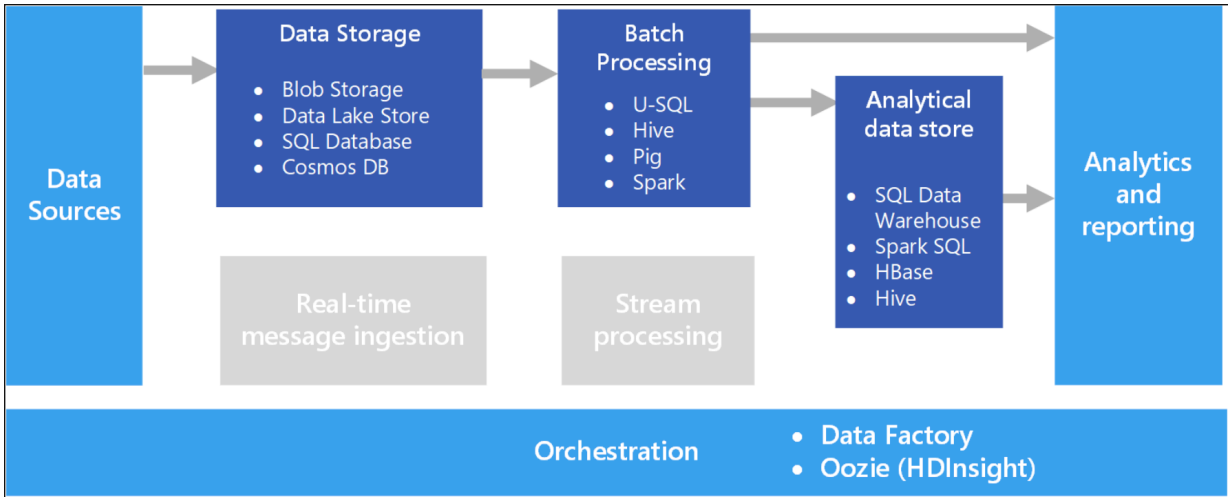
The following diagram shows the logical components that fit into a big data architecture. Individual solutions may not contain every item in this diagram.



Batch Processing

A common big data scenario is batch processing of data at rest. In this scenario, the source data is loaded into data storage, either by the source application itself or by an orchestration workflow. The data is then processed in-place by a parallelized job, which can also be initiated by the orchestration workflow. The processing may include multiple iterative steps before the transformed results are loaded into an analytical data store, which can be queried by analytics and reporting components.

For example, the logs from a web server might be copied to a folder and then processed overnight to generate daily reports of web activity.



Technology choices

The following technologies are recommended choices for batch processing solutions in Azure.

Data storage

- **Azure Storage Blob Containers**. Many existing Azure business processes already use Azure blob storage, making this a good choice for a big data store.
- **Azure Data Lake Store**. Azure Data Lake Store offers virtually unlimited storage for any size of file, and extensive security options, making it a good choice for extremely large-scale big data solutions that require a centralized store for data in heterogeneous formats.

Batch processing

- **U-SQL**. U-SQL is the query processing language used by Azure Data Lake Analytics. It combines the declarative nature of SQL with the procedural extensibility of C#, and takes advantage of parallelism to enable efficient processing of data at massive scale.
- **Hive**. Hive is a SQL-like language that is supported in most Hadoop distributions, including HDInsight. It can be used to process data from any HDFS-compatible store, including Azure blob storage and Azure Data Lake Store.
- **Pig**. Pig is a declarative big data processing language used in many Hadoop distributions, including HDInsight. It is particularly useful for processing data that is unstructured or semi-structured.
- **Spark**. The Spark engine supports batch processing programs written in a range of languages, including Java, Scala, and Python. Spark uses a distributed architecture to process data in parallel across multiple worker nodes.

Analytical data store

- **Azure Synapse Analytics**. Azure Synapse is a managed service based on SQL Server database technologies and optimized to support large-scale data warehousing workloads.
- **Spark SQL**. Spark SQL is an API built on Spark that supports the creation of dataframes and tables that can be queried using SQL syntax.
- **HBase**. HBase is a low-latency NoSQL store that offers a high-performance, flexible option for querying structured and semi-structured data.
- **Hive**. In addition to being useful for batch processing, Hive offers a database architecture that is conceptually similar to that of a typical relational database management system. Improvements in Hive query performance through innovations like the Tez engine and Stinger initiative mean that Hive tables can be used effectively as sources for analytical queries in some scenarios.

Analytics and reporting

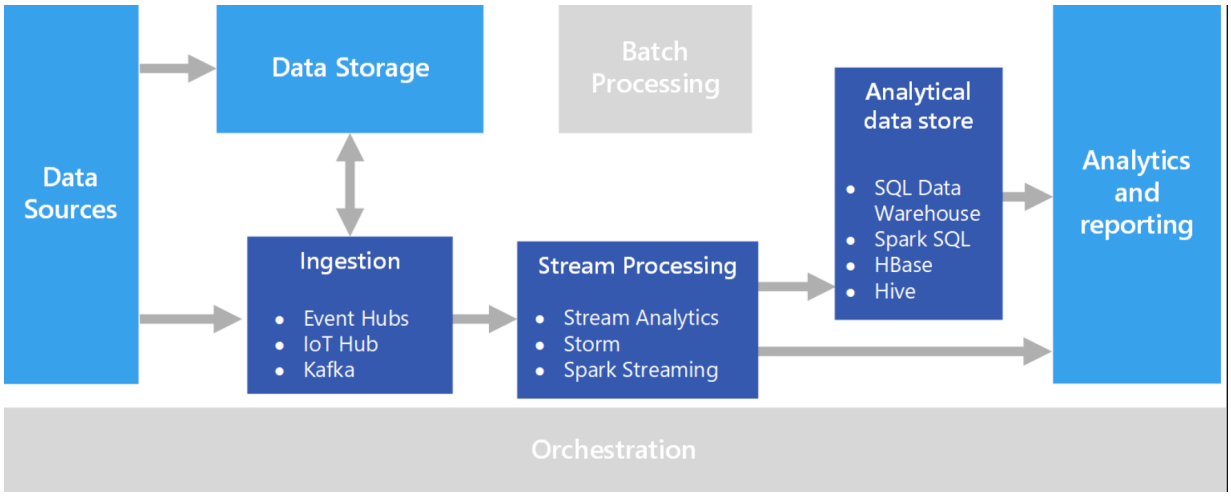
- **Azure Analysis Services**. Many big data solutions emulate traditional enterprise business intelligence architectures by including a centralized online analytical processing (OLAP) data model (often referred to as a cube) on which reports, dashboards, and interactive "slice and dice" analysis can be based. Azure Analysis Services supports the creation of tabular models to meet this need.
- **Power BI**. Power BI enables data analysts to create interactive data visualizations based on data models in an OLAP model or directly from an analytical data store.
- **Microsoft Excel**. Microsoft Excel is one of the most widely used software applications in the world, and offers a wealth of data analysis and visualization capabilities. Data analysts can use Excel to build document data models from analytical data stores, or to retrieve data from OLAP data models into interactive PivotTables and charts.

Orchestration

- **Azure Data Factory**. Azure Data Factory pipelines can be used to define a sequence of activities, scheduled for recurring temporal windows. These activities can initiate data copy operations as well as Hive, Pig, MapReduce, or Spark jobs in on-demand HDInsight clusters; U-SQL jobs in Azure Date Lake Analytics; and stored procedures in Azure Synapse or Azure SQL Database.
- **Oozie** and **Sqoop**. Oozie is a job automation engine for the Apache Hadoop ecosystem and can be used to initiate data copy operations as well as Hive, Pig, and MapReduce jobs to process data and Sqoop jobs to copy data between HDFS and SQL databases.

Real Time Processing

Real time processing deals with streams of data that are captured in real-time and processed with minimal latency to generate real-time (or near-real-time) reports or automated responses. For example, a real-time traffic monitoring solution might use sensor data to detect high traffic volumes. This data could be used to dynamically update a map to show congestion, or automatically initiate high-occupancy lanes or other traffic management systems.



Real-time processing is defined as the processing of unbounded stream of input data, with very short latency requirements for processing — measured in milliseconds or seconds. This incoming data typically arrives in an unstructured or semi-structured format, such as JSON, and has the same processing requirements as batch processing, but with shorter turnaround times to support real-time consumption.

Processed data is often written to an analytical data store, which is optimized for analytics and visualization. The processed data can also be ingested directly into the analytics and reporting layer for analysis, business intelligence, and real-time dashboard visualization.

Architecture

A real-time processing architecture has the following logical components.

- **Real-time message ingestion.** The architecture must include a way to capture and store real-time messages to be consumed by a stream processing consumer. In simple cases, this service could be implemented as a simple data store in which new messages are deposited in a folder. But often the solution requires a message broker, such as Azure Event Hubs, that acts as a buffer for the messages. The message broker should support scale-out processing and reliable delivery.
- **Stream processing.** After capturing real-time messages, the solution must process them by filtering, aggregating, and otherwise preparing the data for analysis.
- **Analytical data store.** Many big data solutions are designed to prepare data for analysis and then serve the processed data in a structured format that can be queried using analytical tools.
- **Analysis and reporting.** The goal of most big data solutions is to provide

insights into the data through analysis and reporting.

Technology choices

The following technologies are recommended choices for real-time processing solutions in Azure.

Real-time message ingestion

- **Azure Event Hubs**. Azure Event Hubs is a messaging solution for ingesting millions of event messages per second. The captured event data can be processed by multiple consumers in parallel. While Event Hubs natively supports AMQP (Advanced Message Queuing Protocol 1.0), it also provides a binary compatibility layer that allows applications using the Kafka protocol (Kafka 1.0 and above) to process events using Event Hubs with no application changes.
- **Azure IoT Hub**. Azure IoT Hub provides bi-directional communication between Internet-connected devices, and a scalable message queue that can handle millions of simultaneously connected devices.
- **Apache Kafka**. Kafka is an open source message queuing and stream processing application that can scale to handle millions of messages per second from multiple message producers, and route them to multiple consumers. Kafka is available in Azure as an HDInsight cluster type.

Data storage

- **Azure Storage Blob Containers** or **Azure Data Lake Store**. Incoming real-time data is usually captured in a message broker (see above), but in some scenarios, it can make sense to monitor a folder for new files and process them as they are created or updated. Additionally, many real-time processing solutions combine streaming data with static reference data, which can be stored in a file store. Finally, file storage may be used as an output destination for captured real-time data for archiving, or for further batch processing in a lambda architecture.

Stream processing

- **Azure Stream Analytics**. Azure Stream Analytics can run perpetual queries against an unbounded stream of data. These queries consume streams of data from storage or message brokers, filter and aggregate the data based on temporal windows, and write the results to sinks such as storage, databases, or directly to reports in Power BI. Stream Analytics uses a SQL-based query language that supports temporal and geospatial constructs, and can be

extended using JavaScript.

- **Storm**. Apache Storm is an open source framework for stream processing that uses a topology of spouts and bolts to consume, process, and output the results from real-time streaming data sources. You can provision Storm in an Azure HDInsight cluster, and implement a topology in Java or C#.
- **Spark Streaming**. Apache Spark is an open source distributed platform for general data processing. Spark provides the Spark Streaming API, in which you can write code in any supported Spark language, including Java, Scala, and Python. Spark 2.0 introduced the Spark Structured Streaming API, which provides a simpler and more consistent programming model. Spark 2.0 is available in an Azure HDInsight cluster.

Analytical data store

- **Azure Synapse Analytics**, **HBase**, **Spark**, or **Hive**. Processed real-time data can be stored in a relational database such Synapse Analytics, a NoSQL store such as HBase, or as files in distributed storage over which Spark or Hive tables can be defined and queried.

Analytics and reporting

- **Azure Analysis Services**, **Power BI**, and **Microsoft Excel**. Processed real-time data that is stored in an analytical data store can be used for historical reporting and analysis in the same way as batch processed data. Additionally, Power BI can be used to publish real-time (or near-real-time) reports and visualizations from analytical data sources where latency is sufficiently low, or in some cases directly from the stream processing output.

Choosing an analytical data store in Azure

There are several options for data serving storage in Azure, depending on your needs:

- Azure Synapse Analytics
- Azure SQL Database
- SQL Server in Azure VM
- HBase/Phoenix on HDInsight
- Hive LLAP on HDInsight
- Azure Analysis Services
- Azure Cosmos DB

General capabilities

Capability	SQL Database	Azure Synapse	HBase/ Phoenix on HDInsight	Hive LLAP on HDInsight	Azure Analysis Services	Cosmos DB
Is managed service	Yes	Yes	Yes 1	Yes 1	Yes	Yes
Primary database model	Relational (columnar format when using columnstore indexes)	Relational tables with columnar storage	Wide column store	Hive/In-Memory	Tabular/ MOLAP semantic models	Document store, graph, key-value store, wide column store
SQL language support	Yes	Yes	Yes (using Phoenix JDBC driver)	Yes	No	Yes
Optimized for speed serving layer	Yes 2	No	Yes	Yes	No	Yes

Scalability capabilities

Capability	SQL Database	Azure Synapse	HBase/ Phoenix on HDInsight	Hive LLAP on HDInsight	Azure Analysis Services	Cosmos DB

Redundant regional servers for high availability	Yes	Yes	Yes	No	No	Yes
Supports query scale out	No	Yes	Yes	Yes	Yes	Yes
Dynamic scalability (scale up)	Yes	Yes	No	No	Yes	Yes
Supports in-memory caching of data	Yes	Yes	No	Yes	Yes	No

Security capabilities

Capability	**SQL Database**	**Azure Synapse**	**HBase/ Phoenix on HDInsight**	**Hive LLAP on HDInsight**	**Azure Analysis Services**	**Cosmos DB**
Authentication	SQL / Azure Active Directory (Azure AD)	SQL / Azure AD	local / Azure AD 1	local / Azure AD 1	Azure AD	database users / Azure AD via access control (IAM)
Data encryption at rest	Yes 2	Yes 2	Yes 1	Yes 1	Yes	Yes

Row-level security	Yes	Yes 3	Yes 1	Yes 1	Yes (through object-level security in model)	No
Supports firewalls	Yes	Yes	Yes 4	Yes 4	Yes	Yes
Dynamic data masking	Yes	No	Yes 1	Yes	No	No

Choosing a data analytics technology in Azure

General capabilities

Capability	**Power BI**	**Jupyter Notebooks**	**Zeppelin Notebooks**	**Microsoft Azure Notebooks**
Connect to big data cluster for advanced processing	Yes	Yes	Yes	No
Managed service	Yes	Yes 1	Yes 1	Yes
Connect to 100s of data sources	Yes	No	No	No
Offline capabilities	Yes 2	No	No	No
Embedding capabilities	Yes	No	No	No
Automatic data refresh	Yes	No	No	No
Access to numerous open source packages	No	Yes 3	Yes 3	Yes 4

Data transformation/cleansing options	Power Query, R	40 languages, including Python, R, Julia, and Scala	20+ interpreters, including Python, JDBC, and R	Python, F#, R
Pricing	Free for Power BI Desktop (authoring), see pricing for hosting options	Free	Free	Free
Multiuser collaboration	Yes	Yes (through sharing or with a multiuser server like JupyterHub)	Yes	Yes (through sharing)

Choosing a batch processing technology in Azure

Azure Synapse Analytics

Azure Synapse is a distributed system designed to perform analytics on large data. It supports massive parallel processing (MPP), which makes it suitable for running high-performance analytics. Consider Azure Synapse when you have large amounts of data (more than 1 TB) and are running an analytics workload that will benefit from parallelism.

Azure Data Lake Analytics

Data Lake Analytics is an on-demand analytics job service. It is optimized for distributed processing of very large data sets stored in Azure Data Lake Store.

- Languages: U-SQL (including Python, R, and C# extensions).
- Integrates with Azure Data Lake Store, Azure Storage blobs, Azure SQL Database, and Azure Synapse.
- Pricing model is per-job.

HDInsight

HDInsight is a managed Hadoop service. Use it deploy and manage Hadoop clusters in Azure. For batch processing, you can use Spark, Hive, Hive LLAP, MapReduce.

- Languages: R, Python, Java, Scala, SQL
- Kerberos authentication with Active Directory, Apache Ranger based access control
- Gives you full control of the Hadoop cluster

Azure Databricks
Azure Databricks is an Apache Spark-based analytics platform. You can think of it as "Spark as a service." It's the easiest way to use Spark on the Azure platform.

- Languages: R, Python, Java, Scala, Spark SQL
- Fast cluster start times, autotermination, autoscaling.
- Manages the Spark cluster for you.
- Built-in integration with Azure Blob Storage, Azure Data Lake Storage (ADLS), Azure Synapse, and other services. See Data Sources.
- User authentication with Azure Active Directory.
- Web-based notebooks for collaboration and data exploration.
- Supports GPU-enabled clusters

Azure Distributed Data Engineering Toolkit
The Distributed Data Engineering Toolkit (AZTK) is a tool for provisioning on-demand Spark on Docker clusters in Azure.
AZTK is not an Azure service. Rather, it's a client-side tool with a CLI and Python SDK interface, that's built on Azure Batch. This option gives you the most control over the infrastructure when deploying a Spark cluster.

- Bring your own Docker image.
- Use low-priority VMs for an 80% discount.
- Mixed mode clusters that use both low-priority and dedicated VMs.
- Built in support for Azure Blob Storage and Azure Data Lake connection.

Capability matrix

The following tables summarize the key differences in capabilities.

General capabilities

Capability	Azure Data Lake Analytics	Azure Synapse	HDInsight	Azure Databricks
Is managed service	Yes	Yes	Yes 1	Yes
Relational data store	Yes	Yes	No	No

Pricing model	Per batch job	By cluster hour	By cluster hour	Databricks Unit2 + cluster hour

Capabilities

Capability	Azure Data Lake Analytics	Azure Synapse	HDInsight with Spark	HDInsight with Hive	HDInsight with Hive LLAP	Azure Databricks
Autoscaling	No	No	No	No	No	Yes
Scale-out granularity	Per job	Per cluster	Per cluster	Per cluster	Per cluster	Per cluster
In-memory caching of data	No	Yes	Yes	No	Yes	Yes
Query from external relational stores	Yes	No	Yes	No	No	Yes
Authentication	Azure AD	SQL / Azure AD	No	Azure AD1	Azure AD1	Azure AD
Auditing	Yes	Yes	No	Yes 1	Yes 1	Yes
Row-level security	No	Yes2	No	Yes 1	Yes 1	No
Supports firewalls	Yes	Yes	Yes	Yes 3	Yes 3	No
Dynamic data masking	No	Yes	No	Yes 1	Yes 1	No

Choosing a data pipeline orchestration technology in Azure

Most big data solutions consist of repeated data processing operations, encapsulated in workflows. A pipeline orchestrator is a tool that helps to automate these workflows. An orchestrator can schedule jobs, execute workflows, and coordinate dependencies among tasks.

What are your options for data pipeline orchestration?

In Azure, the following services and tools will meet the core requirements for pipeline orchestration, control flow, and data movement:

- Azure Data Factory
- Oozie on HDInsight
- SQL Server Integration Services (SSIS)

These services and tools can be used independently from one another, or used together to create a hybrid solution. For example, the Integration Runtime (IR) in Azure Data Factory V2 can natively execute SSIS packages in a managed Azure compute environment. While there is some overlap in functionality between these services, there are a few key differences.

General capabilities

Capability	Azure Data Factory	SQL Server Integration Services (SSIS)	Oozie on HDInsight
Managed	Yes	No	Yes
Cloud-based	Yes	No (local)	Yes
Prerequisite	Azure Subscription	SQL Server	Azure Subscription, HDInsight cluster
Management tools	Azure Portal, PowerShell, CLI, .NET SDK	SSMS, PowerShell	Bash shell, Oozie REST API, Oozie web UI
Pricing	Pay per usage	Licensing / pay for features	No additional charge on top of running the HDInsight cluster

Pipeline capabilities

Capability	Azure Data Factory	SQL Server Integration Services (SSIS)	Oozie on HDInsight

Copy data	Yes	Yes	Yes
Custom transformations	Yes	Yes	Yes (MapReduce, Pig, and Hive jobs)
Azure Machine Learning scoring	Yes	Yes (with scripting)	No
HDInsight On-Demand	Yes	No	No
Azure Batch	Yes	No	No
Pig, Hive, MapReduce	Yes	No	Yes
Spark	Yes	No	No
Execute SSIS Package	Yes	Yes	No
Control flow	Yes	Yes	Yes
Access on-premises data	Yes	Yes	No

Scalability capabilities

Capability	Azure Data Factory	SQL Server Integration Services (SSIS)	Oozie on HDInsight
Scale up	Yes	No	No
Scale out	Yes	No	Yes (by adding worker nodes to cluster)
Optimized for big data	Yes	No	Yes

Choosing a real-time message ingestion technology in Azure

Real time processing deals with streams of data that are captured in real-time and processed with minimal latency. Many real-time processing solutions need a message ingestion store to act as a buffer for messages, and to support scale-out processing, reliable delivery, and other message queuing semantics.

What are your options for real-time message ingestion?

- Azure Event Hubs

- Azure IoT Hub
- Kafka on HDInsight

Azure Event Hubs

Azure Event Hubs is a highly scalable data streaming platform and event ingestion service, capable of receiving and processing millions of events per second. Event Hubs can process and store events, data, or telemetry produced by distributed software and devices. Data sent to an event hub can be transformed and stored using any real-time analytics provider or batching/storage adapters. Event Hubs provides publish-subscribe capabilities with low latency at massive scale, which makes it appropriate for big data scenarios.

Azure IoT Hub

Azure IoT Hub is a managed service that enables reliable and secure bidirectional communications between millions of IoT devices and a cloud-based back end. Feature of IoT Hub include:

- Multiple options for device-to-cloud and cloud-to-device communication. These options include one-way messaging, file transfer, and request-reply methods.
- Message routing to other Azure services.
- Queryable store for device metadata and synchronized state information.
- Secure communications and access control using per-device security keys or X.509 certificates.
- Monitoring of device connectivity and device identity management events.

In terms of message ingestion, IoT Hub is similar to Event Hubs. However, it was specifically designed for managing IoT device connectivity, not just message ingestion. For more information, see Comparison of Azure IoT Hub and Azure Event Hubs.

Kafka on HDInsight

Apache Kafka is an open-source distributed streaming platform that can be used to build real-time data pipelines and streaming applications. Kafka also provides message broker functionality similar to a message queue, where you can publish and subscribe to named data streams. It is horizontally scalable, fault-tolerant, and extremely fast. Kafka on HDInsight provides a Kafka as a managed, highly scalable, and highly available service in Azure.
Some common use cases for Kafka are:

- **Messaging**. Because it supports the publish-subscribe message pattern, Kafka is often used as a message broker.

- **Activity tracking**. Because Kafka provides in-order logging of records, it can be used to track and re-create activities, such as user actions on a web site.
- **Aggregation**. Using stream processing, you can aggregate information from different streams to combine and centralize the information into operational data.
- **Transformation**. Using stream processing, you can combine and enrich data from multiple input topics into one or more output topics.

Capability matrix

The following tables summarize the key differences in capabilities.

Capability	IoT Hub	Event Hubs	Kafka on HDInsight
Cloud-to-device communications	Yes	No	No
Device-initiated file upload	Yes	No	No
Device state information	Device twins	No	No
Protocol support	MQTT, AMQP, HTTPS 1	AMQP, HTTPS	Kafka Protocol
Security	Per-device identity; revocable access control.	Shared access policies; limited revocation through publisher policies.	Authentication using SASL; pluggable authorization; integration with external authentication services supported.

Choosing a search data store in Azure

What are your options when choosing a search data store?

In Azure, all of the following data stores will meet the core requirements for search against free-form text data by providing a search index:

- Azure Search
- Elasticsearch
- HDInsight with Solr

- Azure SQL Database with full text search

Capability matrix

The following tables summarize the key differences in capabilities.

General capabilities

Capability	Azure Search	Elasticsearch	HDInsight with Solr	SQL Database
Is managed service	Yes	No	Yes	Yes
REST API	Yes	Yes	Yes	No
Programmability	.NET	Java	Java	T-SQL
Document indexers for common file types (PDF, DOCX, TXT, and so on)	Yes	No	Yes	No

Manageability capabilities

Capability	Azure Search	Elasticsearch	HDInsight with Solr	SQL Database
Updateable schema	No	Yes	Yes	Yes
Supports scale out	Yes	Yes	Yes	No

Analytic workload capabilities

Capability	Azure Search	Elasticsearch	HDInsight with Solr	SQL Database
Supports analytics beyond full text search	No	Yes	Yes	Yes
Part of a log analytics stack	No	Yes (ELK)	No	No

Supports semantic search	Yes (find similar documents only)	Yes	Yes	Yes

Security capabilities

Capability	Azure Search	Elasticsearch	HDInsight with Solr	SQL Database
Row-level security	Partial (requires application query to filter by group id)	Partial (requires application query to filter by group id)	Yes	Yes
Transparent data encryption	No	No	No	Yes
Restrict access to specific IP addresses	No	Yes	Yes	Yes
Restrict access to allow virtual network access only	No	Yes	Yes	Yes
Active Directory authentication (integrated authentication)	No	No	No	Yes

Choosing a stream processing technology in Azure

Real-time stream processing consumes messages from either queue or file-based storage, process the messages, and forward the result to another message queue, file store, or database. Processing may include querying, filtering, and aggregating messages. Stream processing engines must be able to consume an endless streams of data and produce results with minimal latency.

What are your options when choosing a technology for real-time processing?

In Azure, all of the following data stores will meet the core requirements supporting real-time processing:

- Azure Stream Analytics
- HDInsight with Spark Streaming
- Apache Spark in Azure Databricks
- HDInsight with Storm
- Azure Functions
- Azure App Service WebJobs

Capability matrix

The following tables summarize the key differences in capabilities.

General capabilities

Capability	Azure Stream Analytics	HDInsight with Spark Streaming	Apache Spark in Azure Databricks	HDInsight with Storm	Azure Functions	Azure App Service WebJobs
Programmability	Stream analytics query language, JavaScript	C#/F#, Java, Python, Scala	C#/F#, Java, Python, R, Scala	C#, Java	C#, F#, Java, Node.js, Python	C#, Java, Node.js, PHP, Python
Programming paradigm	Declarative	Mixture of declarative and imperative	Mixture of declarative and imperative	Imperative	Imperative	Imperative

Pricing model	Streaming units	Per cluster hour	Databricks units	Per cluster hour	Per function execution and resource consumption	Per app service plan hour

Integration capabilities

Capability	**Azure Stream Analytics**	**HDInsight with Spark Streaming**	**Apache Spark in Azure Databricks**	**HDInsight with Storm**	**Azure Functions**	**Azure App Service WebJobs**
Inputs	Azure Event Hubs, Azure IoT Hub, Azure Blob storage	Event Hubs, IoT Hub, Kafka, HDFS, Storage Blobs, Azure Data Lake Store	Event Hubs, IoT Hub, Kafka, HDFS, Storage Blobs, Azure Data Lake Store	Event Hubs, IoT Hub, Storage Blobs, Azure Data Lake Store	Supported bindings	Service Bus, Storage Queues, Storage Blobs, Event Hubs, WebHooks, Cosmos DB, Files

Sinks	Azure Data Lake Store, Azure SQL Database, Storage Blobs, Event Hubs, Power BI, Table Storage, Service Bus Queues, Service Bus Topics, Cosmos DB, Azure Functions	HDFS, Kafka, Storage Blobs, Azure Data Lake Store, Cosmos DB	HDFS, Kafka, Storage Blobs, Azure Data Lake Store, Cosmos DB	Event Hubs, Service Bus, Kafka	Supported bindings	Service Bus, Storage Queues, Storage Blobs, Event Hubs, WebHooks, Cosmos DB, Files

Processing capabilities

Capability	Azure Stream Analytics	HDInsight with Spark Streaming	Apache Spark in Azure Databricks	HDInsight with Storm	Azure Functions	Azure App Service WebJobs
Built-in temporal/windowing support	Yes	Yes	Yes	Yes	No	No
Input data formats	Avro, JSON or CSV, UTF-8 encoded	Any format using custom code	Any format using custom code	Any format using custom code	Any format using custom code	Any format using custom code

Scalability	Query partitions	Bounded by cluster size	Bounded by Databricks cluster scale configuration	Bounded by cluster size	Up to 200 function app instances processing in parallel	Bounded by app service plan capacity
Late arrival and out of order event handling support	Yes	Yes	Yes	Yes	No	No